<u>Practical Deep Learning Workshop assignments 2</u>

1) a.

i. The dataset is tabular data representing time-series measurements from wearable sensors. It contains acceleration data (x, y, z) collected for various activities performed by individuals. Each record corresponds to a specific activity, labeled for classification tasks, with metadata like the user ID, sensor type, body part, and side.

ii. The data is heterogeneous, varying in several ways:
Sensors: Data is recorded using 2 different sensor types (smartwatch and vicon).
Users: Data is collected from 8 distinct individuals, each contributing a varying number of samples (from 5571 to 7387).
Activities: The dataset includes 18 unique activities with varying frequencies, creating a class imbalance.
Sequence Length: The number of measurements per activity varies, introducing variability in the time-series data (3000, 3500 or 4000).
Body Part: The body part where the sensor was placed.
Side: The side of the body where the sensor was placed.

iii. Each record is labeled with one of 18 activities, such as "walking," "stairs down," or "using phone." These labels represent the movement or action being performed during the recording session.

iv. The activity distribution shows significant class imbalance. Activities like "using phone," "walking freely," and "walking holding a tray" are much more frequent, while others like "stairs down," "stairs up," and "brushing teeth" are underrepresented.

Treating all labels equally without addressing the imbalance might bias the model toward predicting the dominant classes. Techniques like over sampling, under sampling, or class weights should be applied during model training. One option Is to sample from each combination of the data so no class will be underrepresented.

Certain activities might have more practical importance in real-world applications. For example:

"Stairs down" and "stairs up" could be critical for fall detection systems.

"Walking with hands in pockets" could be relevant for ergonomic or safety analysis.

v. In the data we have 8 users in the train section and 21 users in the test section. The competition data was split in the way that in the train we have 50,248 rows, and in the test 74,744 rows.

b.

This task is a **classification problem** where the goal is to predict the type of activity performed based on movement data.

We are not predicting future events or forecasting but rather classifying an activity based on sensor readings.

c.

Autoencoders:

Train an autoencoder to compress and reconstruct the data, learning meaningful representations in the process.

Anomaly Detection

Train the model to identify unusual patterns in the sensor data as anomalies. To do that we need to learn the typical structure of movement data.

2) a.

For our validation strategy, we allocated 20% of the data as the validation set, ensuring that samples were proportionally taken from all activity types and sensor categories to maintain a representative distribution. This approach guarantees that the model is evaluated in diverse scenarios, reflecting real-world variability. Additionally, we decided to build two separate models: one dedicated to processing data from the Vicon sensor and the other for the smartwatch sensor. This separation allows each model to specialize in the unique characteristics of its respective sensor data, maximizing performance and interpretability.

b. For a naive baseline we used:

most common label -

```
Train accuracy: 0.08267431018406603
Validation accuracy: 0.08258403651957394
```

Last Observation Carried Forward -

```
Train accuracy: 0.06212660151258763
Validation accuracy: 0.06280260063632591
```

c. In this section we used random forest which predicted much better.

```
Validation Accuracy: 0.7985969014907922
```

```
                              precision    recall  f1-score   support

               brushing_teeth       0.83      0.82      0.83       101
                         idle       0.85      0.82      0.83       185
            preparing_sandwich      0.81      0.82      0.81       149
                  reading_book       0.89      0.87      0.88       174
                   stairs_down       0.90      0.90      0.90        80
                     stairs_up       0.88      0.86      0.87        87
                        typing       0.89      0.89      0.89        84
                   using_phone       0.81      0.79      0.80       287
          using_remote_control       0.78      0.78      0.78       173
                walking_freely       0.74      0.80      0.77       313
         walking_holding_a_tray       0.84      0.86      0.85       298
         walking_with_handbag       0.76      0.79      0.78       303
walking_with_hands_in_pockets       0.83      0.82      0.83       288
 walking_with_object_underarm       0.79      0.72      0.75       300
       washing_face_and_hands       0.68      0.74      0.71       172
                  washing_mug       0.74      0.71      0.72       176
                washing_plate       0.70      0.73      0.72       166
                      writing       0.88      0.75      0.81        85

                     accuracy                           0.80      3421
                    macro avg       0.81      0.81      0.81      3421
                 weighted avg       0.80      0.80      0.80      3421

[[ 83   1   2   0   0   0   0   0   2   0   1   0   1   2   5   1   2   1]
 [  0 151   2   8   0   0   2   1  11   0   3   0   0   0   0   1   1   5]
 [  4   1 122   0   0   0   0   0   1   1   0   0   1   0   7   2  10   0]
 [  0   6   3 152   0   0   1   0   5   0   2   0   0   0   2   3   0   0]
 [  0   0   0   0  72   7   0   0   0   1   0   0   0   0   0   0   0   0]
 [  1   0   0   0   7  75   0   0   0   1   0   3   0   0   0   0   0   0]
 [  0   2   1   2   0   0  75   0   0   0   0   1   1   0   0   0   1   1]
 [  2   1   0   0   0   0   0 228   0  12   8  16   8  10   1   0   1   0]
 [  3  12   2   5   0   0   1   0 135   0   5   0   0   1   5   3   0   1]
 [  1   0   0   0   0   2   0  11   0 251   5  25   7  11   0   0   0   0]
 [  0   0   1   0   0   0   0   4   4   8 256   6   7   9   1   1   1   0]
 [  1   0   0   0   1   0   0   7   1  28   2 240   9  12   1   1   0   0]
 [  0   1   1   0   0   0   0   9   0  13   3   9 237  12   1   0   2   0]
 [  1   0   2   0   0   1   1  17   0  21  14  15   8 217   3   0   0   0]
 [  1   0   3   0   0   0   1   3   0   1   0   0   0   1 127  19  16   0]
 [  1   1   4   2   0   0   0   0   2   0   0   0   1   1  19 125  19   1]
 [  0   0   6   1   0   0   0   0   7   0   1   0   4   0  15  10 122   0]
 [  2   2   2   1   0   0   3   0   4   0   3   0   0   0   0   4   0  64]]
```

d. From here we decided to separate the two sensors and build a separate model for each sensor.

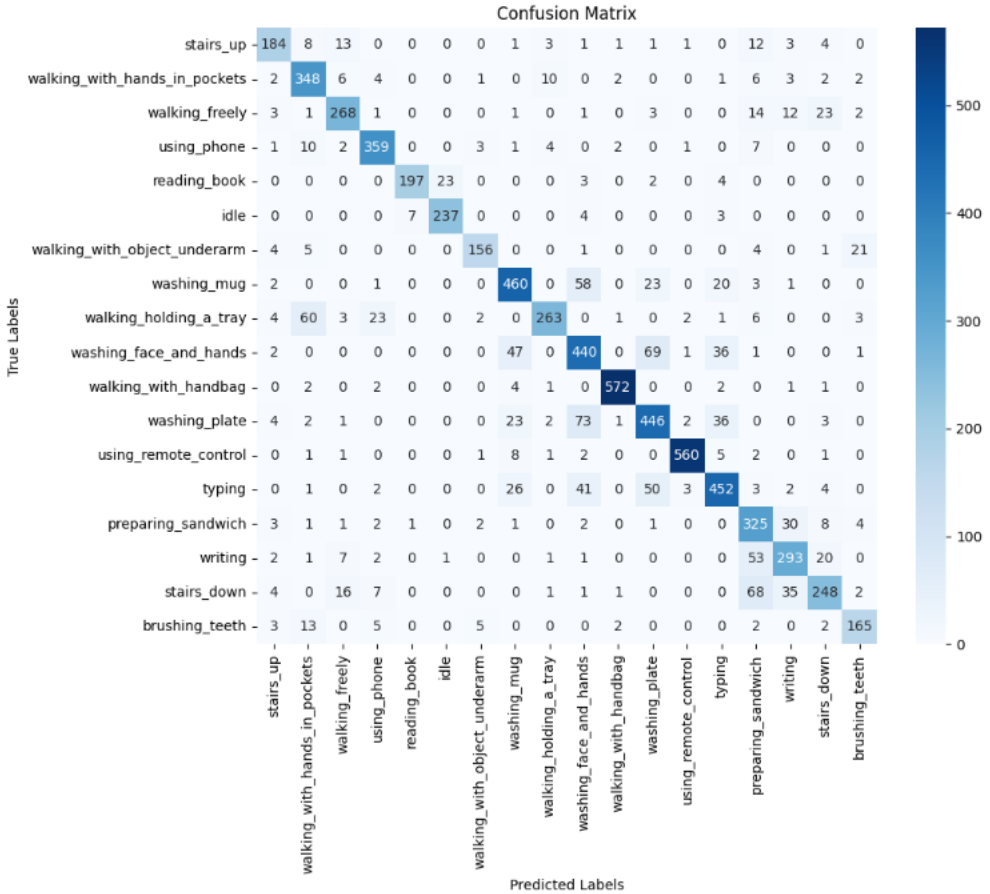
I. 1D-CNN

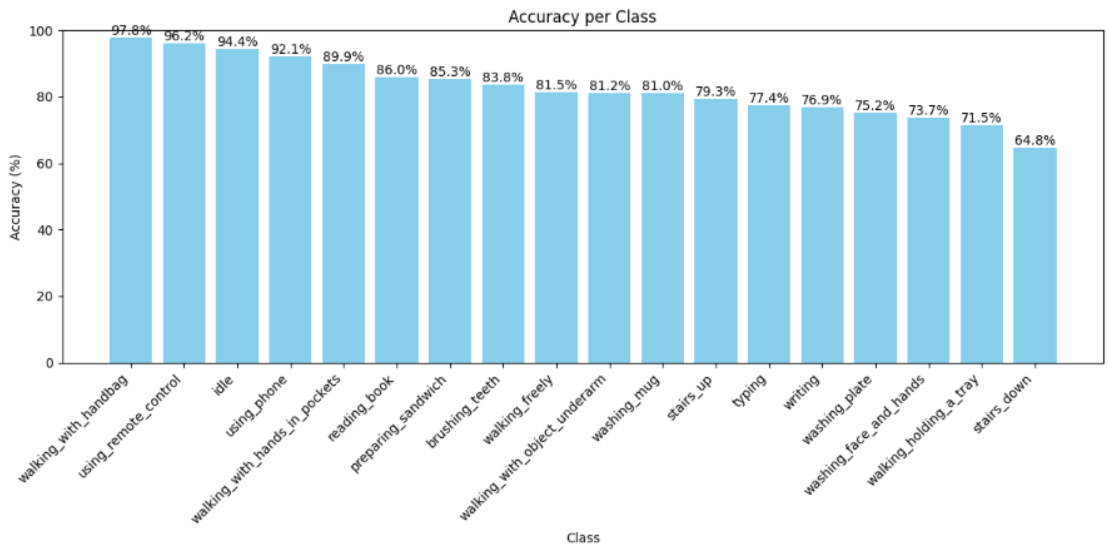
For Smart Watch:

Accuracy: 82.63%

Correct Predictions: 5973

Incorrect Predictions: 1256



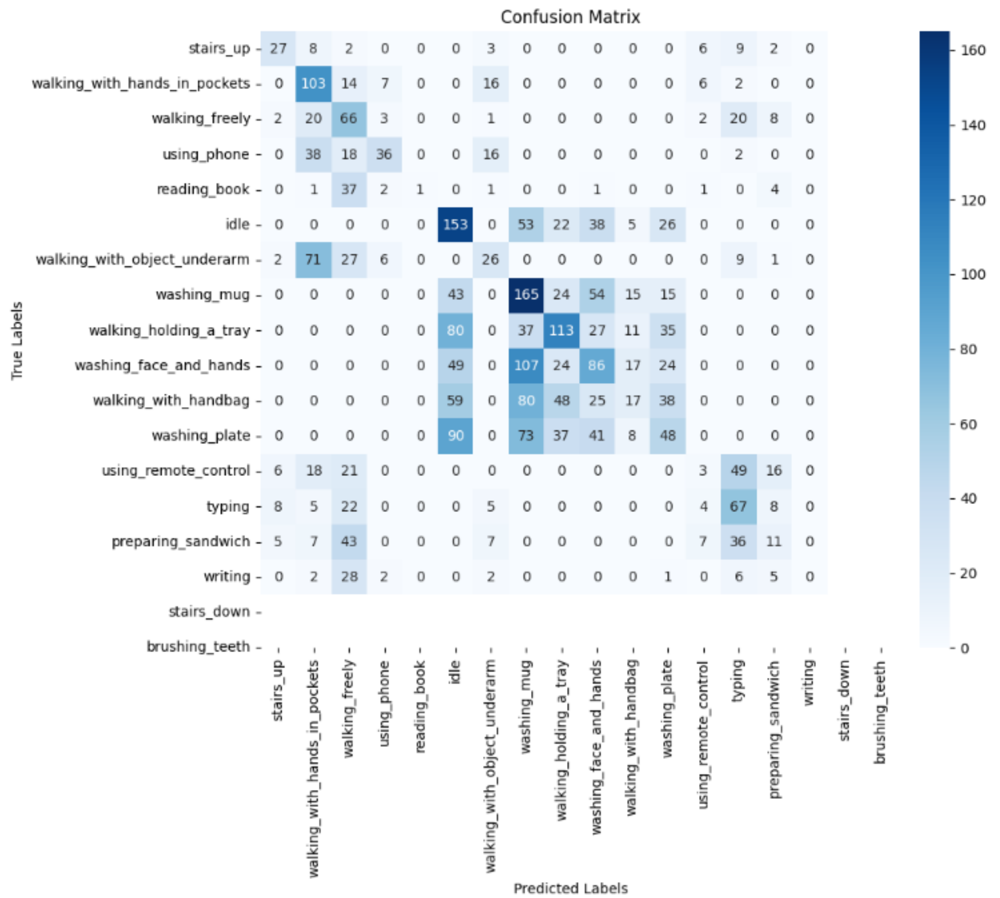


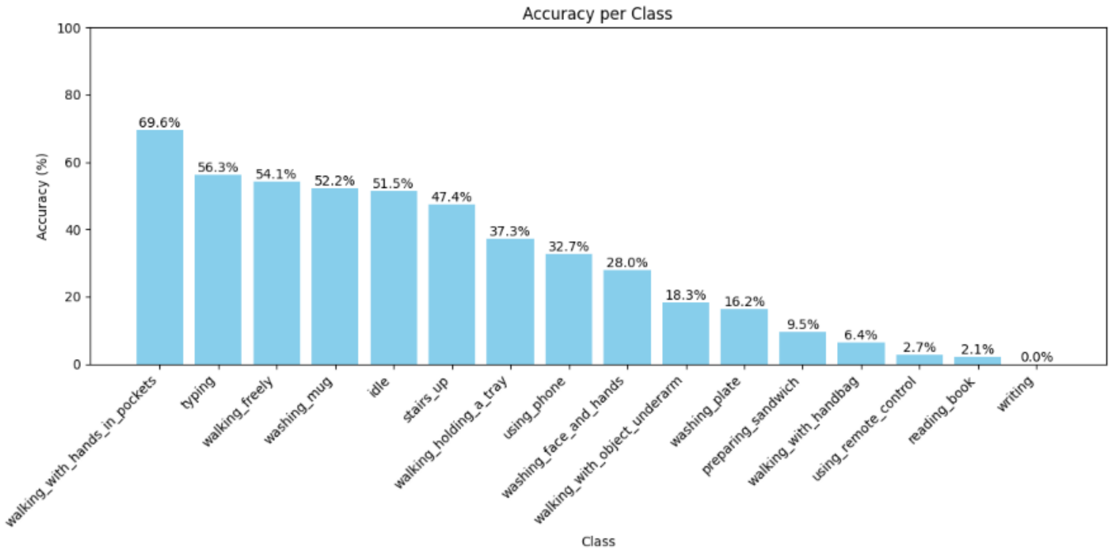
For Vicon:

Accuracy: 32.83%

Correct Predictions: 922

Incorrect Predictions: 1886





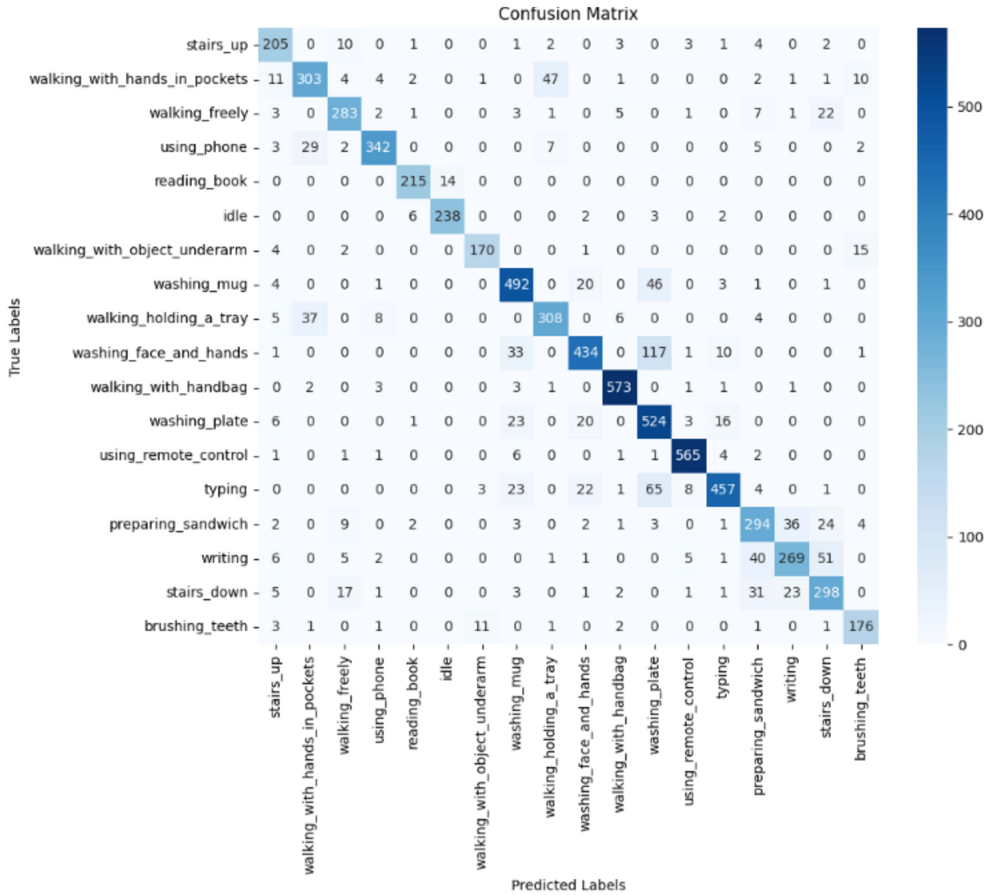
2. LSTM:

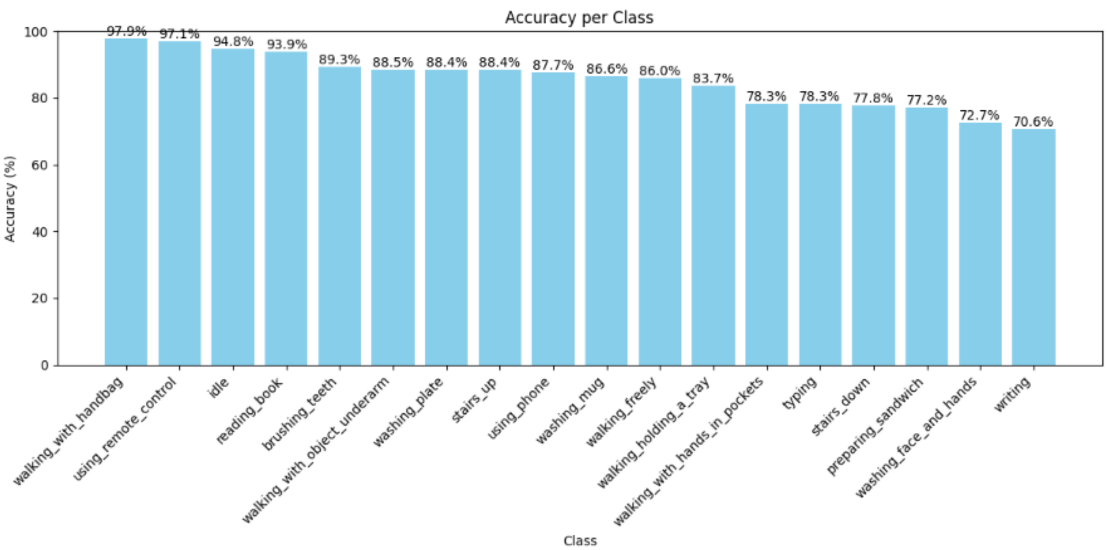
Smart Watch

Accuracy: 85.02%

Correct Predictions: 6146

Incorrect Predictions: 1083



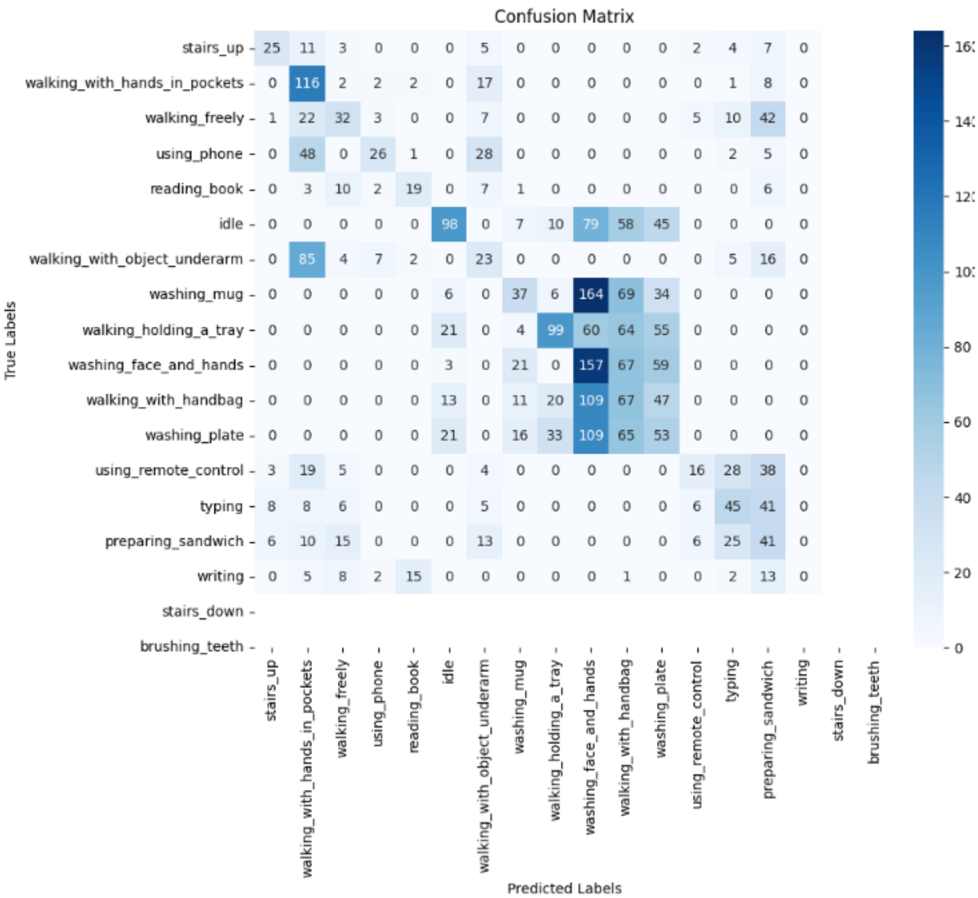


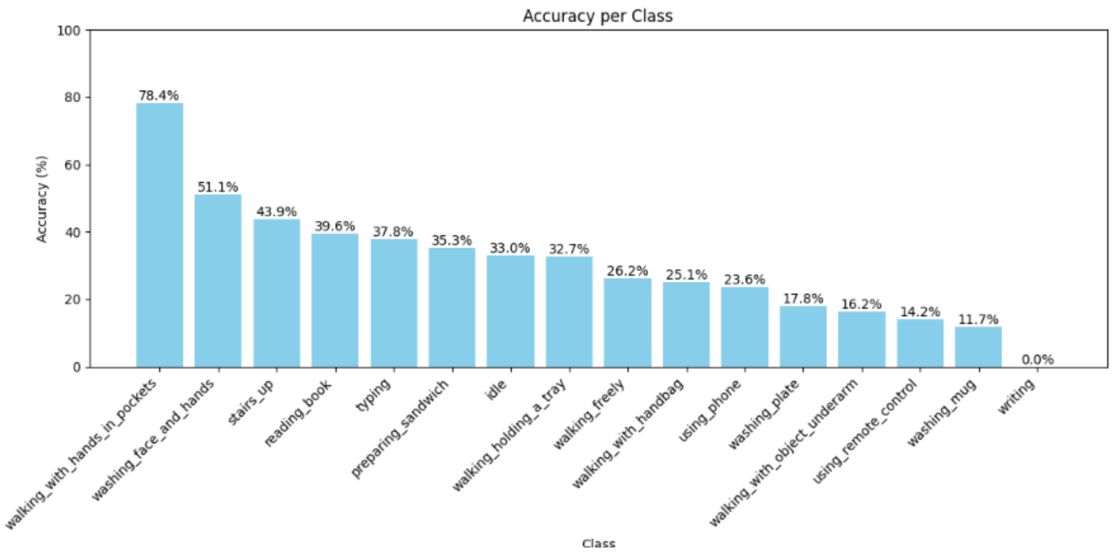
Vicon

Accuracy: 30.41%

Correct Predictions: 854

Incorrect Predictions: 1954





e.

We used anomaly detection and normalized the data, we analyzed the data and extracted a lot of features from the data such as mean, variance, skew and much more, we tried different types of models from random forest to complex neural networks. And the best results we got are:

```
Smartwatch Model Performance:
Accuracy: 0.9901

Confusion Matrix:
[[219   0   1   0   0   0   0   0   0   0   0   0   0   1   2   0   2   0]
 [  0 359   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0 321   1   0   0   0   0   2   0   0   0   0   0   0   0   5   0]
 [  0   0   0 376   0   0   0   0   0   0   0   0   0   0   1   0   0   0]
 [  0   0   0   0 245   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0 251   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0 165   0   0   1   0   0   0   0   0   0   0   3]
 [  0   0   0   0   0   0   0 575   0   1   0   1   0   1   0   0   0   0]
 [  1   2   0   2   0   0   0   0 360   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   1   0 614   0   1   0   1   0   0   0   0]
 [  0   0   0   0   0   0   0   0   2   0 578   0   2   0   2   0   0   0]
 [  1   1   0   0   1   0   0   2   0   5   0 593   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 544   0   1   0   1   0]
 [  0   0   0   0   0   0   0   0   0   2   0   1   0 597   0   0   0   0]
 [  1   0   2   0   0   0   0   0   0   0   0   0   0   0 410   7   2   1]
 [  0   0   1   0   0   0   0   0   0   0   0   0   0   0   0 367   1   0]
 [  1   0   2   0   0   0   0   0   0   0   0   0   0   0   1   1 402   0]
 [  0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0 190]]

Classification Report:
                              precision    recall  f1-score   support

               brushing_teeth      0.98      0.97      0.98       225
                         idle      0.99      1.00      1.00       359
           preparing_sandwich      0.98      0.98      0.98       329
                 reading_book      0.99      1.00      0.99       377
                  stairs_down      1.00      1.00      1.00       245
                    stairs_up      1.00      1.00      1.00       251
                       typing      0.99      0.98      0.99       169
                  using_phone      0.99      0.99      0.99       578
          using_remote_control      0.99      0.99      0.99       365
               walking_freely      0.99      1.00      0.99       617
         walking_holding_a_tray      1.00      0.99      0.99       584
          walking_with_handbag      0.99      0.98      0.99       603
walking_with_hands_in_pockets      1.00      1.00      1.00       546
 walking_with_object_underarm      0.99      0.99      0.99       600
        washing_face_and_hands      0.98      0.97      0.98       423
                  washing_mug      0.98      0.99      0.99       369
                washing_plate      0.97      0.99      0.98       407
                      writing      0.98      0.99      0.99       191

                     accuracy                          0.99      7238
                    macro avg      0.99      0.99      0.99      7238
                 weighted avg      0.99      0.99      0.99      7238
```

```
Vicon Model Performance:
Accuracy: 0.9684

Confusion Matrix:
[[ 52   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0 141   0   1   0   0   4   0   0   0   0   0   0   0   0   0]
 [  0   0 118   0   0   0   0   0   0   0   0   0   1   1   1   0]
 [  0   2   0 117   0   0   2   0   0   0   0   0   0   0   0   0]
 [  0   1   1   0  45   0   0   0   0   0   0   0   0   0   0   1]
 [  0   0   0   0   0 268   0   3   0   1   0   2   0   0   0   0]
 [  0   3   0   0   0   0 147   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   6   0 290   3   7   3   2   0   0   0   0]
 [  0   0   0   0   0   3   0   0 307   0   1   1   0   0   0   0]
 [  0   0   0   0   0   1   0   5   3 280   1   2   0   0   0   0]
 [  0   0   0   0   0   0   0   6   5   4 247   2   0   0   0   0]
 [  0   0   0   0   0   3   0   1   2   0   0 321   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 102   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0 121   1   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   1   0 124   0]
 [  0   0   0   0   0   0   1   0   0   0   0   0   0   0   0  44]]

Classification Report:
                              precision    recall  f1-score   support

               brushing_teeth      1.00      0.98      0.99        53
                         idle      0.96      0.97      0.96       146
           preparing_sandwich      0.98      0.98      0.98       121
                 reading_book      0.99      0.97      0.98       121
                       typing      1.00      0.94      0.97        48
                  using_phone      0.95      0.98      0.97       274
          using_remote_control      0.95      0.98      0.97       150
               walking_freely      0.95      0.93      0.94       311
         walking_holding_a_tray      0.96      0.98      0.97       312
          walking_with_handbag      0.96      0.96      0.96       292
walking_with_hands_in_pockets      0.98      0.94      0.96       264
 walking_with_object_underarm      0.97      0.98      0.98       327
        washing_face_and_hands      0.98      1.00      0.99       102
                  washing_mug      0.99      0.99      0.99       122
                washing_plate      0.98      0.99      0.99       125
                      writing      0.98      0.98      0.98        45

                     accuracy                          0.97      2813
                    macro avg      0.97      0.97      0.97      2813
                 weighted avg      0.97      0.97      0.97      2813
```

```
Log Loss on Validation Set: 0.18023187367516463
```

f.

In general, The Vicon model predicted a bit less good, the hardest thing to predict for him was similar activities. I was mistaken in things like walking freely with walking with object under the arm or using remote control and using phone. There are things that the mode predicted perfectly like brushing teeth, probably because it is unique movement.

3 ways to improve the model are:

<u>Feature importance</u>: we will reduce some of the features and try to send only the most influent parameters so the model will be more efficient.
<u>Dimensional reduction methods</u>: we tried methods like PCA.
<u>Normalization methods</u>: we used several different methods to normalize the data from Standard Scaler to minmax to try and find the best normalization methods for the data.

g.

We prioritized the suggestions and decided that feature importance and Normalization are more important, and we focused on them more. We also tried dimensional reduction, but it worked a bit less good.

After these improvements we got:

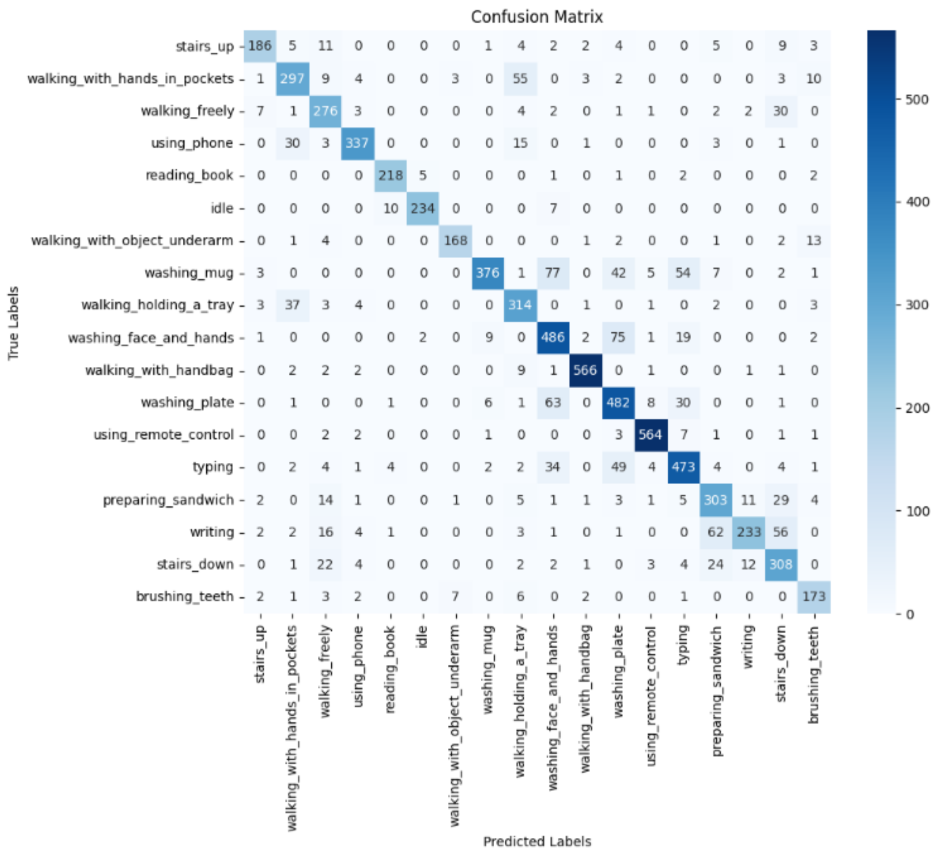
1. 1D-CNN:

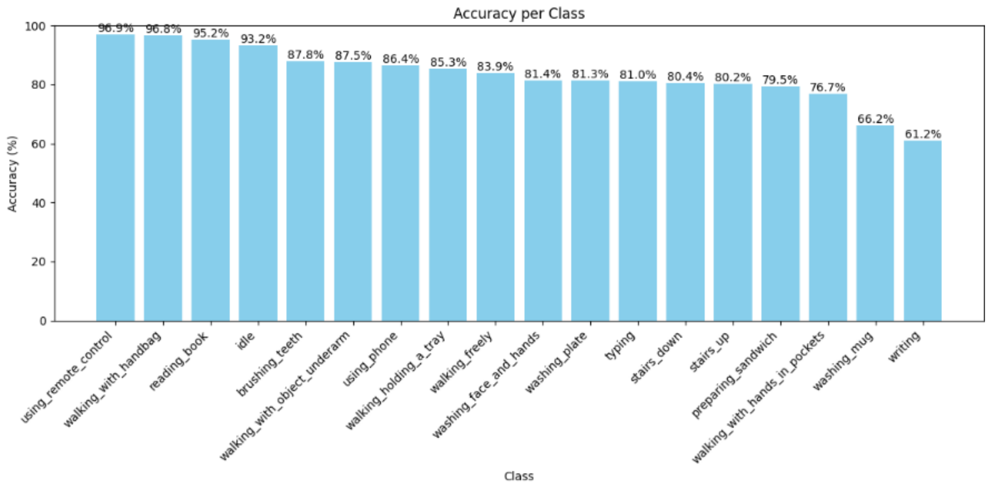
Smart Watch

Accuracy: 83%

Correct Predictions: 5994

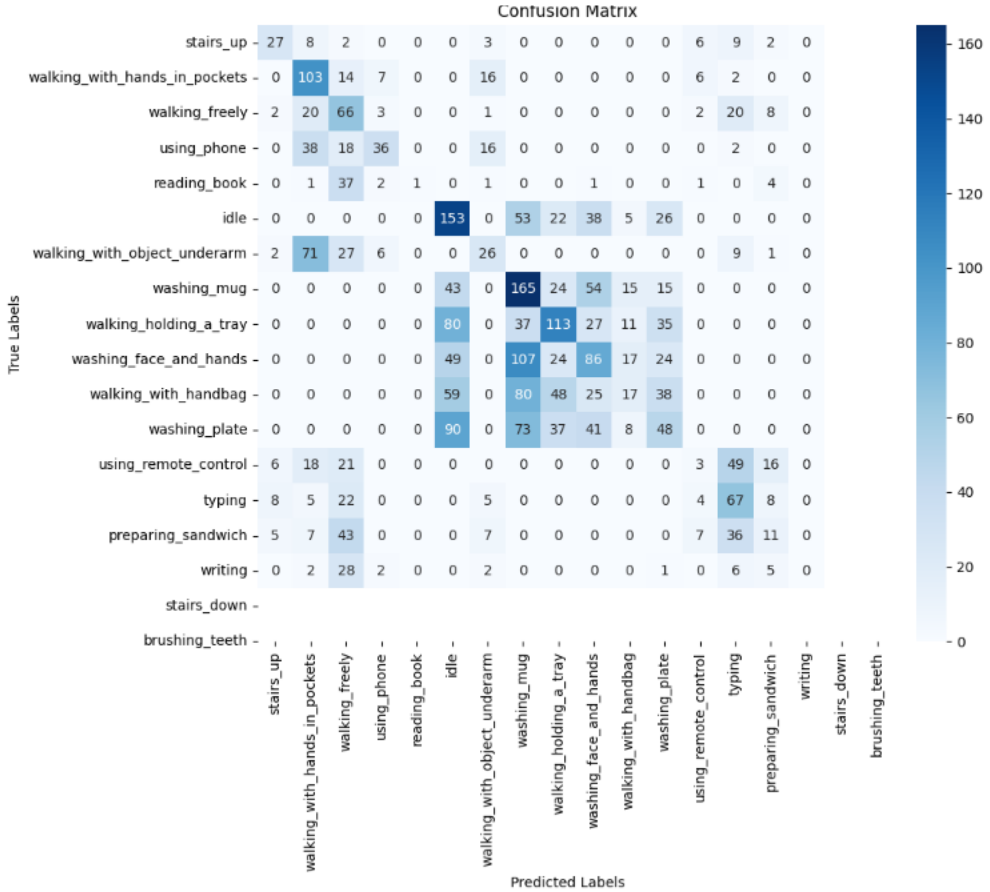
Incorrect Predictions: 1235



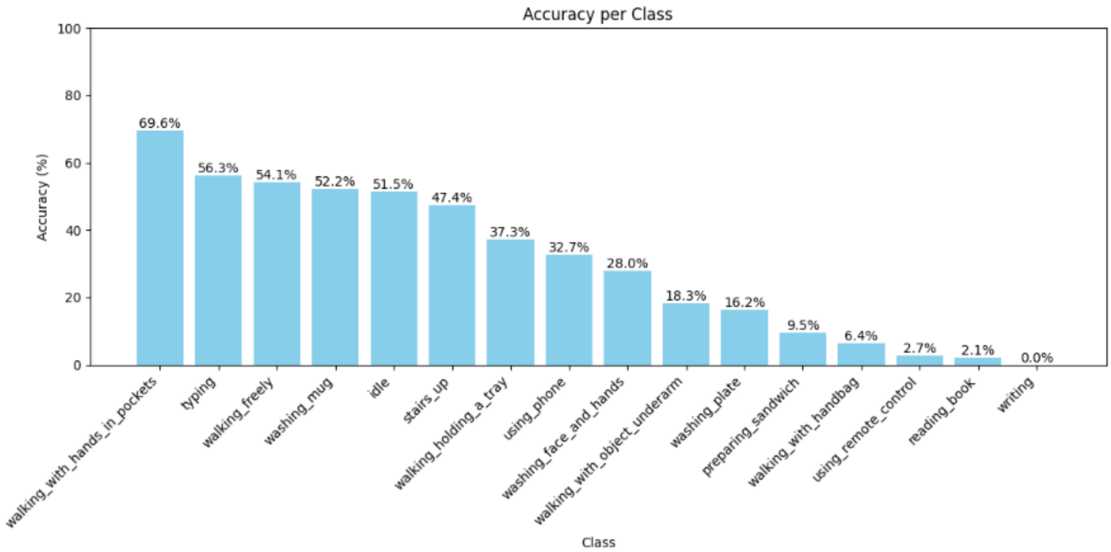


Vicon

Accuracy: 33%
Correct Predictions: 922
Incorrect Predictions: 1886



	stairs_up	walking_with_hands_in_pockets	walking_freely	using_phone	reading_book	idle	walking_with_object_underarm	washing_mug	walking_holding_a_tray	washing_face_and_hands	walking_with_handbag	washing_plate	using_remote_control	typing	preparing_sandwich	writing	stairs_down	brushing_teeth
stairs_up	27	8	2	0	0	0	3	0	0	0	0	0	6	9	2	0		
walking_with_hands_in_pockets	0	103	14	7	0	0	16	0	0	0	0	0	6	2	0	0		
walking_freely	2	20	66	3	0	0	1	0	0	0	0	0	2	20	8	0		
using_phone	0	38	18	36	0	0	16	0	0	0	0	0	0	2	0	0		
reading_book	0	1	37	2	1	0	1	0	0	1	0	0	1	0	4	0		
idle	0	0	0	0	0	153	0	53	22	38	5	26	0	0	0	0		
walking_with_object_underarm	2	71	27	6	0	0	26	0	0	0	0	0	0	9	1	0		
washing_mug	0	0	0	0	0	43	0	165	24	54	15	15	0	0	0	0		
walking_holding_a_tray	0	0	0	0	0	80	0	37	113	27	11	35	0	0	0	0		
washing_face_and_hands	0	0	0	0	0	49	0	107	24	86	17	24	0	0	0	0		
walking_with_handbag	0	0	0	0	0	59	0	80	48	25	17	38	0	0	0	0		
washing_plate	0	0	0	0	0	90	0	73	37	41	8	48	0	0	0	0		
using_remote_control	6	18	21	0	0	0	0	0	0	0	0	0	3	49	16	0		
typing	8	5	22	0	0	0	5	0	0	0	0	0	4	67	8	0		
preparing_sandwich	5	7	43	0	0	0	7	0	0	0	0	0	7	36	11	0		
writing	0	2	28	2	0	0	2	0	0	0	0	1	0	6	5	0		
stairs_down																		
brushing_teeth																		



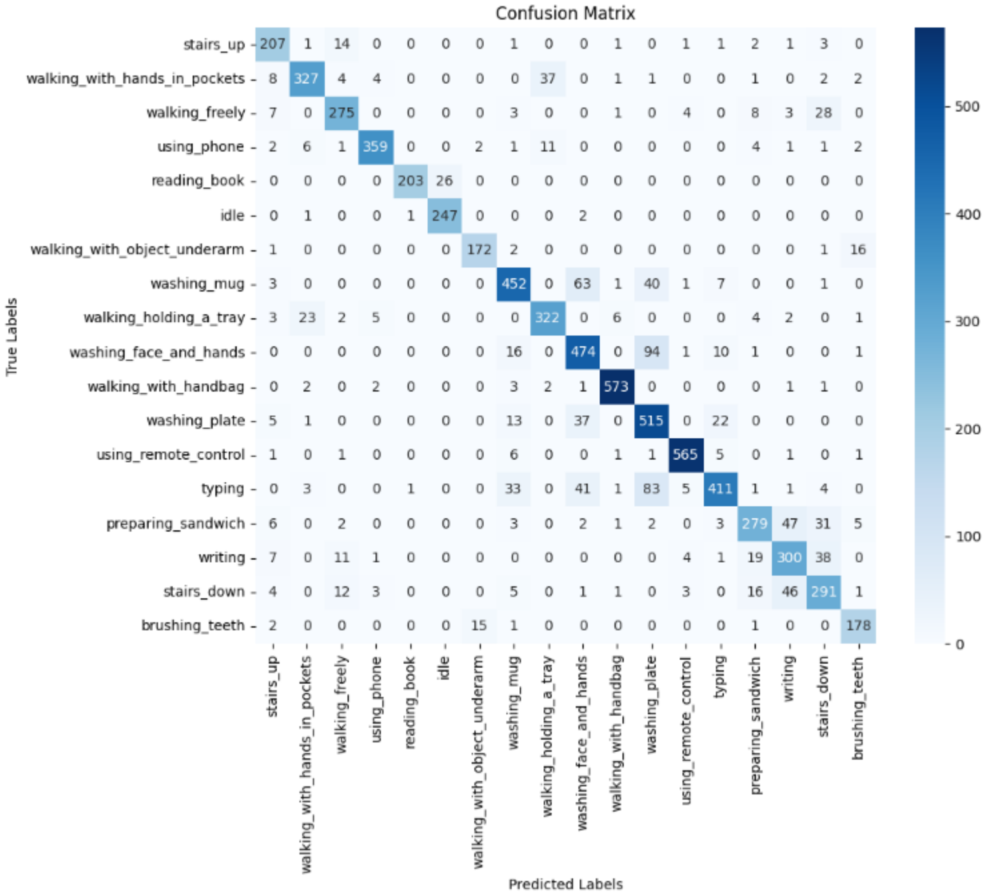
LSTM:

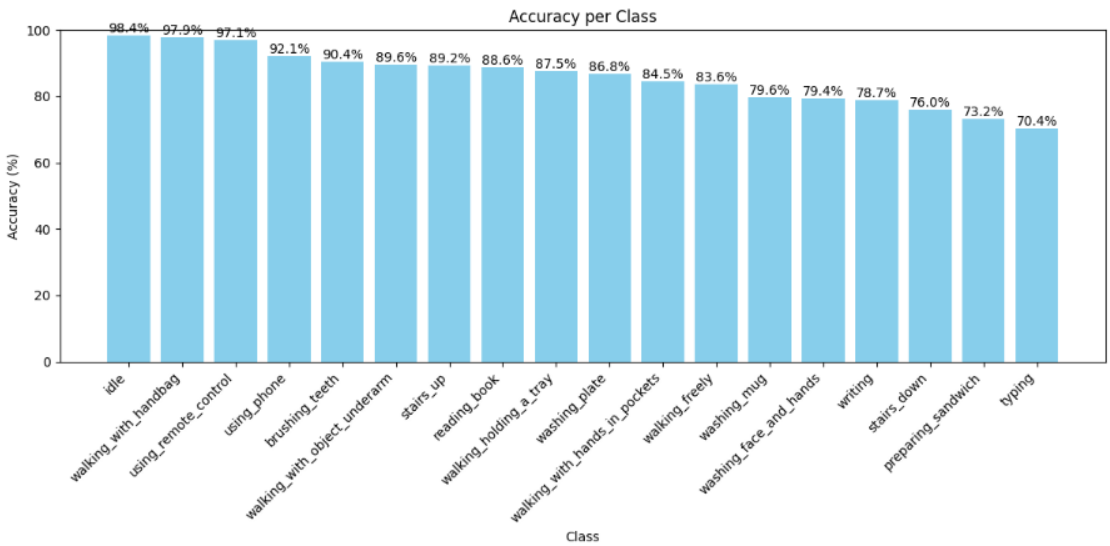
For Smart Watch:

Accuracy: 85.07%

Correct Predictions: 6150

Incorrect Predictions: 1079



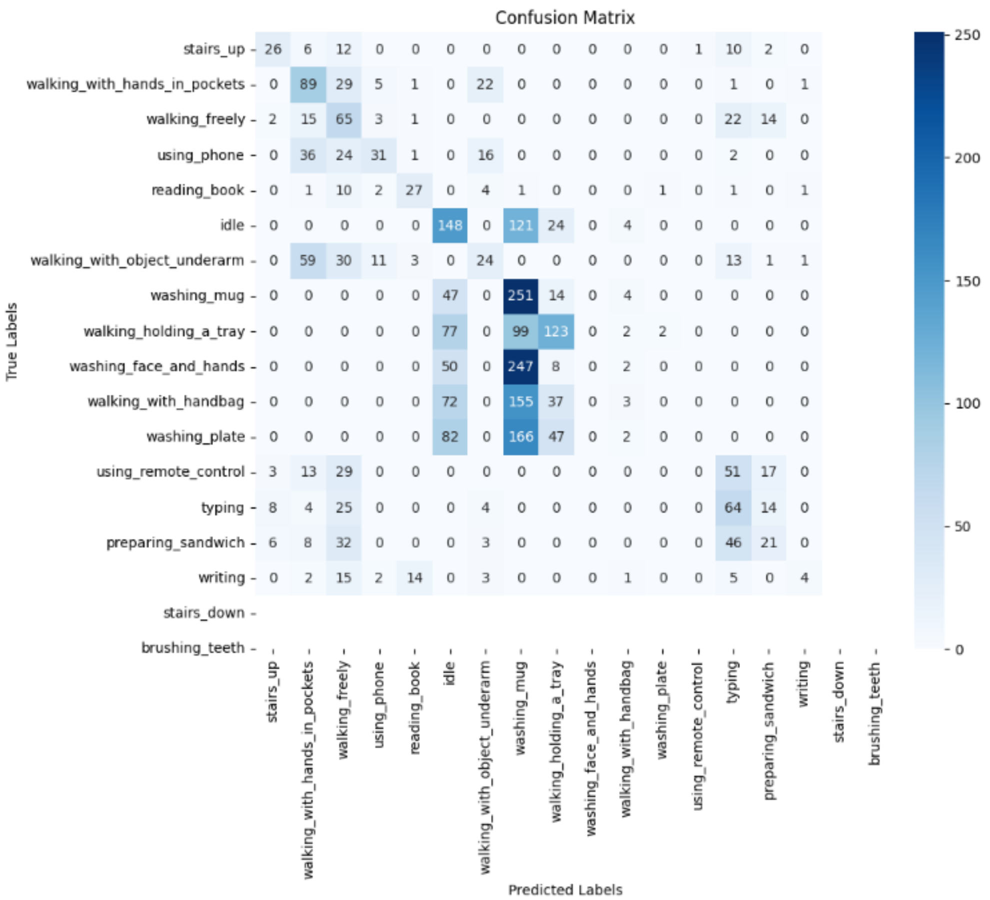


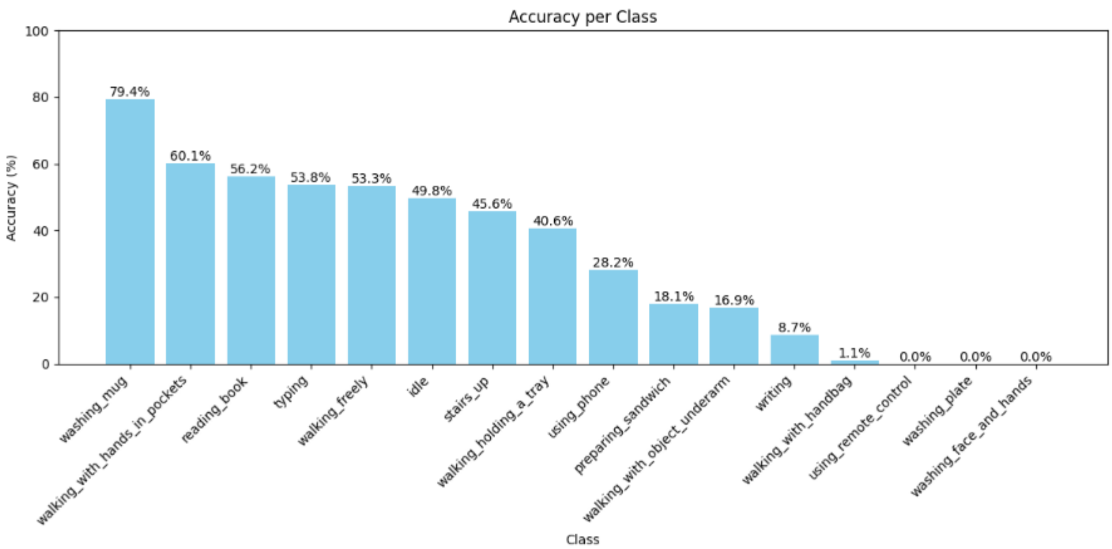
For vicon:

Accuracy: 31.20%

Correct Predictions: 876

Incorrect Predictions: 1932





We received minor changes in the results that does not determine in Distinction that the changes have helped.

3)

In this project, we focused on classifying human activities based on sequences of numerical data that represent the movement. To achieve this, we conducted an informative analysis to better understand the behavior of the data. We began by splitting the dataset into two parts based on the type of sensor, as each sensor measures activities differently. This required us to address the gap between the two measurement methods and analyze the statistical variables.

The dataset consisted of a total of 175 million records, which posed a significant computational challenge. Following the data analysis, we decided to represent the data through feature vectors composed of various statistical metrics. This allowed us to condense the information into a more compact and manageable format. We normalized the data, identified the most impactful statistical features, and performed statistical tests to examine correlations and variances.

Next, we applied dimensionality reduction techniques such as PCA and statistical tests to further streamline the data. The processed data was then fed into two different models: a 1D-CNN and an LSTM. Each sensor type was assigned to its own model, and the results from both were later compared. To optimize performance and reduce the log loss, we incorporated additional models, including a Multi-Layer Perceptron (MLP) and a Random Forest.

This project enhanced our ability to process and analyze large datasets, address challenges, and identify the optimal methods for achieving the best-performing model.