Human Activity Recognition Using Smartphone Sensor Data: Analysis and Classification using Azure ML and Power BI

Séamus Knightly
1MECE1
University Of Galway
Galway, Ireland
s.knightly1@universityofgalway.ie

Seán Kelly
1MECE1
University Of Galway
Galway, Ireland
s.kelly178@universityofgalway.ie

***Abstract*—blah blah**

***Index Terms*—IoT, Remote Patient Monitoring, Accelerometer, Gyroscope, Classification, Assisted Living**

I. Introduction

Human Activity Recognition (HAR) is an important area in the domain of machine learning and medical computing. The focus is the identification and classification of human physical activities based on provided sensor data. The data can be collecte by a wide range of devices, from stationary room sensing video or radar that can monitor multiple subjects, to wearable or mobile devices that can monitor individual subjects. HAR has become an important component in applications such as remote patient monitoring and assisted living.

There are a number of large, labelled datasets available for human activity recognition. For this project, the *Human Activity Recognition Using Smartphones* dataset [1] was chosen. This dataset contains sensor data collected from smartphone accelerometers and gyroscopes as participants performed activities such walking, sitting, and standing. The observations are represented by multiple time and frequency features extracted from the motion signals. Sensor signals from the device's accelerometer and gyroscope were recorded at 50 Hz and processed into 561 time and frequency features.

This dataset was chosen over other datasets, such as HARTH dataset [2] [3] for its simplicity, requiring only 1 worn sensor. And for the simpler set of classifications, which correspond better with scenarios that would arise in a care facility or other healthcare setting.

This paper aims to analyse and classify human activities using the UCI HAR dataset by developing and evaluating multiple machine learning models in Azure Machine Learning Studio. Power BI is employed to visualize data characteristics, feature distributions, and model performance metrics. The results provide insights into the role of sensor based data and machine learning in Human Activity Recognition.

The introductory paper [4] demonstrated that a multiclass Support Vector Machine (SVM) model could achieve an overall classification accuracy of 96% on this dataset, comparable to or exceeding the performance of systems using specialised wearable sensors. Their work highlighted the feasibility of using simpler accelerometer and gyroscope sensors, such as smart phones, as unobtrusive, affordable, and reliable sensing tools for HAR.

II. Dataset Description

A. Dataset Source

The dataset used in this study is the *Human Activity Recognition Using Smartphones* dataset [1], first introduced in 2013. It was created as part of a study on human centered computing at the University of Genova and the Universitat Politècnica de Catalunya [4]. The dataset consists of sensor data collected from 30 volunteers aged between 19 and 48 years, performing six basic activities of daily living.

Each participant carried a Samsung Galaxy S II smartphone on the waist while performing the activities. The smartphone's embedded triaxial accelerometer and gyroscope were used to capture linear acceleration and angular velocity at a constant sampling rate of 50 Hz. The experiment was carried out in a controlled laboratory environment, with participants following a standardised sequence of activities to ensure consistency across samples. The signals were normalised and bounded within [1, -1], and each feature vector is also standardised.

The dataset is publicly available from the UCI Machine Learning Repository under an open license.

B. Feature Overview

The dataset provides both raw sensor readings and preprocessed feature vectors based on that raw data. Each observation represents a 2.56 second window of sensor data with 50% overlap between adjacent windows, there are 128 readings per window. From each window, 561 features were extracted from the time and frequency domains.

The features include statistical and signal-based measures such as mean, standard deviation, median absolute deviation, signal magnitude area, energy, entropy, interquartile range, and correlation between sensor axes. Frequency-domain features were obtained using Fast Fourier Transform (FFT) on the windowed signals. These features capture important motion characteristics for each activity and are normalized to ensure comparability across subjects and sessions.

Each record in the dataset is labeled with one of six activity classes:

- Walking
- Walking Upstairs
- Walking Downstairs
- Sitting
- Standing
- Laying

Additionally, subject identifiers (1–30) are included, allowing for subject specific or subject independent analysis. In the dataset, these activities are encoded numerically as follows: 1–*Walking*, 2–*Walking Upstairs*, 3–*Walking Downstairs*, 4–*Sitting*, 5–*Standing*, and 6–*Laying*.

C. Data Preparation

Prior to model training, a subset of the original dataset features was selected to simplify the analysis and reduce computational overhead. The full UCI HAR dataset contains 561 features derived from both time and frequency transformations. This feature set provides detailed motion characterisation, but it is computationally expensive to process, particularly on the edge in the Rasbperry Pi.

To address this, a representative subset of features was selected, focusing on descriptive statistics (mean, maximum, and minimum values) from the accelerometer and gyroscope signals in both body and gravity components. The retained features include the following categories:

The retained features include the following categories:

- **Time domain features:** body acceleration, body acceleration jerk, gyroscope, and gyroscope jerk (mean, max, and min for each axis)
- **Magnitude features:** signal magnitude for acceleration, jerk, and gyroscope signals
- **Frequency domain features:** selected mean, max, and min values for acceleration, jerk, and gyroscope signals

This reduced dataset comprises 101 variables in total, including 100 descriptive features and one target label representing the activity class. The selected features preserve the essential statistical and dynamic characteristics of human motion while reducing data dimensionality. This reduction improves computational efficiency, especially on the edge where limited processing power and energy consumption are key considerations.

Furthermore, the original study [4] employed a Support Vector Machine (SVM) classifier, which required extensive pre-processed features to achieve high accuracy. In contrast, this work explores more modern machine learning techniques capable of capturing non-linear relationships and implicit feature interactions. It is therefore expected that the reliance on an extensive set of features is reduced without sacrificing accuracy.

On the Azure Machine Learning side, the `subject` feature was excluded from model training to prevent the classifier from learning participant specific patterns. This ensures that the model generalises to unseen individuals rather than overfitting to sensor characteristics unique to particular subjects.

The dataset was then partitioned into training (80%) and testing (20%) subsets for model evaluation.

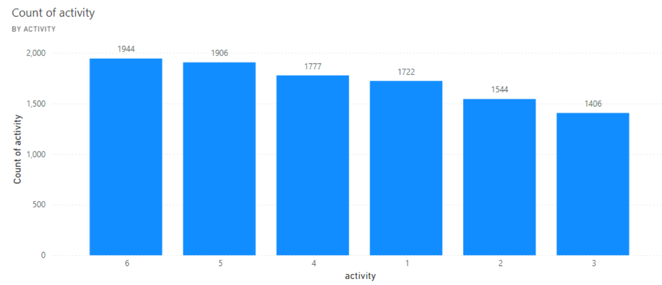


Fig. 1. Distribution of recorded samples across the six activity classes.

III. Data Analysis

A. Data Characteristics

The class distribution, illustrated in Fig. 1, shows that the dataset is generally well balanced across the six activity categories. Each activity contains roughly a similar number of samples, ensuring that the classification problem is not significantly biased toward any single class. However, two activities, *Walking Upstairs* and *Walking Downstairs*, are slightly underrepresented, with approximately ≈1,400 and ≈1,500 samples respectively, compared to the remaining activities which range between ≈1,700 and ≈1,900 samples each. This imbalance is not expected to adversely affect model performance but is worth noting when interpreting classification accuracy across classes.

Figure 2 plots the mean gyroscope signal along the X-axis against the corresponding mean accelerometer signal. The data form a horizontally elongated cluster centered near 0.25 on the gyroscope axis and 0 on the accelerometer axis. This indicates that angular velocity varies more widely across activities than linear acceleration. The limited vertical spread reflects the fact that participants do not remain perfectly upright—small postural adjustments introduce minor accelerometer fluctuations around zero. The positive offset of the gyroscope mean likely arises because most activities involve forward body motion, causing a consistent rotational component in the positive X-direction. This bias is expected given the orientation of the device and that half of the dataset involves walking movements. The weak overall correlation confirms that the gyroscope and accelerometer capture distinct motion characteristics.

Figure 3 shows the distributions of several representative accelerometer and gyroscope mean features. The features are bounded within $[-1, 1]$, confirming that normalisation was applied to the dataset to remove sensor bias and ensure comparable scaling across variables. Most distributions are centered near zero, consistent with mean centering of the sensor signals, although slight offsets are visible in some axes due to natural asymmetries in human motion and device

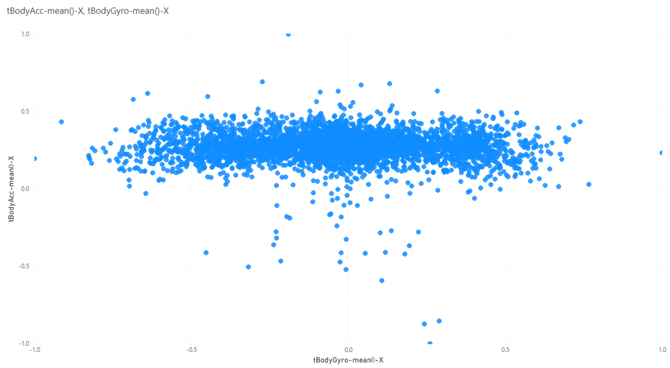


Fig. 2. Scatter plot showing the relationship between the mean body gyroscope signal (`tBodyGyro-mean()-X`) and the mean body acceleration signal (`tBodyAcc-mean()-X`).

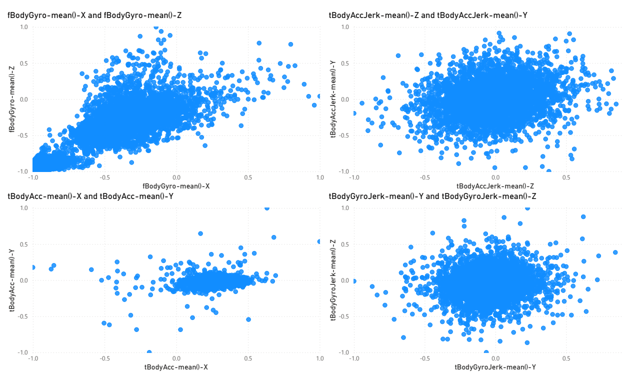


Fig. 3. Histograms of representative accelerometer and gyroscope mean features, showing that all sensor readings are centered near zero and lie within the normalized range of $[-1, 1]$.

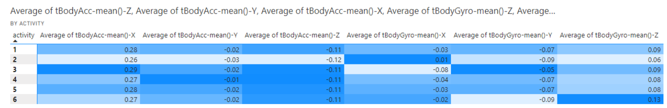
orientation. The histograms verify that the dataset is in a state suitable for machine learning analysis.

The frequency-domain gyroscope features (`fBodyGyro-mean()-X` and `fBodyGyro-mean()-Z`) in the top left of Fig. 3 display a wedge shaped distribution, in contrast to the more symmetric time domain features. This reflects the characteristics of spectral magnitudes produced by the Fast Fourier Transform. Low frequency components appear more consistently, and high frequency components occur less frequently but with greater variability.

The differences show the complementary relationship between time and frequency domain, with time features capturing instantaneous motion and orientation, frequency features show periodicity and energy characteristics. Including both will give the model better information about the motion patterns.

B. Trends and Patterns

Figure 4 illustrates the mean body acceleration and gyroscope signals across all activities. The mean acceleration values are relatively uniform across classes, reflecting the normalisation of sensor data and the short, steady recording intervals. Subtle variations are visible in the gyroscope means, particularly along the Z-axis, corresponding to torso rotation


Average of tBodyAcc-mean()-Z, Average of tBodyAcc-mean()-Y, Average of tBodyAcc-mean()-X, Average of tBodyGyro-mean()-Z, Average...
BY ACTIVITY

activity	Average of tBodyAcc-mean()-X	Average of tBodyAcc-mean()-Y	Average of tBodyAcc-mean()-Z	Average of tBodyGyro-mean()-X	Average of tBodyGyro-mean()-Y	Average of tBodyGyro-mean()-Z
1	0.28	-0.02	-0.11	-0.03	-0.07	0.09
2	0.26	-0.03	-0.12	0.01	-0.09	0.06
3	0.29	-0.02	-0.11	-0.08	-0.05	0.09
4	0.27	-0.01	-0.11	-0.04	-0.07	0.08
5	0.28	-0.02	-0.11	-0.03	-0.07	0.08
6	0.27	-0.02	-0.11	-0.02	-0.09	0.13

Fig. 4. Heatmap of selected mean accelerometer and gyroscope features across activity classes. Dynamic activities show higher mean magnitudes than static postures.

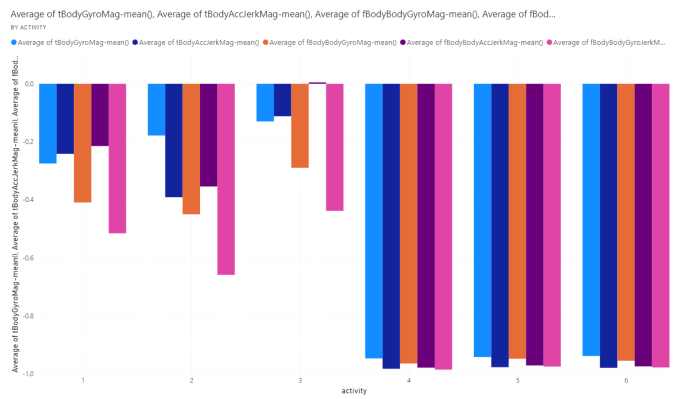


Fig. 5. Comparison of mean motion magnitude features across activities. Dynamic activities exhibit higher overall intensity than static postures.

and orientation differences during activities such as walking or laying. These differences are expected given that the device was worn at the waist, where small postural changes and rotational motion are captured more strongly in the gyroscope signals. Overall, the heatmap confirms that the dataset's body motion features capture the small variations between activity types.

C. Feature Comparison

Figure 5 compares the mean magnitudes of selected motion features across all activity classes. These features, including the body acceleration magnitude (`tBodyAccMag-mean()`), gyroscope magnitude (`tBodyGyroMag-mean()`), and jerk-derived magnitudes, represent the overall intensity of body motion.

A separation is visible between dynamic and static activities. *Walking*, *Walking Upstairs*, and *Walking Downstairs* (classes 1–3) exhibit notably higher magnitude values across all features, indicating stronger overall motion and rotational activity. Among these, *Walking Downstairs* shows significantly higher jerk magnitudes, reflecting the more abrupt deceleration and impact forces during descent.

In contrast, static postures such as *Sitting*, *Standing*, and *Laying* (classes 4–6) display consistently lower mean magnitudes across all sensors. The near zero variation among these static classes suggests limited body movement and stable device orientation. These results confirm that magnitude based features can capture the motion intensity of each activity and give discriminative information for classification.

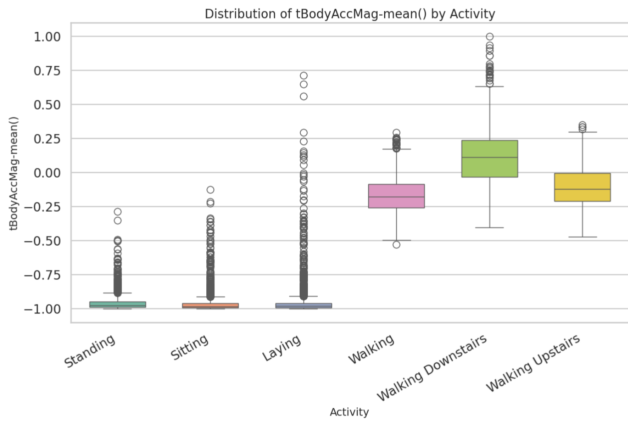


Fig. 6. Box plot of mean body acceleration magnitude (`tBodyAccMag-mean()`) across activity classes. Dynamic activities display higher medians and greater variability than static postures, indicating stronger motion dynamics.

D. Box Plot Analysis

Box plots were generated to compare the distributions of selected features across activity categories (Fig. 6). The mean body acceleration magnitude (`tBodyAccMag-mean()`) has a clear separation between static and dynamic activities. Static postures, *Standing*, *Sitting*, and *Laying*, cluster tightly around -1, indicating near zero net body acceleration after gravity removal. In contrast, dynamic activities (*Walking*, *Walking Upstairs*, and *Walking Downstairs*) show higher medians and broader interquartile ranges, as they have higher motion intensity and variation.

For the dynamic categories, *Walking Downstairs* shows the highest median acceleration magnitude, while *Walking Upstairs* has the greatest spread, consistent with the more irregular motion of climbing. Although this feature alone distinguishes static from dynamic activities, the overlap between static classes (e.g., *Sitting* and *Standing*) suggests that multiple features are required for finer classification.

E. Findings and Discussion

The data analysis highlights several consistent and physically meaningful trends across the dataset. First, the activity classes are well balanced, ensuring that subsequent classification experiments are not biased toward any single class. Normalisation has scaled all features within the range $[-1, 1]$, removing sensor bias and facilitating comparison across variables.

Feature comparisons show separations between static and dynamic activities across both time and frequency domains. Accelerometer features mostly capture translational body motion, making them effective for distinguishing movement, while gyroscope features show the rotation, which separates similar locomotion patterns such as *Walking Upstairs* and *Walking Downstairs*. Frequency features provide extra information about periodicity and energy of movement, which will be useful for classification.

Overall, the data analysis confirms that the selected features are as expected, are physically interpretable, and have distinct patterns across activity classes. These characteristics make the dataset suitable for supervised machine learning models in Azure ML studio.

IV. Machine Learning Model Development and Evaluation

Machine learning models were developed and evaluated in Microsoft Azure ML Studio using the processed dataset described in Section II-C. A multi-class classification problem was defined, where the target label corresponds to one of six human activity classes. All models were trained using an 70/30 train–test split, and performance was assessed through the use of the `Evaluate Model` component in ML Studio.

The experiments were conducted according to the project specification.

A. Single Feature Models

In the first experiment, 99 models were trained using a single feature each from the dataset. This was done using the *Execute Python Script* component in Azure ML Studio, which was useful to automate the mass training and testing of the models. Logistic Regression was used for all, the code for this can be found in Appendix A

The resulting accuracies ranged from ≈17%, roughly equivalent to a random choice, to ≈ 52%, depending on the selected feature. The most informative features were primarily derived from mean accelerometer and gyroscope signals in the time domain, specifically: `fBodyAcc-max()-X`, `tGravityAcc-min()-Y`, `tGravityAcc-mean()-Y`, `tGravityAcc-max()-Y`, `tBodyAcc-max()-X`, and `fBodyGyro-max()-X`. These activities had accuracies around ≈ 50%, With Gravity related features being over-represented. This matches the physical expectation however, as these features will distinguish the static and dynamic categories. This instantly gives a boost as it reduces the chance to 1 in 3 on a random choice as 3 of the 6 activities are dynamic.

The top-performing single-feature model achieved an accuracy of approximately 52.18%, using `fBodyAcc-max()-X`. This indicates that horizontal body acceleration alone carries substantial information about activity state, but cannot independently distinguish similar postures such as *Sitting* and *Standing*. Conversely, frequency-domain features (e.g., `fBodyAcc-bandsEnergy()`) showed lower individual predictive power, confirming that composite motion characteristics are better captured through combinations of features. Conversely, the worst performing was tBodyAcc-mean()-Y with 0.170874

The resulting accuracies ranged from ≈ 17%, up to ≈ 52%, depending on the selected feature. The top 6 most informative single features were: `fBodyAcc-max()-X`, tGravityAcc-min()-Y, tGravityAcc-mean()-Y, tGravityAcc-max()-Y, tBodyAcc-max()-X, and fBodyGyro-max()-X. These features achieved accuracies around 50%, with gravity related features being over-represented.

This aligns with the expected physical behaviour of the system. Gravity features separate static postures (such as *Standing*, *Sitting*, and *Laying*) from dynamic movements (*Walking*, *Walking Upstairs*, and *Walking Downstairs*). This binary separation immediately improves the classification likelihood from one in six to approximately one in three by distinguishing motion categories.

The top single feature model achieved an accuracy of 52.18% using fBodyAcc-max()-X, indicating that horizontal body acceleration carries substantial information about activity state. However, it alone cannot resolve fine distinctions between static activities such as *Sitting* and *Standing*.

The weakest performing feature was tBodyAcc-mean()-Y, which achieved only 17.1% accuracy, which is roughly equivalent to random guessing on the 6 activities. This feature likely captures minimal variation between activities because the Y-axis component of linear acceleration is effected by constant gravitational force of the earth when the device is worn on the waist, masking activity dependent dynamics.

Overall, the single feature experiment demonstrates that individual sensor channels can contain some discriminative power, but accurate human activity recognition requires the integration of multiple complementary signals capturing both translational and rotational motion characteristics.

B. Combination of Features

In the second experiment, a small representative subset of features was used to evaluate how combining motion signals from multiple axes influences classification performance. The selected features represent the mean time domain linear acceleration and angular velocity along the three body axes:

- tBodyAcc-mean()-X, tBodyAcc-mean()-Y, tBodyAcc-mean()-Z
- tBodyGyro-mean()-X, tBodyGyro-mean()-Y, tBodyGyro-mean()-Z

These six features were chosen because they correspond directly to the type of raw sensor readings that would be available on a typical IoT edge device. Each measurement represents a mean value of acceleration or angular velocity from the onboard inertial sensors. This configuration therefore simulates a minimal data representation scenario in which only low-dimensional, easily transmitted sensor data are available prior to cloud processing.

A *Multiclass Logistic Regression* classifier was trained using this six-dimensional feature set. The resulting model achieved test accuracy, precision, and recall values of approximately 27%, a poor improvement above random guessing. This relatively poor performance indicates that while these features capture fundamental movement patterns, they do not provide sufficient discriminatory information on their own to differentiate between activities.

The limited accuracy can be attributed to several factors. Firstly, mean values omit temporal variation and frequency information that are important for distinguishing activities with similar average motion but different periodic structures, like *Walking Upstairs* and *Walking Downstairs*. Secondly, Logistic Regression, being a linear model, is not able to capture the non-linear interactions between accelerometer and gyroscope signals that make up complex human movements, however, incorporating additional derived features like signal magnitude, minimum and maximum values, and frequency domain statistics, can bring out and encode some of these non-linear relationships, allowing linear models to achieve improved performance without changing the underlying algorithm.

This experiment demonstrates that accurate human activity recognition requires both richer feature extraction and potentially more expressive classification algorithms. Non-linear models such as Decision Forests, Support Vector Machines, or Neural Networks are expected to perform better on the same feature subset by capturing interactions that a linear model cannot. These observations motivated the subsequent experiments using expanded feature sets and more complex models.

C. All Features

In the third experiment, all 99 available features in the dataset were used to train a *Multiclass Logistic Regression* classifier. This configuration represents the upper bound of model performance using the full set of derived accelerometer and gyroscope features, including both time and frequency statistics such as mean, standard deviation, magnitude, and energy. These features better capture the motion characteristics of human activity.

The trained model achieved an overall test accuracy of 92.8%, with Micro Precision and Micro Recall values of 0.928, and Macro Precision and Macro Recall values of approximately 0.931 and 0.930, respectively. These results indicate good performance across all six activity classes.

Compared to the previous experiments, the improvement in accuracy demonstrates the importance of including the set of pre-processed features. As discussed previously, the additional features expose the non-linear relationships in the motion signals.

This experiment highlights that high performance can be achieved even with a relatively simple classification algorithm when a sufficiently rich feature set is provided. In IoT, this emphasises the benefit of performing lightweight feature extraction either on the gateway or in the cloud, rather than attempting classification directly on limited raw sensor data at the edge.

D. Feature Selection Methods

To reduce dimensionality and identify the most informative features, three selection techniques were evaluated in

TABLE I
FEATURE SELECTION METHOD PERFORMANCE SUMMARY.

Method	Accuracy (%)
Permutation Feature Importance (Decision Forest)	96.1
Filter Based (Pearson Correlation)	59.6
Filter Based (Chi-Squared)	81.7

TABLE II
SUMMARY OF MODEL PERFORMANCE ACROSS ALL EXPERIMENTS.

Model / Feature Set	Accuracy (%)
Single Feature (best: `fBodyAcc-max()-X`)	52.2
Six Mean Features (Accelerometer + Gyroscope)	27.0
All 99 Features (Logistic Regression)	92.8
Filter Based (Pearson Correlation)	59.6
Filter Based (Chi-Squared)	81.7
Permutation Feature Importance (Decision Forest)	96.1

Azure ML Studio: *Permutation Feature Importance*, and *Filter Based Feature Selection* using the *Pearson Correlation* and *Chi-Squared* criteria (using a `Permutation Feature Importance` component and `Filter Based Feature Selection` components) Each method was followed by model retraining to assess its effect on performance.

Permutation Feature Importance (PFI) was applied using a trained *Multiclass Decision Forest*, achieving an accuracy of 96.1%. The most influential features included `tGravityAcc-min()-X`, `tGravityAcc-min()-Y`, `tBodyAcc-max()-X`, and `fBodyAccMag-max()`.

Using **Filter-Based Selection**, the *Pearson Correlation* method selected ten features with the highest linear correlation to activity class, achieving 59.6% accuracy. In contrast, the *Chi-Squared* method achieved a higher accuracy of 81.7%, identifying both time and frequency domain features such as `tBodyAcc-max()-X`, `tGravityAccMag-max()`, and `fBodyAccMag-mean()` as most relevant.

A summary of results is given in Table I. The PFI method with a Decision Forest provided the best overall accuracy. Chi-Squared filtering offered a good compromise between simplicity and accuracy, making it suitable for resource limited IoT applications.

V. RESULTS COMPARISON AND DISCUSSION

Table II summarises the performance of all experiments conducted in Azure ML Studio. The progression from single features to the full feature set demonstrates a clear relationship between feature richness, model complexity, and classification accuracy.

The single-feature and small-subset experiments highlight that limited raw sensor data provide only partial information about human activity. Individual features could distinguish static from dynamic motion but failed to capture finer variations between activities such as *Walking Upstairs* and *Walking Downstairs*. The poor accuracy of the six-feature model (27%) reinforces this limitation and reflects the reduced representational power of mean values that discard temporal and frequency variation.

In contrast, the full feature model achieved 92.8% accuracy, showing that the engineered features available in the dataset significantly improve separability. Derived statistical, magnitude, and frequency-domain attributes expose latent non-linear relationships that even a linear classifier like Logistic Regression can exploit. This demonstrates that well-designed feature extraction can compensate for model simplicity.

The feature-selection experiments further confirmed this relationship. The Chi-Squared filter retained the most relevant subset of features, achieving 81.7% accuracy—close to the full model but with much lower dimensionality. The Pearson correlation filter, limited to linear dependencies, performed poorly at 59.6%. The Permutation Feature Importance method, evaluated using a Decision Forest, produced the highest overall accuracy (96.1%), reflecting the capability of ensemble models to capture non-linear feature interactions without explicit feature engineering.

From an IoT system design perspective, these results highlight several key trade-offs:

- **Edge vs. Cloud Processing:** Lightweight models on raw sensor data achieve limited accuracy and are suitable only for coarse classification. More complex processing, including feature extraction and model inference, should be offloaded to a cloud platform or high-performance gateway to achieve reliable recognition.
- **Feature Dimensionality:** While the full 99-feature model delivers high accuracy, feature selection can substantially reduce computation and bandwidth costs with minimal performance loss—important for constrained IoT deployments.
- **Model Choice:** Linear models such as Logistic Regression are interpretable and efficient but limited in expressiveness. Non-linear models like Decision Forests or Neural Networks provide superior performance at the cost of higher computational load.

Overall, the experiments demonstrate that high classification accuracy in Human Activity Recognition depends primarily on the availability of informative features. Combining moderate feature selection with non-linear classifiers offers an optimal balance between computational efficiency and predictive accuracy, aligning with the design goals of scalable IoT systems.

REFERENCES

[1] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra. "*Human Activity Recognition Using Smartphones*", UCI Machine Learning Repository, 2013 [Online]. https://doi.org/10.24432/C54S4K.

[2] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "HARTH: A Human Activity Recognition Dataset for Machine Learning," *Sensors*, vol. 21, no. 23, Art. no. 23, Jan. 2021, doi: 10.3390/s21237853.

[3] K. Bach, A. Logacjov, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "A Machine Learning Classifier for Detection of Physical Activity Types and Postures During Free-Living," *Journal for the Measurement of Physical Behaviour*, vol. 1, no. aop, pp. 1–8, Dec. 2021, doi: 10.1123/jmpb.2021-0015.

[4] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," *Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2013)*, Bruges, Belgium, pp. 437–442, Apr. 2013.

Appendix A
Script Used for Single Feature Evaluation in Exectute Python Script Component

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# The script MUST contain a function named azureml_main
# which is the entry point for this module.


# The entry point function MUST have two input arguments.
# If the input port is not connected, the corresponding
# dataframe argument will be None.
#   Param<dataframe1>: a pandas.DataFrame
#   Param<dataframe2>: a pandas.DataFrame
def azureml_main(dataframe1 = None, dataframe2 = None):

df = dataframe1.copy() # copy dataset
y = df['activity'] # activity is the category
X = df.drop(columns=['activity']) # Dataset

results = []

# iterate over the features available
for feature in X.columns:
        X_single = X[[feature]]
        X_train, X_test, y_train, y_test = train_test_split(
                X_single, y, test_size=0.2, random_state=42
        )
        # Train a logistic regression model
        model = LogisticRegression(max_iter=1000)
        model.fit(X_train, y_train)
        # grab the accuracy
        acc = accuracy_score(y_test, model.predict(X_test))
        results.append({'Feature': feature, 'Accuracy': acc})

results_df = pd.DataFrame(results).sort_values(by='Accuracy', ascending=False)

# output the results dataframe
return results_df,
```