

Аналіз вимірювань акселерометра та гіроскопа для різних видів фізичних властивостей

Постановка задачі:

- 1) Дозбирати датасет (спробувати промодельювати (вік, висоту людини і т.п.) чи назбирати оригінальні дані);
- 2) Збір даних бажано не менше 100 Гц (50);
- 3) Збір даних за допомогою різних додатків чи заліза;
- 4) Зібраний датасет промаркувати та поділити на вибірки тренувальні та тестувальні (70/30);
- 5) Датасети поділити на вікна тривалістю N (1..5 с);
- 6) Порахувати статистичні параметри осей в межах вікна та представити розподіли даних параметрів;
- 7) Визначити потенційно головні осі та параметри, які будуть відрізняти один клас активностей від інших;
- 8) Побудувати алгоритм, який буде класифікувати типи активностей лише на базі статистичних параметрів;
- 9) Протестувати алгоритм на базі тестової вибірки;
- 10) Представити чисельні результати;
- 11) Знайти оптимальну тривалість вікна, частоти (downsampling), при якій буде найвища точність;
- 12) Побудувати confusion matrix між класами.

Етап 1: Збір даних

Дані було зібрано з допомогою програми Physics Toolbox Suite.

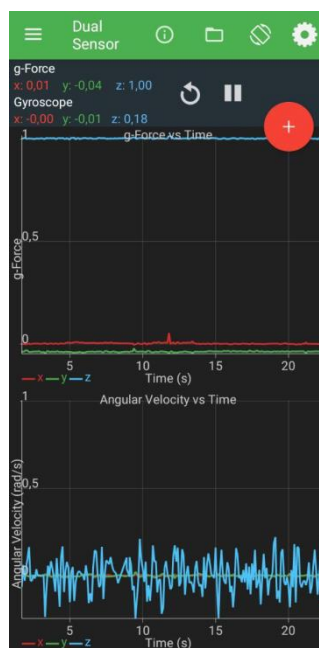


Рис. 1. Запис даних акселерометра та гіроскопа

Було здійснено спробу дізнатися, чи можна використовувати додаток для комерційних цілей, використовуючи Play Маркет.

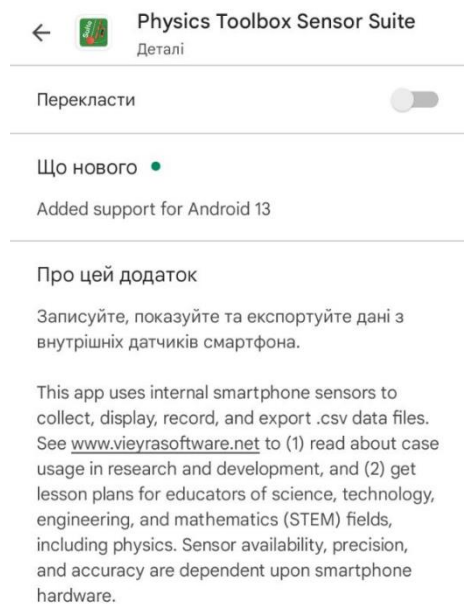


Рис. 2. Інформація про додаток у Play Маркет

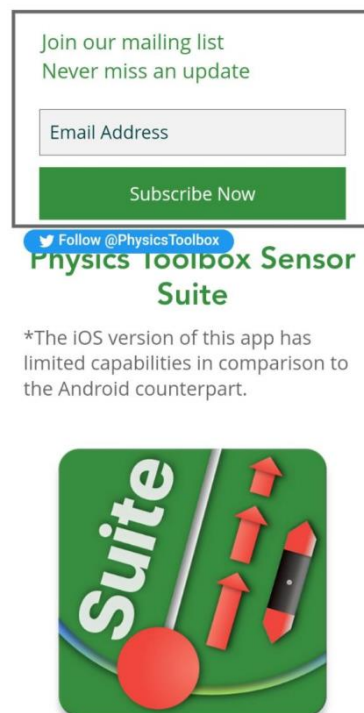


Рис. 3. Інформація про додаток на офіційному сайті
(<https://www.vieyrasoftware.net/>)

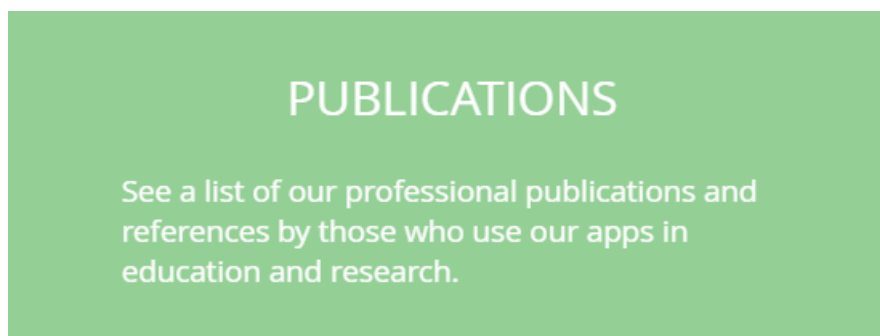


Рис. 4. Інформація про додатки розробників на сайті додатку (<https://www.vieyrasoftware.net/>)

Із результатів, наведених на рис. 2 – 4 можна зробити висновок, що додаток можна використовувати у навчальних цілях та для досліджень.

Наступним кроком я переглянув інформацію про додаток Physics Toolbox Suite у самій програмі.

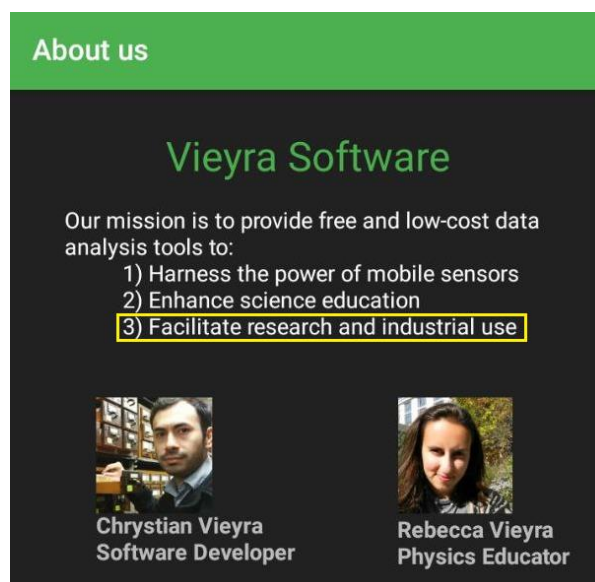


Рис. 5. Інформація про додаток у секції 'About us'

Із рис. 5 видно, що розробники додатку визначають однією з своїх місій сприяння дослідницькому та промисловому використанню програми. Однак, для 100% впевненості можна скontaktуватися безпосередньо із розробниками.

Альтернативою до програми Physics Toolbox Suite може слугувати

додаток Sensor Logger 

Під час запису даних смартфон знаходився у правій руці.

Датасет містить записи 6 різних видів активностей:

- 1) Присідання;
- 2) Випади на кожную ногу (рис. 6);
- 3) Ходьба по кімнаті із різкими поворотами на 180° (це 2 окремі види);
- 4) Стрибки із сторони в сторону (рис. 7);
- 5) Lateral Squat Slide (Переміщення центру маси з однієї ноги на іншу) (рис. 8).

Між кожним видом активності робилися паузи, щоб стабілізувати покази приладів (аксерометра та гіроскопа), забезпечити відпочинок м'язів та підготуватися до виконання наступної вправи.

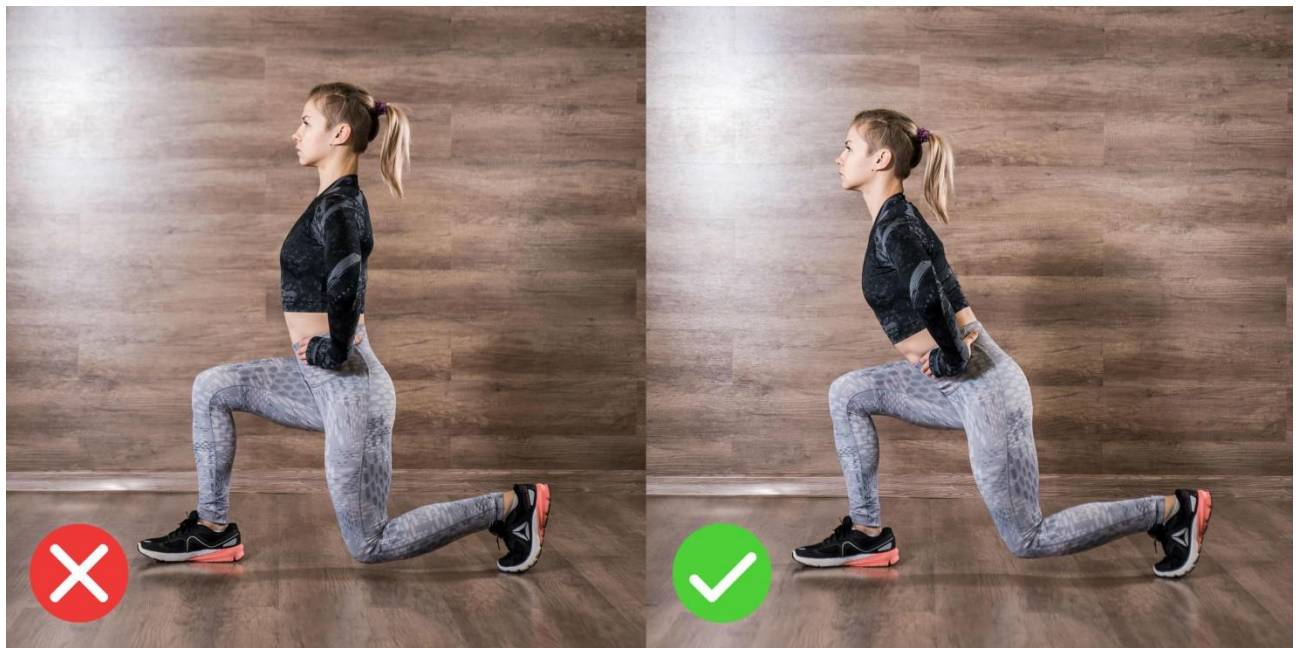


Рис. 6. Випади на кожную ногу

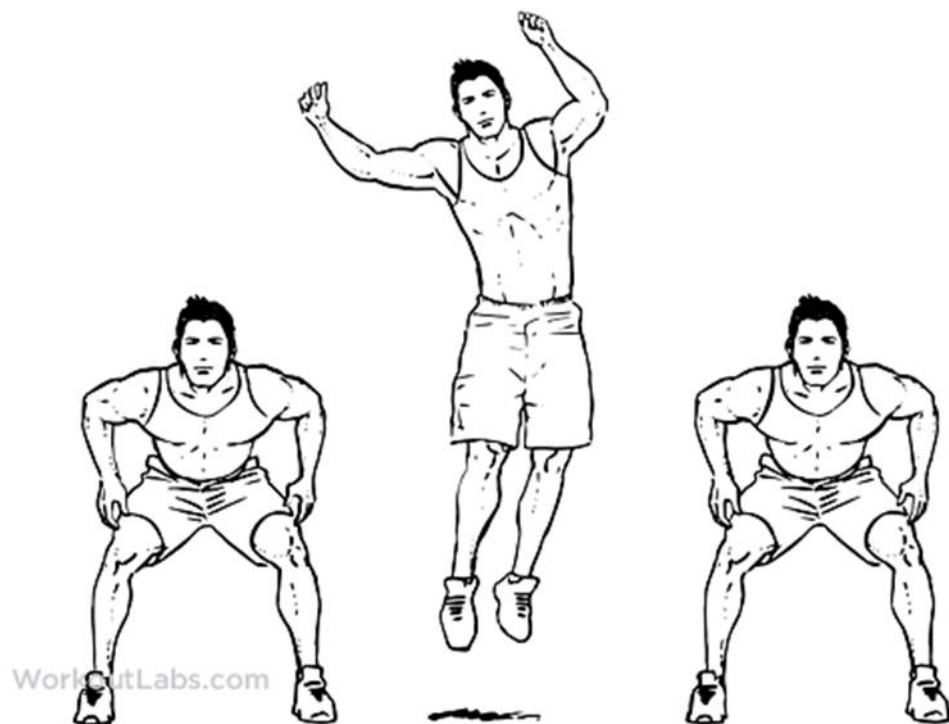


Рис. 7. Стрибки із сторони в сторону

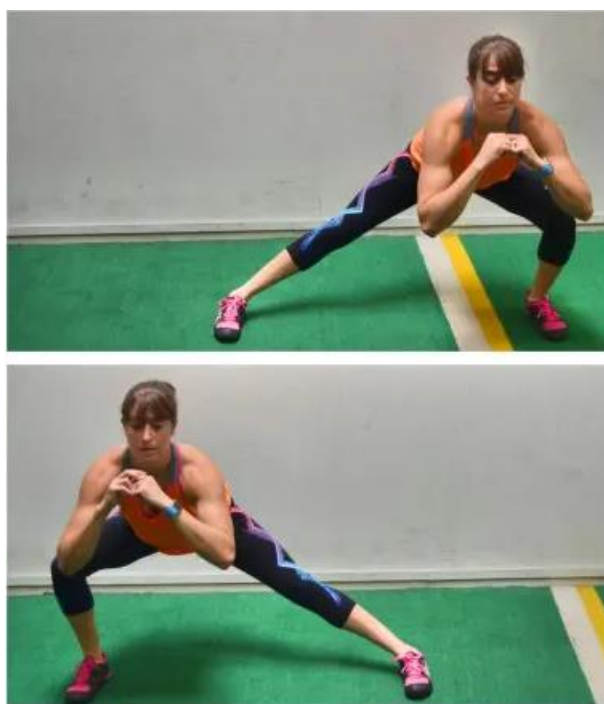


Рис. 8. Lateral squat slide

Опис датасету:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 123116 entries, 0 to 123115
Data columns (total 7 columns):
#   Column  Non-Null Count  Dtype  
---  -
0   time     123116 non-null  float64
1   gFx      123116 non-null  float64
2   gFy      123116 non-null  float64
3   gFz      123116 non-null  float64
4   wx       123116 non-null  float64
5   wy       123116 non-null  float64
6   wz       123116 non-null  float64
dtypes: float64(7)
memory usage: 6.6 MB
```

Рис. 9. Інформація про стовпці датафрейму

```
time_diffs = df['time'].diff()
freq = 1.0 / time_diffs.mean()
print(f"Measurement time = {df.iloc[-1]['time'] - df.iloc[0]['time']} s")
print(f"Number of measurements (number of rows in the data set) = {len(df)}")
print(f"Frequency of measurement = {freq} Hz")
```

```
Measurement time = 152.80100000000002 s
Number of measurements (number of rows in the data set) = 123116
Frequency of measurement = 805.721166746291 Hz
```

Рис. 10. Інформація про час та частоту збирання даних

```
df.head()
```

	time	gFx	gFy	gFz	wx	wy	wz
0	0.003	-0.01	0.02	0.98	-0.01	0.01	-0.07
1	0.004	-0.01	0.02	0.98	-0.01	0.01	-0.07
2	0.004	-0.01	0.02	0.98	-0.01	0.01	-0.07
3	0.005	-0.01	0.02	0.98	-0.01	0.01	-0.07
4	0.005	-0.00	0.01	0.98	-0.01	0.01	-0.07

Рис. 11. Вміст датафрейму

Етап 2: Data Preprocessing та маркування даних у Label Studio

Перед маркуванням даних у Label Studio здійснено перевірку стовпця 'time':

- 1) Перевірка на наявність значень ≤ 0 ;
- 2) Перевірка, чи дані стовпця розміщені у строго зростаючому порядку.

```
len(df[df['time'] <= 0])
```

0

```
# Check if the 'time' column of the dataframe is sorted in ascending order  
df['time'].is_monotonic_increasing
```

True

Рис. 12. Результати перевірки стовпця 'time'

Як бачимо, вміст досліджуваного стовпця співпадає із очікуванням. Тому можемо перейти до маркування даних у Label Studio.

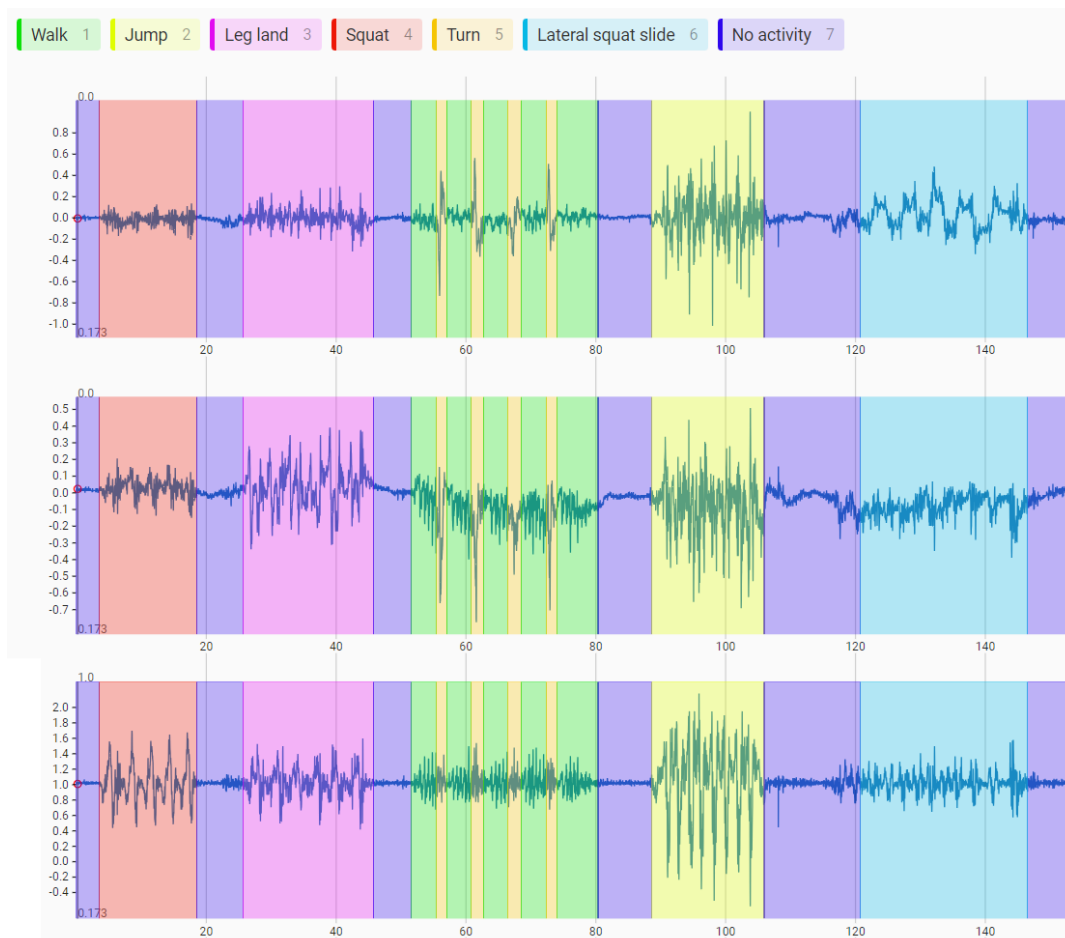


Рис. 13. Промарковані сигнали для акселерометра

Результати маркування були експортовані із Label Studio у форматі csv та додані до початкового датафрейму df у вигляді стовпця 'activity'.

```
add_activity_column(path='data/labeled_data/Test_3_labeled_2.csv', df=df, df_merge_col='time')
df.head()
```

	time	gFx	gFy	gFz	wx	wy	wz	activity
0	0.003	-0.01	0.02	0.98	-0.01	0.01	-0.07	No activity
1	0.004	-0.01	0.02	0.98	-0.01	0.01	-0.07	No activity
2	0.004	-0.01	0.02	0.98	-0.01	0.01	-0.07	No activity
3	0.005	-0.01	0.02	0.98	-0.01	0.01	-0.07	No activity
4	0.005	-0.00	0.01	0.98	-0.01	0.01	-0.07	No activity

Рис. 14. Додавання до df стовпця 'activity'

Етап 3: Data Filtering

Розглянемо результати вимірювань перед фільтруванням.

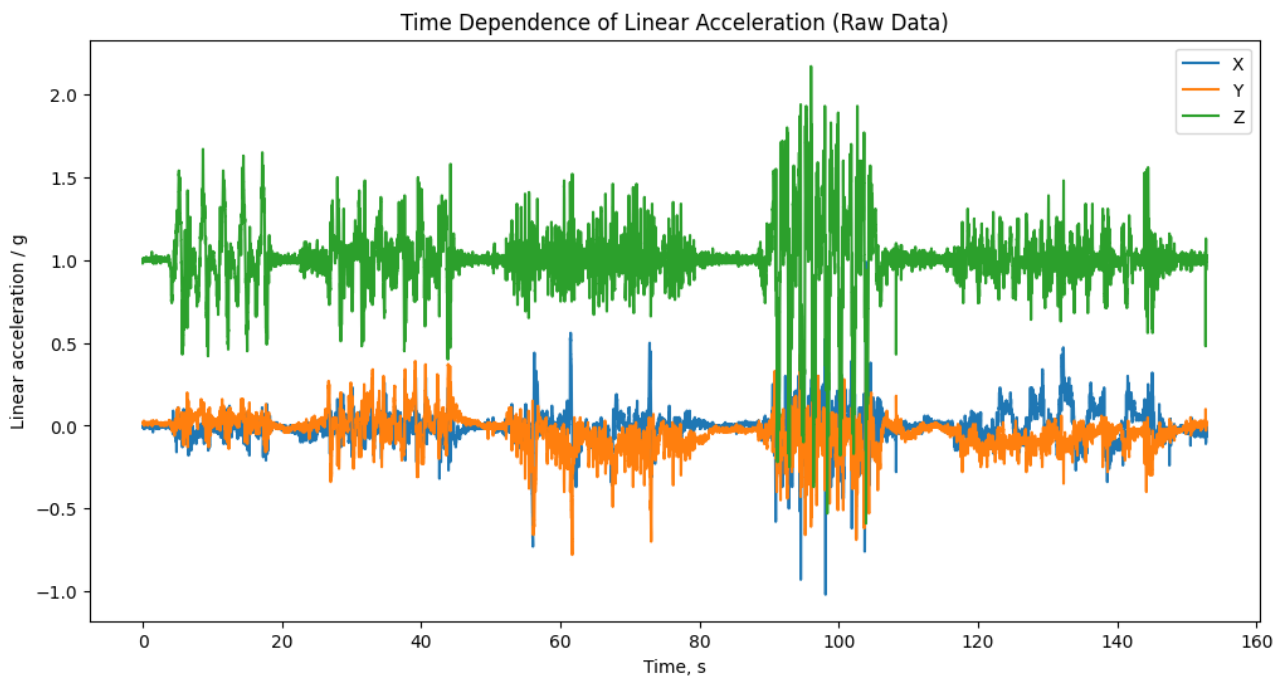


Рис. 15. Результати вимірювань акселерометра перед фільтруванням

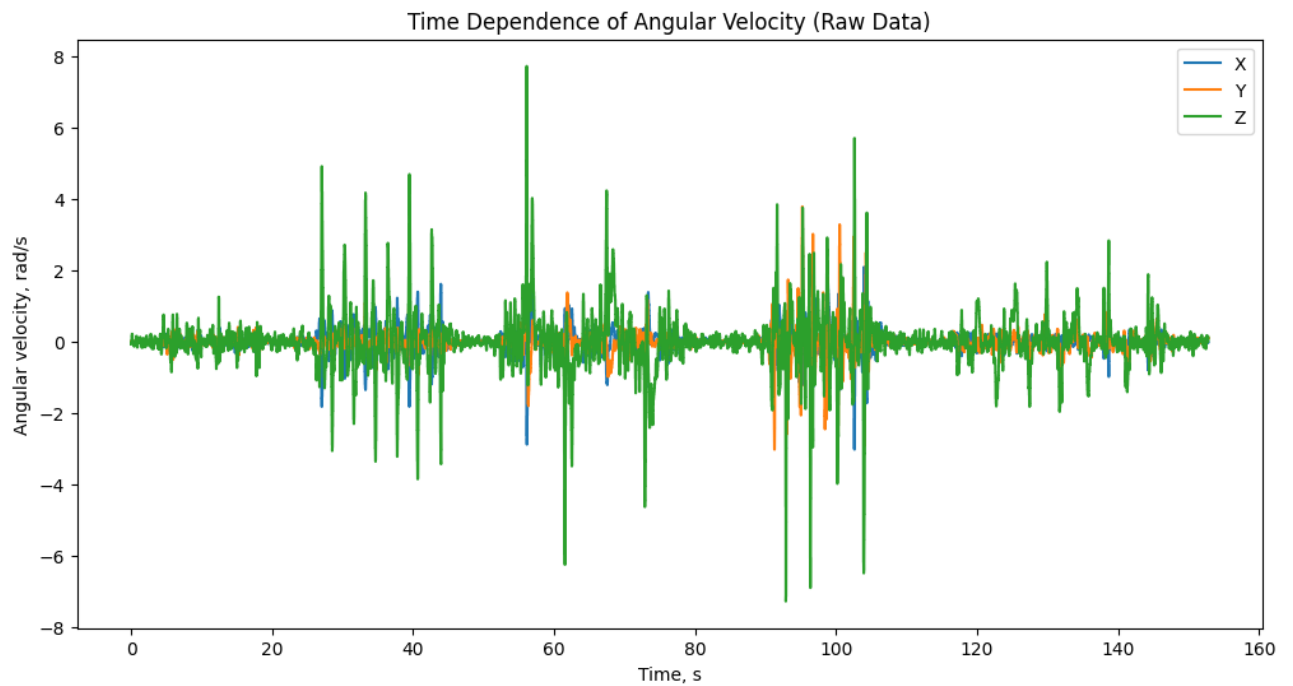


Рис. 16. Результати вимірювань гіроскопа перед фільтруванням

Для фільтрування результатів використано алгоритм ковзного середнього (moving average) із вікном 10.

Розглянемо, як виглядають графіки після фільтрування.

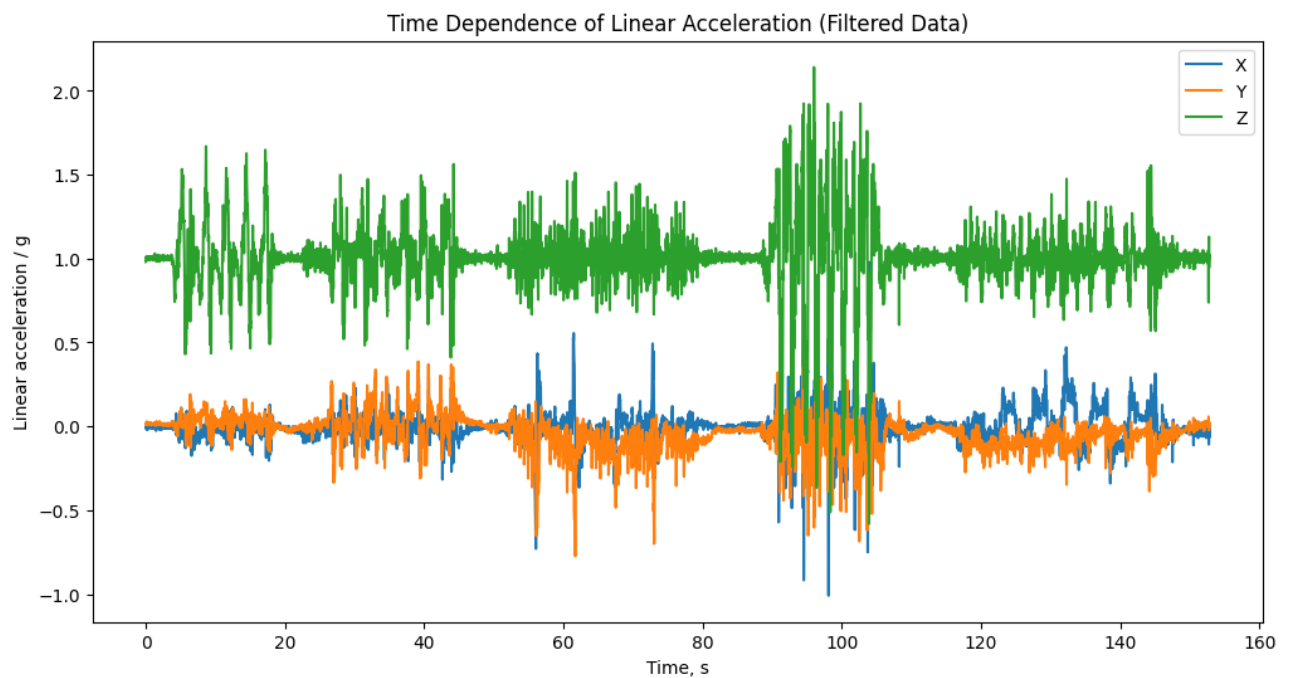


Рис. 17. Результати вимірювань акселерометра після фільтрування

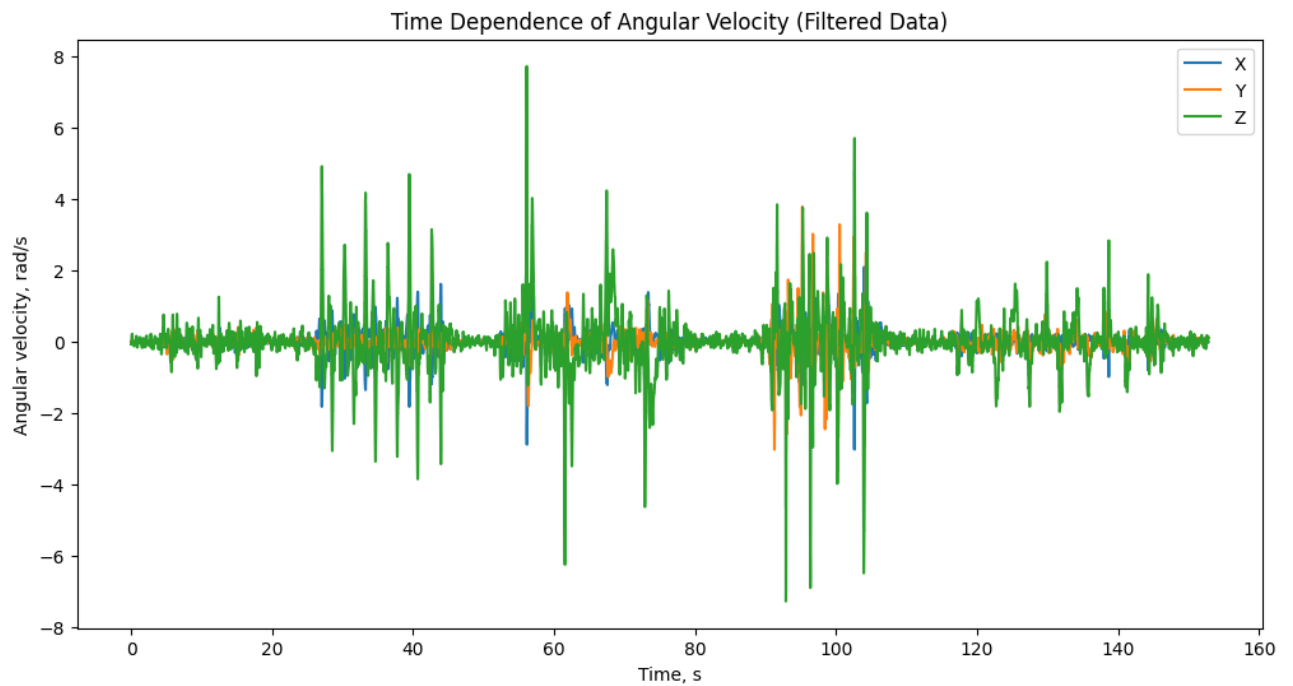


Рис. 18. Результати вимірювань гіроскопа після фільтрування

Як бачимо, фільтрування з допомогою алгоритму ковзного середнього зберегло особливості поведінки пристроїв.

На прикладі осі ОХ акселерометра розглянемо вплив фільтрування.

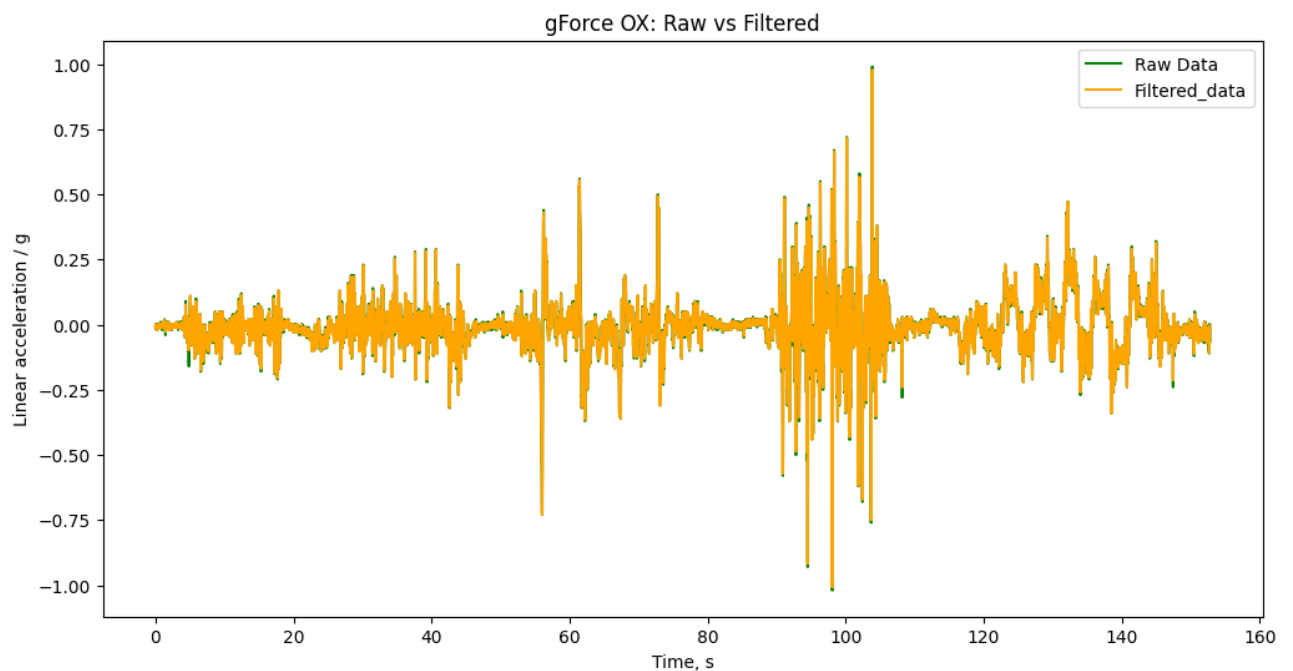


Рис. 19. Вплив фільтрування на вісь ОХ акселерометра

Із рис. 19 видно, що внаслідок фільтрування відбулося згладження деяких викидів, а результати загалом були усереднені.

Етап 4: Exploratory Data Analysis

Спочатку проаналізуємо розподіл міток видів активностей (які активності переважають у датасеті, а яких найменше).

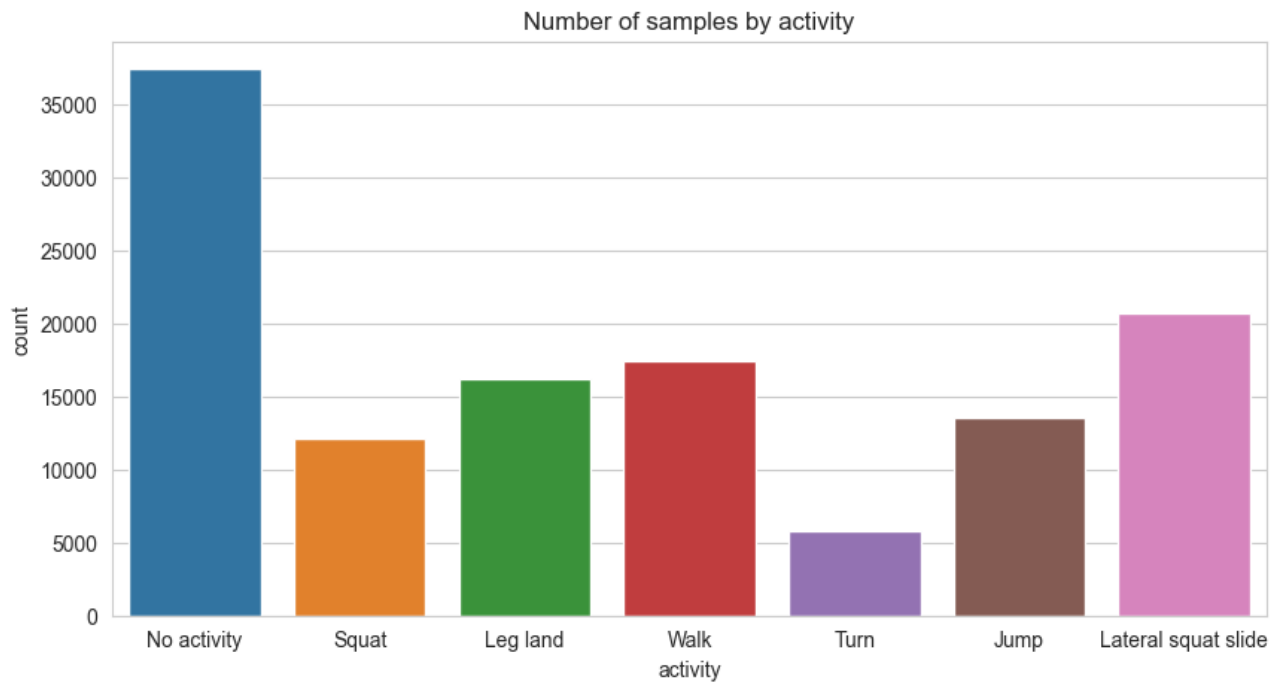


Рис. 20. Аналіз розподілу міток класів

Для кращої наглядності співвідношення між мітками класів розглянемо також кругову діаграму.

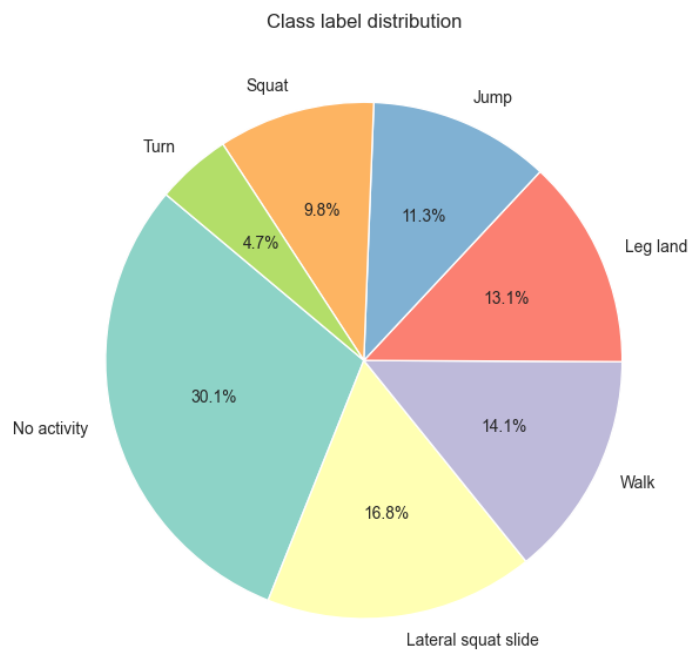


Рис. 21. Аналіз розподілу міток класів

Із рис. 20 та рис. 21 видно, що існує дисбаланс класів: найбільше даних відповідають значенню 'No activity', що цілком очікувано враховуючи перерви між кожним видом активності. Дані для активностей 'Squat', 'Leg land', 'Walk', 'Jump' та 'Lateral squat slide' розподілені +/- однаково. Найменшу частину становлять дані активності 'Turn', що теж логічно пояснюється малою тривалістю виконання повороту на 180°.

Наступним кроком побудуємо розподіл даних сигналу по осях OX, OY та OZ для того, щоб побачити, чи є якась очевидна закономірність на основі розподілу значень.

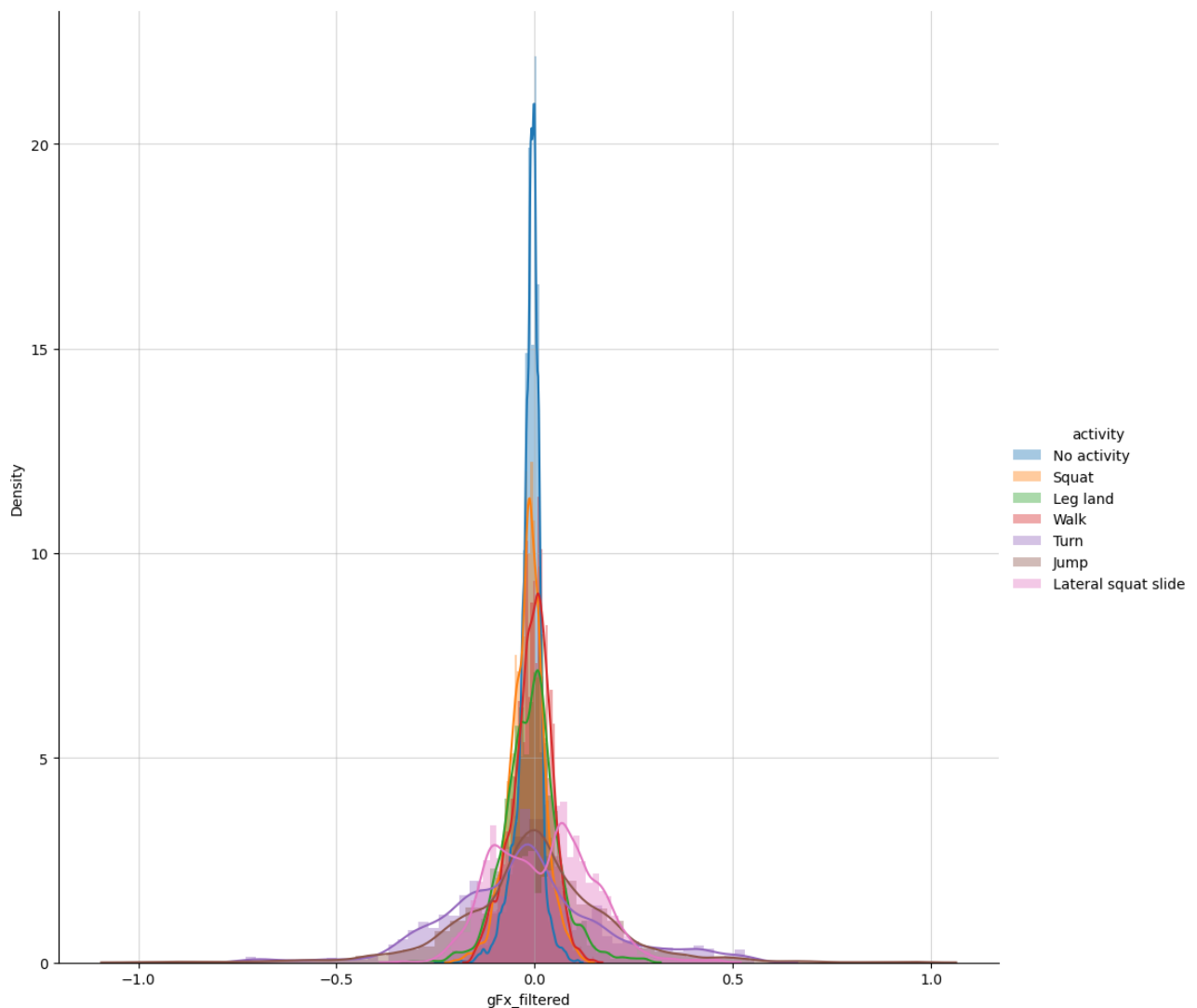


Рис. 22. Розподіл сигналів акселерометра на осі OX

Із рис. 22 можна зробити такі висновки:

- 1) Найменше розсіювання мають такі види активності: 'No activity' та 'Squat';
- 2) Незначне розсіювання значень мають 'Walk' та 'Leg land';
- 3) Найбільше розсіювання характерне для 'Jump', 'Turn' (функції густини розподілу нагадують нормальний розподіл) та 'Lateral squat slide'. Це

свідчить про наявність значних коливань на осі ОХ акселерометра для цих видів діяльності,

Зроблені висновки стосовно осі ОХ акселерометра підтверджуються результатами, зображеними на рис. 17.

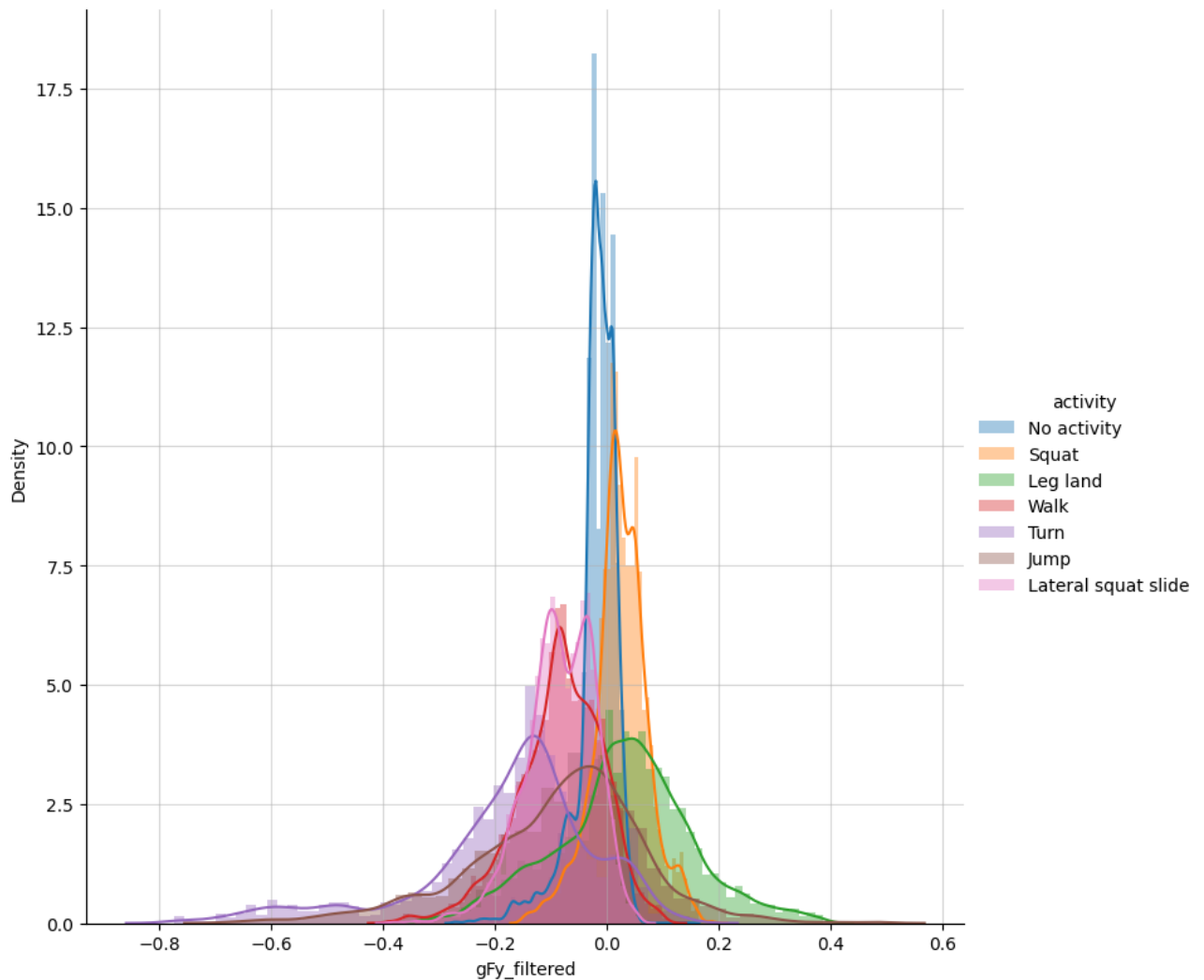


Рис. 23. Розподіл сигналів акселерометра на осі ОУ

Із рис. 23 можна зробити такі висновки:

- 1) Найменше розсіювання мають такі види активності: 'No activity' та 'Squat';
- 2) Значне розсіювання значень мають 'Walk' та 'Lateral squat slide';
- 3) Найбільше розсіювання характерне для 'Jump', 'Leg land' (функції густини розподілу нагадують нормальний розподіл) та 'Turn'.

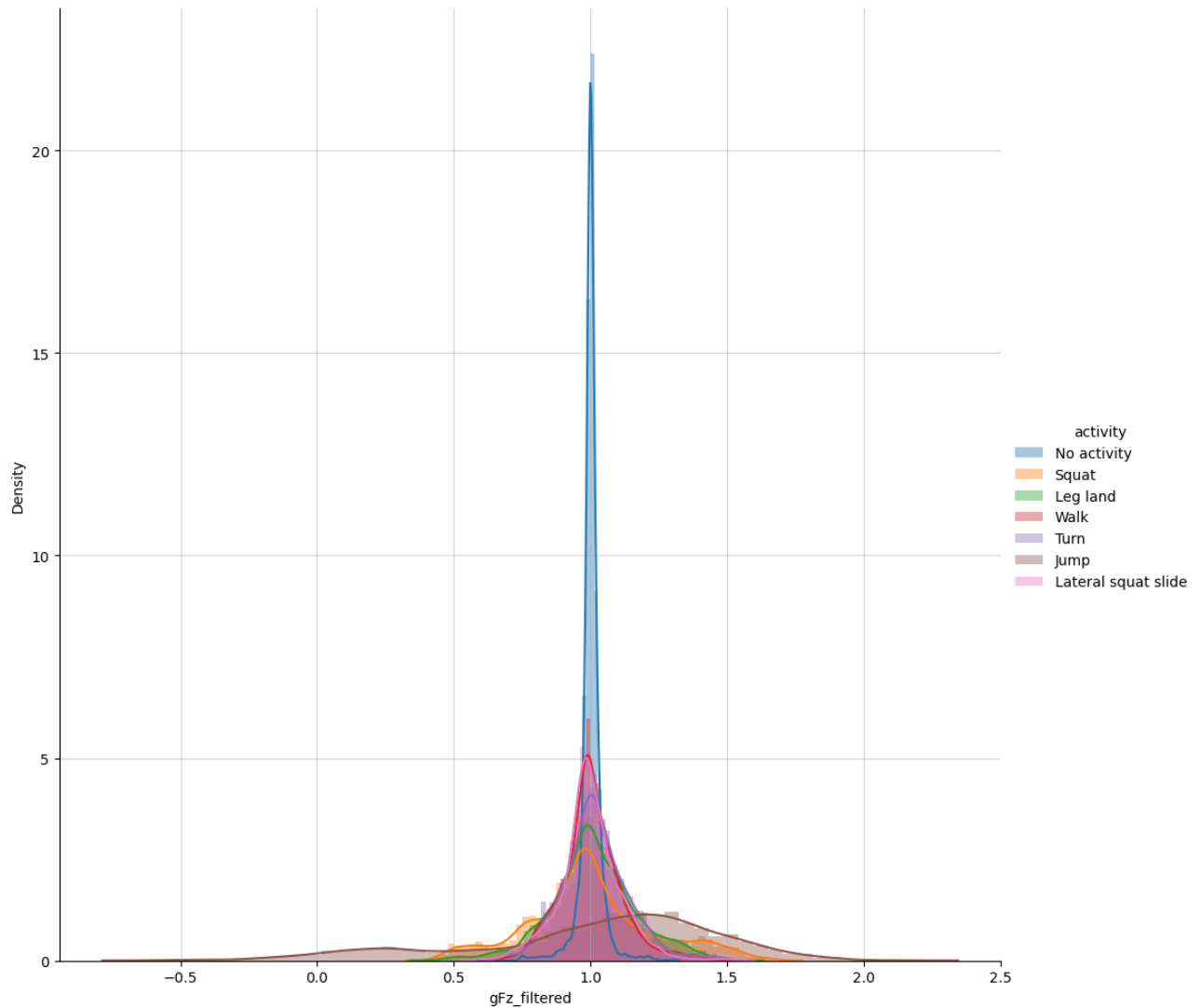


Рис. 24. Розподіл сигналів акселерометра на осі OZ

Із рис. 24 можна зробити такі висновки:

- 1) Найменше розсіювання має 'No activity';
- 2) Значне розсіювання значень мають 'Walk' та 'Turn';
- 3) Найбільше розсіювання характерне для 'Jump', а також 'Leg land' і 'Squat' (функції густини розподілу нагадують нормальний розподіл).

Координата x вершин графіків функцій густин розподілу ≈ 1 , що пояснюється наявністю прискорення вільного падіння g для осі OZ ($g=1$).

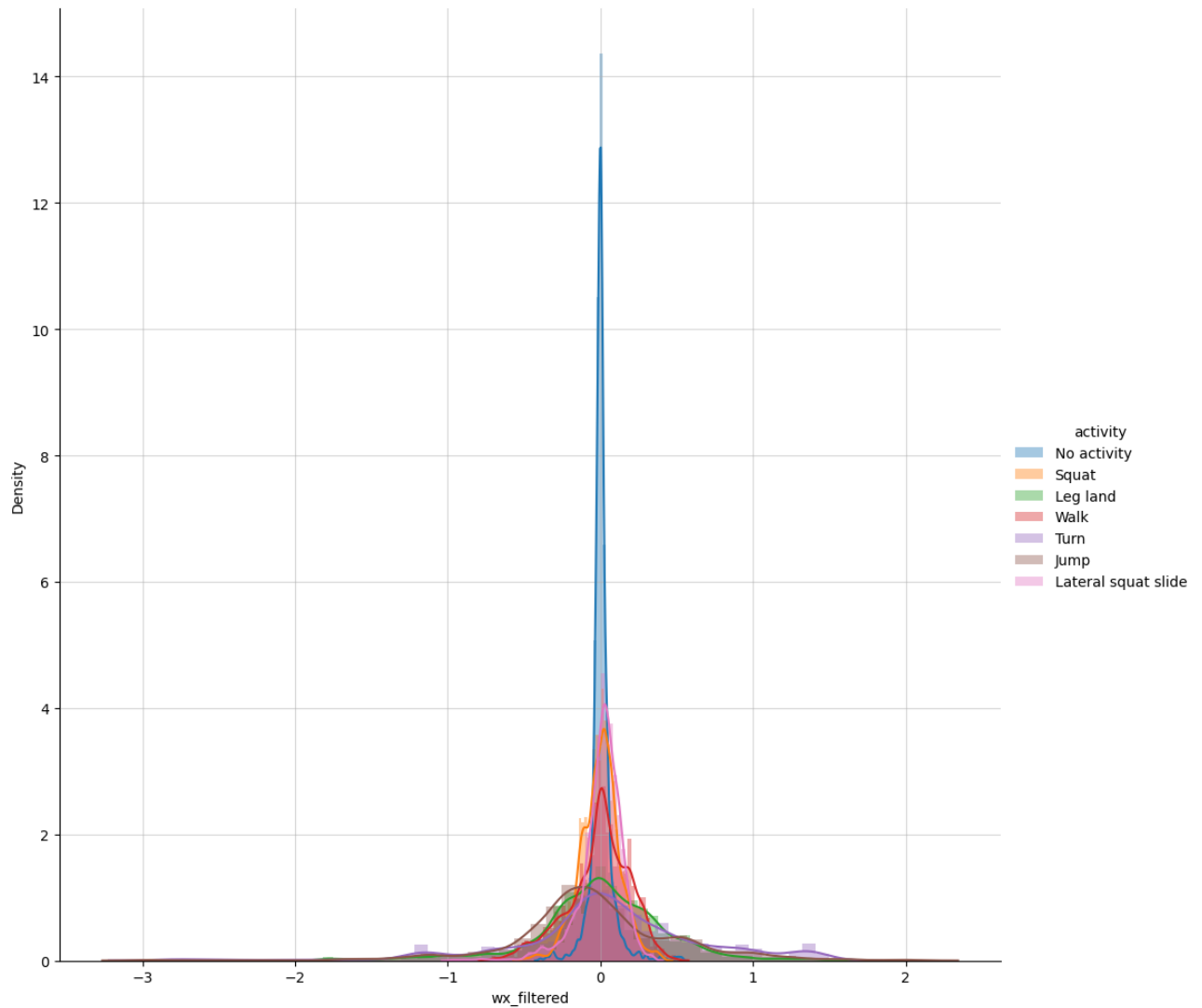


Рис. 25. Розподіл сигналів гіроскопа на осі OX

Із рис. 25 можна зробити такі висновки:

- 1) Найменше розсіювання має 'No activity';
- 2) Значне розсіювання значень мають 'Lateral squat slide', 'Squat' та 'Walk';
- 3) Найбільше розсіювання характерне для 'Jump', 'Leg land' і 'Turn' (функції густини розподілу нагадують нормальний розподіл).

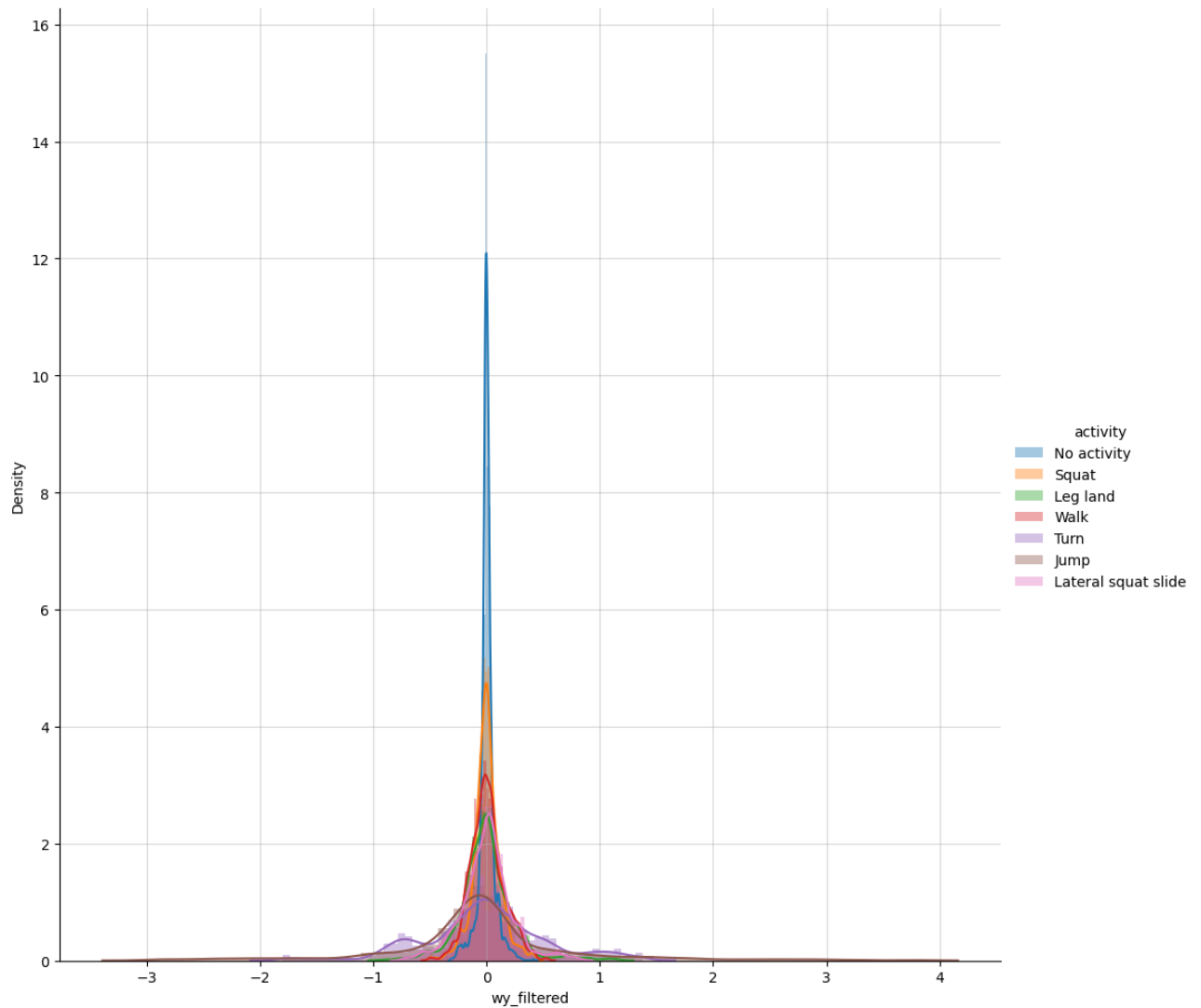


Рис. 26. Розподіл сигналів гіроскопа на осі OY

Із рис. 26 можна зробити такі висновки:

- 1) Найменше розсіювання має 'No activity';
- 2) Значне розсіювання значень мають 'Squat', 'Walk', 'Leg land' і 'Lateral squat slide';
- 3) Найбільше розсіювання характерне для 'Jump' і 'Turn' (функції густини розподілу нагадують нормальний розподіл).

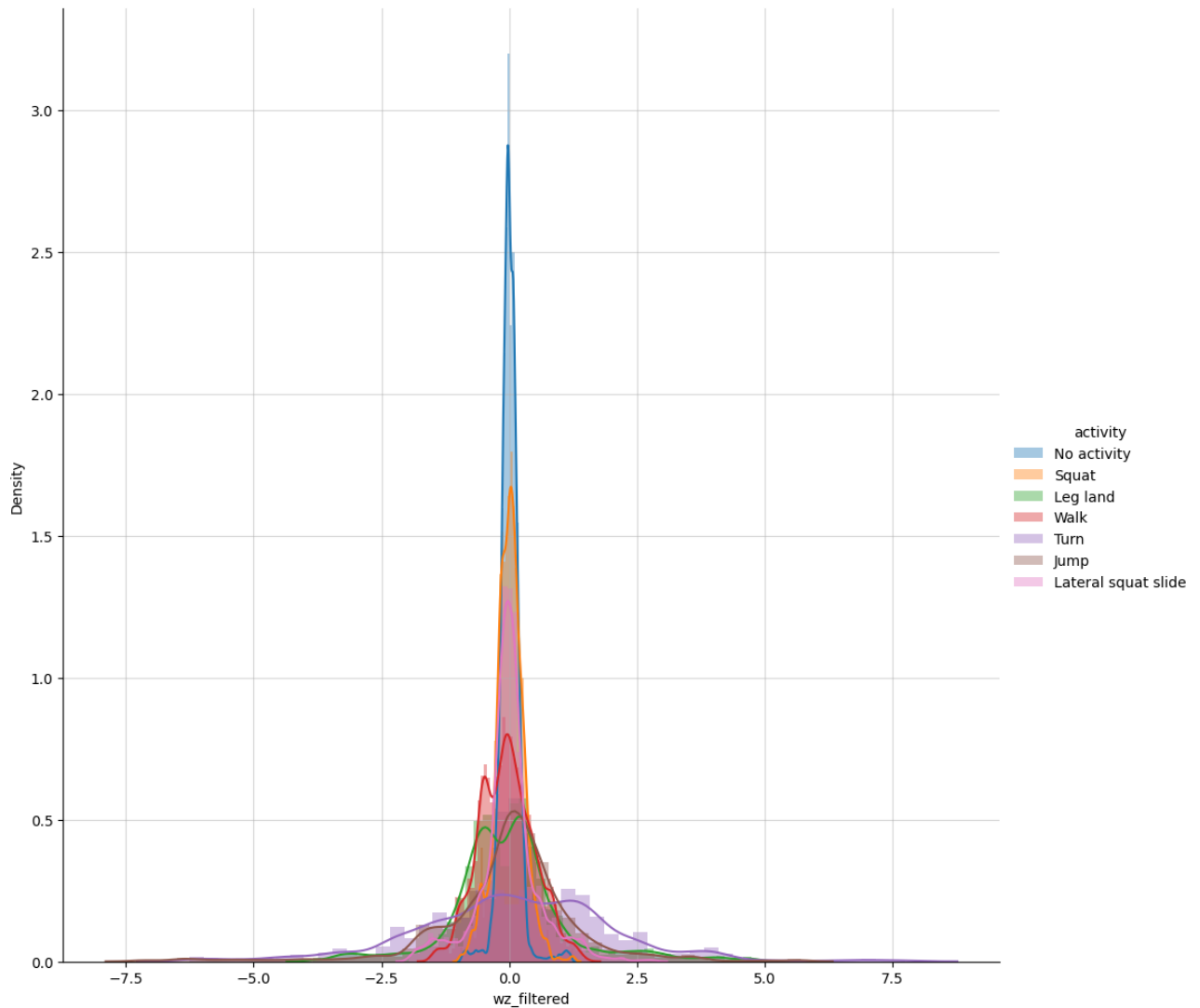


Рис. 27. Розподіл сигналів гіроскопа на осі OZ

Із рис. 27 можна зробити такі висновки:

- 1) Найменше розсіювання має 'No activity', 'Squat' і 'Lateral squat slide';
- 2) Значне розсіювання значень мають 'Walk', 'Leg land' і 'Jump';
- 3) Найбільше розсіювання характерне для 'Turn'.

Розглянувши рис. 22-27, зроблено висновки про те, що розподіли активностей дуже схожі між собою. Існує значне збігання даних між такими видами діяльності: 'Squat' і 'Leg land', 'No activity' і 'Walk'. 'Jump' вирізняється дуже великим розсіюванням на осі OZ акселерометра, а 'Turn' на осі OZ гіроскопа. Для 'Lateral squat slide' характерна наявність двох піків на осі OX акселерометра.

Етап 5: Data Transformation

Здійснено поділ оригінального датафрейму `df` на вікна тривалістю 2 с. Кількість рядків, які вміщалися в одне вікно становить:

$$\text{window_size} = \text{ceil}(\text{sampling_frequency} * \text{window_duration}),$$

де

- `sampling_frequency` – середня частота збору даних в Гц (точне її значення наведено на рис. 10);
- `window_duration` – тривалість вікна у с (у даному випадку `window_duration = 2`);
- `ceil` – функція бібліотеки `math` у Python, яка завжди заокругляє дріб до більшого цілого (у більшу сторону).

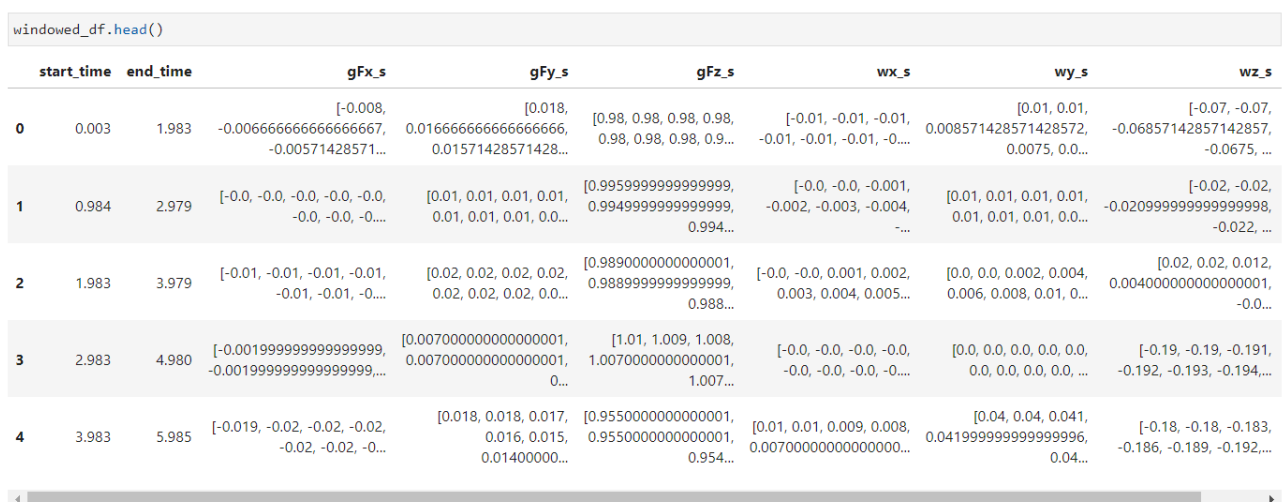
Отже,

$$\text{window_size} = \text{ceil}(805.721 * 2) = 1612$$

Варто також зазначити, що замість окремих вікон ми беремо вікна, що перекриваються, із 50% перекриттям (тобто крок = `window_size / 2 = 806`). Це гарантує, що кожен наступний рядок у перетвореному наборі даних також містить певну інформацію з даних у попередньому вікні.

Для призначення мітки класу активності береться найчастіша активність у цьому вікні.

В результаті виконання цієї операції отримуємо новий датафрейм – `windowed_df`.



	start_time	end_time	gFx_s	gFy_s	gFz_s	wx_s	wy_s	wz_s
0	0.003	1.983	[-0.008, -0.006666666666666667, -0.00571428571...]	[0.018, 0.016666666666666666, 0.01571428571428...]	[0.98, 0.98, 0.98, 0.98, 0.98, 0.98, 0.98, 0.9...]	[-0.01, -0.01, -0.01, -0.01, -0.01, -0.01, -0.01, -0.0...]	[0.01, 0.01, 0.008571428571428572, -0.06857142857142857, 0.0075, 0.0...]	[-0.07, -0.07, -0.0675, ...]
1	0.984	2.979	[-0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0...]	[0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.0...]	[0.9959999999999999, 0.9949999999999999, 0.994...]	[-0.0, -0.0, -0.001, -0.002, -0.003, -0.004, -...]	[0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.0...]	[-0.02, -0.02, -0.020999999999999998, -0.022, ...]
2	1.983	3.979	[-0.01, -0.01, -0.01, -0.01, -0.01, -0.01, -0.01, -0.0...]	[0.02, 0.02, 0.02, 0.02, 0.02, 0.02, 0.02, 0.0...]	[0.9890000000000001, 0.9889999999999999, 0.988...]	[-0.0, -0.0, 0.001, 0.002, 0.003, 0.004, 0.005...]	[0.0, 0.0, 0.002, 0.004, 0.006, 0.008, 0.01, 0...]	[0.02, 0.02, 0.012, 0.004000000000000001, -0.0...]
3	2.983	4.980	[-0.0019999999999999999, -0.0019999999999999999, ...]	[0.007000000000000001, 0.007000000000000001, 0...]	[1.01, 1.009, 1.008, 1.0070000000000001, 1.007...]	[-0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0...]	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0...]	[-0.19, -0.19, -0.191, -0.192, -0.193, -0.194...]
4	3.983	5.985	[-0.019, -0.02, -0.02, -0.02, -0.02, -0.02, -0.02, -0...]	[0.018, 0.018, 0.017, 0.016, 0.015, 0.01400000...]	[0.9550000000000001, 0.9550000000000001, 0.954...]	[0.01, 0.01, 0.009, 0.008, 0.00700000000000000...]	[0.04, 0.04, 0.041, 0.04199999999999996, 0.04...]	[-0.18, -0.18, -0.183, -0.186, -0.189, -0.192...]

Рис. 28. Вміст датафрейму `windowed_df`

Таким чином, з оригінального датафрейму `df`, який містив 123116 рядків ми отримали `windowed_df` довжиною 153 рядки.

```
print(f'len(df) = {len(df)}')
print(f'len(windowed_df) = {len(windowed_df)}')

len(df) = 123116
len(windowed_df) = 153
```

Рис. 29. Довжини датафреймів df та windowed_df

Розглянемо, чи не змінилося співвідношення між класами діяльності внаслідок виконання процесу віконування (windowing).

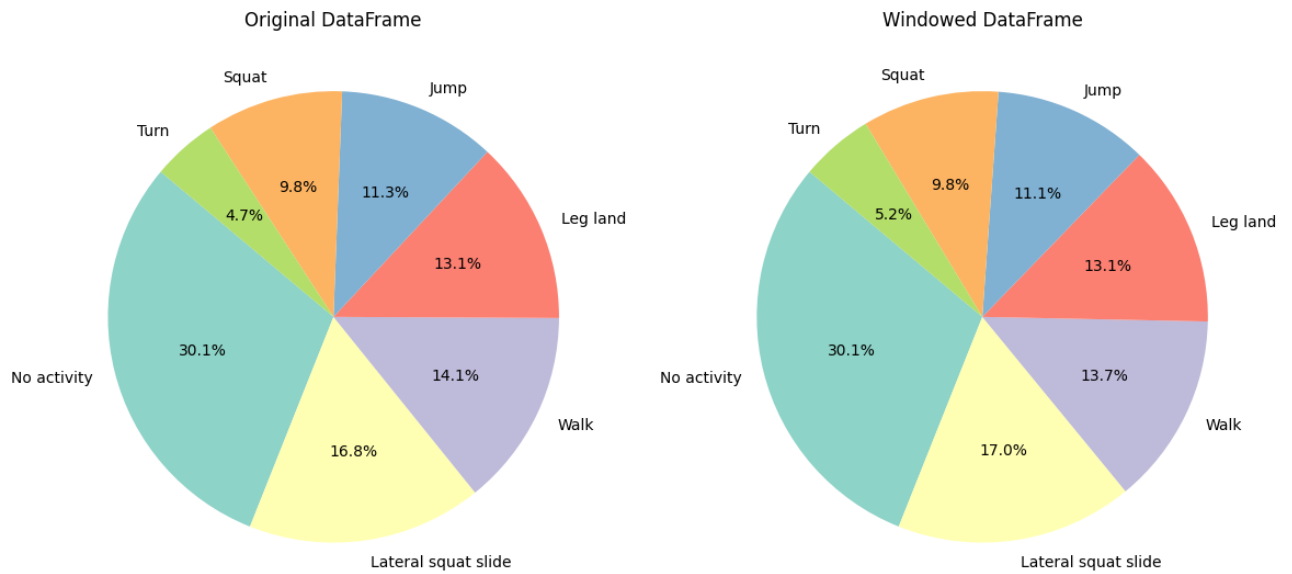


Рис. 30. Класи активності в df та windowed_df

На рис. 30 можна побачити, що процентне відношення між різними видами активності майже зберігається у windowed_df.

Наступним кроком поділимо windowed_df на train_df та test_df у відношенні 70:30 та знову перевіримо класи активностей в обох датасетах, щоб впевнитися в тому, що всі види активності потрапляють як в тренувальний датасет, так і в тестовий.

```
print(f"len(train_df) = {len(train_df)}")
print(f"len(test_df) = {len(test_df)}")

len(train_df) = 110
len(test_df) = 43
```

Рис. 31. Довжини train_df та test_df

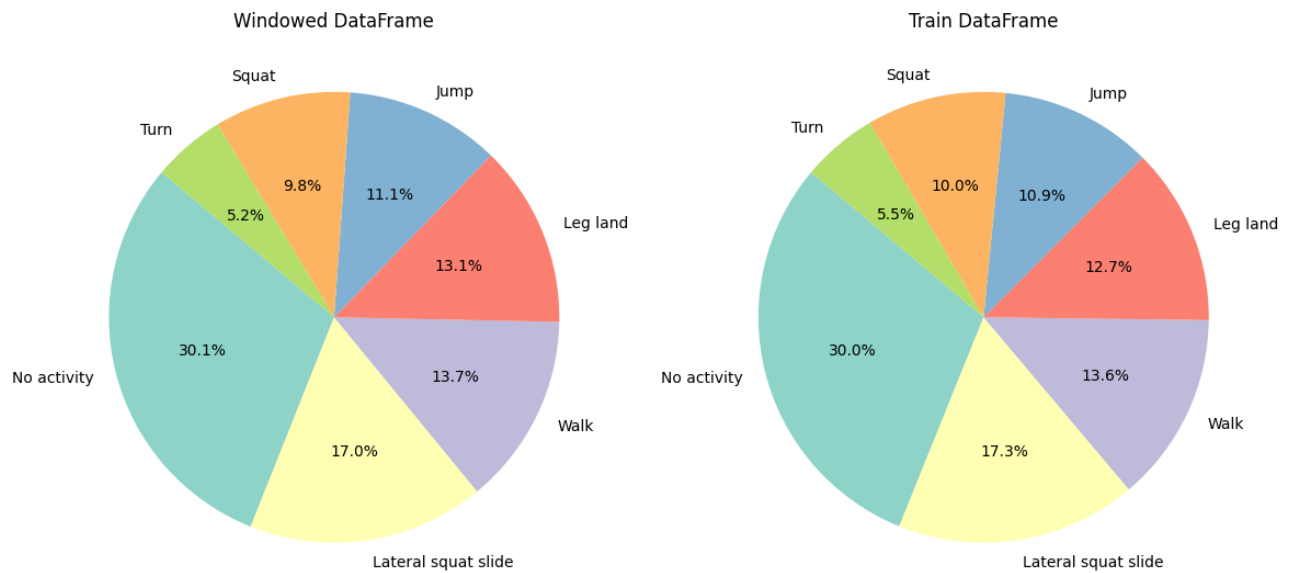


Рис. 32. Класи активності в windowed_df та train_df

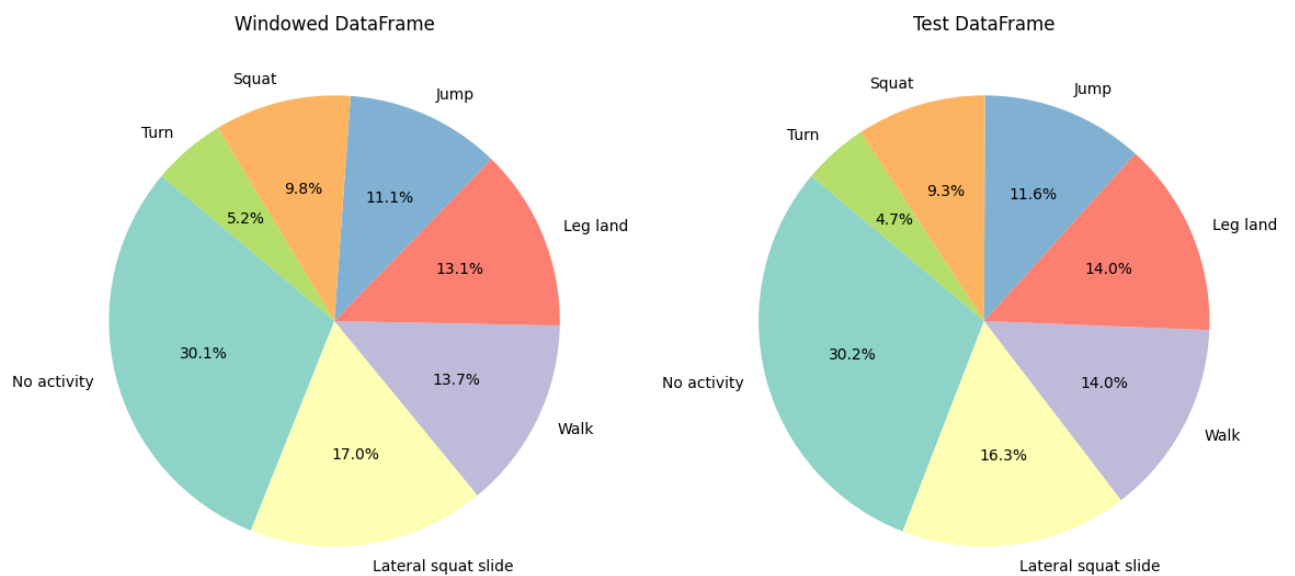


Рис. 33. Класи активності в windowed_df та test_df

Як бачимо із рис. 32 і рис. 33, процентне відношення між класами активностей зберігається як в train_df, так і в test_df.

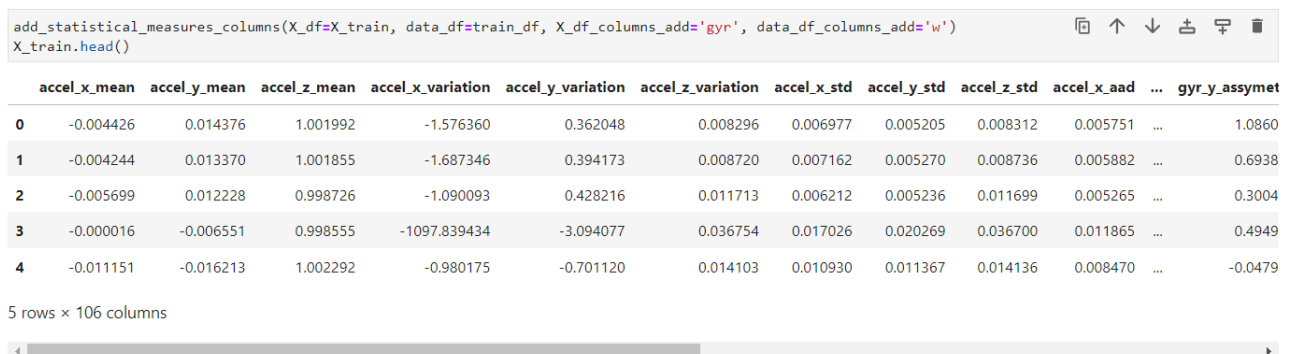
Етап 6: Feature Engineering

Для тренування моделі було виділено 19 простих статистичних параметрів (features):

1. середнє статистичне (mean);
2. стандартне відхилення (standard deviation);
3. варіація (variation);
4. середнє абсолютне відхилення (average absolute deviation);
5. мінімальне значення (min);
6. максимальне значення (max);
7. розмах = різниця максимального і мінімального значень (range);
8. медіана (median);
9. медіанне абсолютне відхилення (median absolute deviation);
10. міжквартильний діапазон (interquartile range = IQR);
11. кількість від'ємних значень (negative values count);
12. кількість позитивних значень (positive values count);
13. кількість значень вище середнього (number of values above mean);
14. кількість вершин (number of peaks);
15. асиметрія (skewness = asymmetry);
16. ексцес (kurtosis);
17. енергія (energy);
18. середнє результуюче прискорення/кутова швидкість (average resultant acceleration/angular velocity);
19. область величини сигналу (signal magnitude area).

Значення вказаних статистичних параметрів для кожного вікна зберігаються у датафреймах `X_train` та `X_test`.

В результаті обчислення всіх вищевказаних статистичних параметрів для трьох осей акселерометра та гіроскопа, `X_train` та `X_test` містять по 106 стовпців.



	accel_x_mean	accel_y_mean	accel_z_mean	accel_x_variation	accel_y_variation	accel_z_variation	accel_x_std	accel_y_std	accel_z_std	accel_x_aad	...	gyr_y_assymet
0	-0.004426	0.014376	1.001992	-1.576360	0.362048	0.008296	0.006977	0.005205	0.008312	0.005751	...	1.0860
1	-0.004244	0.013370	1.001855	-1.687346	0.394173	0.008720	0.007162	0.005270	0.008736	0.005882	...	0.6938
2	-0.005699	0.012228	0.998726	-1.090093	0.428216	0.011713	0.006212	0.005236	0.011699	0.005265	...	0.3004
3	-0.000016	-0.006551	0.998555	-1097.839434	-3.094077	0.036754	0.017026	0.020269	0.036700	0.011865	...	0.4949
4	-0.011151	-0.016213	1.002292	-0.980175	-0.701120	0.014103	0.010930	0.011367	0.014136	0.008470	...	-0.0479

5 rows × 106 columns

Рис. 34. Вміст датафрейму `X_train`

Відповідні мітки класів активностей містяться у масивах `y_train` та `y_test`.

Етап 7: Model Training

Модель Logistic Regression тренуватиметься на даних X_{train} та мітках y_{train} , а її ефективність оцінюватиметься на X_{test} .

Результат тренування моделі із використанням лише статистичних параметрів наведено нижче.

Accuracy: 0.7441860465116279

-----Classification Report-----

	precision	recall	f1-score	support
Jump	1.00	1.00	1.00	5
Lateral squat slide	0.75	0.86	0.80	7
Leg land	1.00	0.83	0.91	6
No activity	0.80	0.62	0.70	13
Squat	0.80	1.00	0.89	4
Turn	1.00	0.50	0.67	2
Walk	0.33	0.50	0.40	6
accuracy			0.74	43
macro avg	0.81	0.76	0.77	43
weighted avg	0.79	0.74	0.75	43

Рис. 35. Оцінка ефективності моделі

Як видно із рис. 35, вдалося отримати загальну точність 74.42 % на даних тестування, використовуючи модель базової логістичної регресії для розроблених функцій.

Розглянемо матрицю плутанини (confusion matrix), щоб побачити, які види активності сплутала модель.

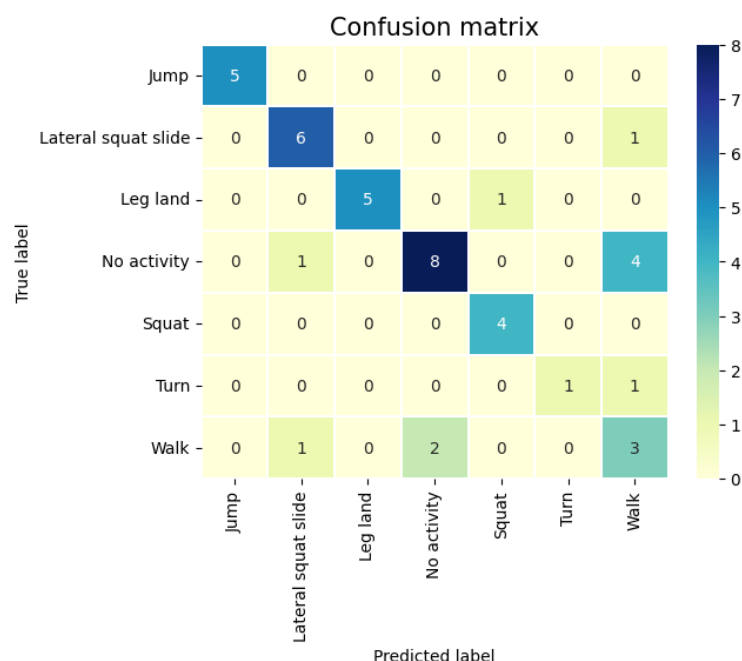


Рис. 36. Матриця плутанини

Також було здійснено спробу збільшити кількість фіч (features) за рахунок використання швидкого перетворення Фур'є.

Зокрема, було визначено 17 статистичних параметрів для результатів швидкого перетворення Фур'є:

1. середнє статистичне (mean);
2. стандартне відхилення (standard deviation);
3. варіація (variation);
4. середнє абсолютне відхилення (average absolute deviation);
5. мінімальне значення (min);
6. максимальне значення (max);
7. розмах = різниця максимального і мінімального значень (range);
8. медіана (median);
9. медіанне абсолютне відхилення (median absolute deviation);
10. міжквартильний діапазон (interquartile range = IQR);
11. кількість значень вище середнього (number of values above mean);
12. кількість вершин (number of peaks);
13. асиметрія (skewness = assymetry);
14. ексцес (kurtosis);
15. енергія (energy);
16. середнє результуюче прискорення/кутова швидкість (average resultant acceleration/angular velocity);
17. область величини сигналу (signal magnitude area).

Це дозволило збільшити кількість стовпців тренувального та тестового датасетів до 200.

```
add_fft_columns(X_df=X_train, data_df=train_df, X_df_columns_add='gyr', window_size=window_size, data_df_columns_add='w')
X_train.head()
```

fft	gyr_z_assymetry_fft	gyr_x_kurtosis_fft	gyr_y_kurtosis_fft	gyr_z_kurtosis_fft	gyr_x_energy_fft	gyr_y_energy_fft	gyr_z_energy_fft	gyr_avg_result_accl_fft	gyr_sma_fft
20	10.765238	95.238096	149.982779	126.062966	2.804934	4.200630	227.761378	0.603819	12.121368
57	9.963570	88.918279	136.937446	111.697192	4.623672	8.002463	216.677071	0.718098	14.498933
65	11.273881	178.364609	179.421930	155.924163	5.097205	8.386235	288.059433	0.818817	16.396718
25	8.851530	81.646329	82.273557	82.840466	272.586872	347.473872	1089.571546	2.061181	52.828103
40	12.530712	316.544845	127.576236	183.664460	18.406052	9.585608	520.230881	0.854412	17.863415

len(X_train.columns) must be equal to 200 (added 47 new columns)

Рис. 37. Вміст датафрейму X_train

Однак точність моделі збільшення кількості фіч не покращило:

Accuracy: 0.7209302325581395

-----Classification Report-----

	precision	recall	f1-score	support
Jump	0.75	0.60	0.67	5
Lateral squat slide	0.75	0.86	0.80	7
Leg land	0.75	1.00	0.86	6
No activity	0.75	0.69	0.72	13
Squat	1.00	1.00	1.00	4
Turn	0.00	0.00	0.00	2
Walk	0.43	0.50	0.46	6
accuracy			0.72	43
macro avg	0.63	0.66	0.64	43
weighted avg	0.69	0.72	0.70	43

Рис. 38. Оцінка ефективності моделі

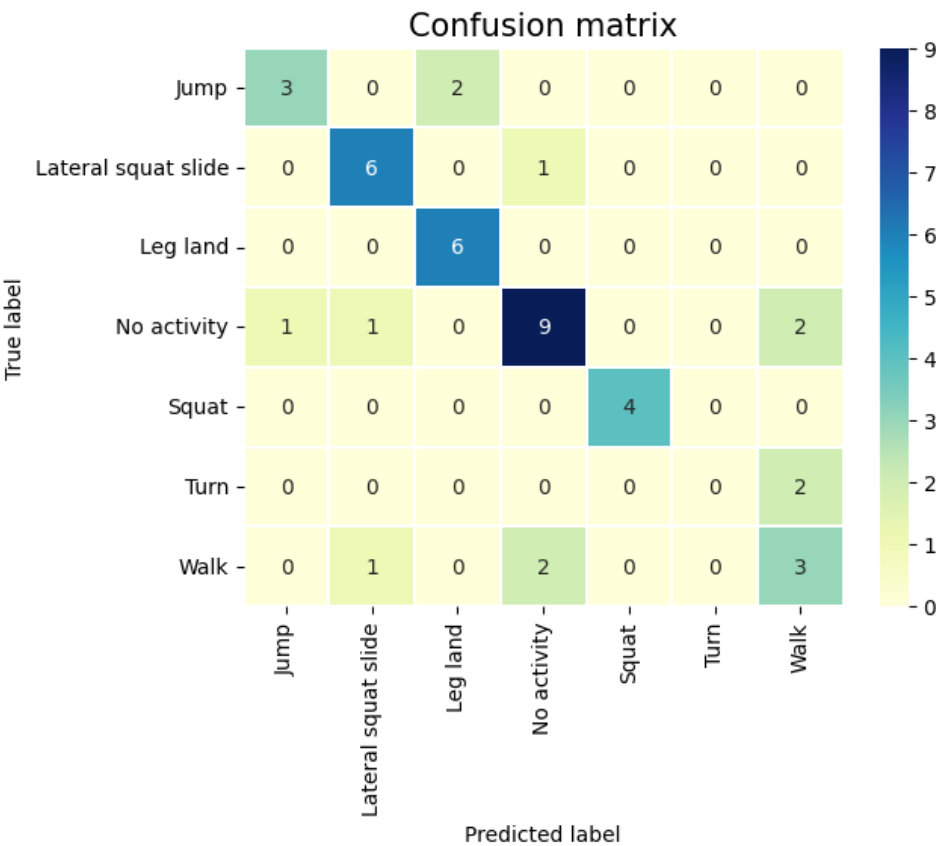


Рис. 39. Матриця плутанини

Збільшення вікна до 3 с незначно покращило результати моделі (вразовані лише статистичні параметри без швидкого перетворення Фур'є – 106 стовпців в X_train):

Accuracy: 0.75

-----Classification Report-----

	precision	recall	f1-score	support
Jump	1.00	1.00	1.00	3
Lateral squat slide	0.67	0.80	0.73	5
Leg land	1.00	0.67	0.80	3
No activity	0.78	0.78	0.78	9
Squat	1.00	1.00	1.00	3
Turn	0.00	0.00	0.00	1
Walk	0.40	0.50	0.44	4
accuracy			0.75	28
macro avg	0.69	0.68	0.68	28
weighted avg	0.75	0.75	0.74	28

Рис. 40. Оцінка ефективності моделі

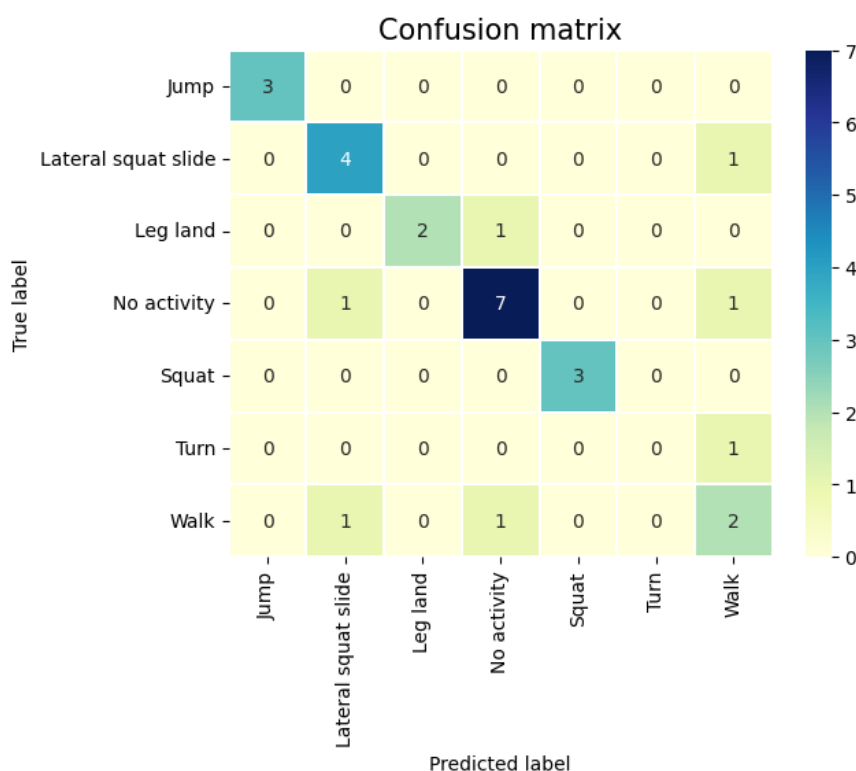


Рис. 41. Матриця плутанини

Висновки

На власний смартфон було зібрано датасет із даними акселерометра (g-Force) та гіроскопа для таких видів активностей:

- 1) Присідання;
- 2) Випади на кожную ногу;
- 3) Ходьба по кімнаті із різкими поворотами на 180°;
- 4) Стрибки із сторони в сторону;
- 5) Lateral Squat Slide (Переміщення центру маси з однієї ноги на іншу).

Зібраний датасет містить 123116 рядків із даними акселерометра та гіроскопа. Середня частота вимірювання становила 805.721 Гц, вимірювання проводилися протягом 153 с (2.5 хв).

Процентне відношення між класами активностей оригінального датасету таке:

- 1) Присідання – 9.8%;
- 2) Стрибки зі сторони в сторону – 11.3%;
- 3) Випади на ноги – 13.1%;
- 4) Ходьба – 14.1%;
- 5) Повороти на 180° – 4.7%;
- 6) Lateral squat slide – 16.8%;
- 7) Відсутність активності (No activity) – 30.1%.

Оригінальний датасет спочатку було поділено на вікна тривалістю 2 с, що дозволило з 123116 рядків оригінального датафрейму отримати 153 рядки `windowed_df`. Потім `windowed_df` було поділено на тренувальний і тестовий датасети у відношенні 70:30. Процентне відношення між видами активності для тренувального і тестового датасетів збереглося.

На етапі Feature Engineering виділено 19 статистичних параметрів, в результаті чого отримано 106 фіч (features) для тренування моделі машинного навчання.

Модель Logistic Regression показала точність 74.42 % на даних тестування (вікно тривалістю 2 с). Найчастіше модель плутала ходьбу ('Walk') та відсутність активності ('No activity'). Також модель один раз сплутала присідання ('Squat') та випади на ноги ('Leg land'), що могло бути пов'язаним іще неправильною технікою виконання присідання (я нагнув спину вперед під час присідання). Похибки моделі також можна пояснити колізіями під час поділу датафрейму на вікна, а також доволі малим обсягом тренувальних даних (110 елементів).

Модель Logistic Regression показала точність 75 % на даних тестування (вікно тривалістю 3 с). Найчастіше модель неправильно визначала активність 'Walk'.