MLPR PROJECT

Team

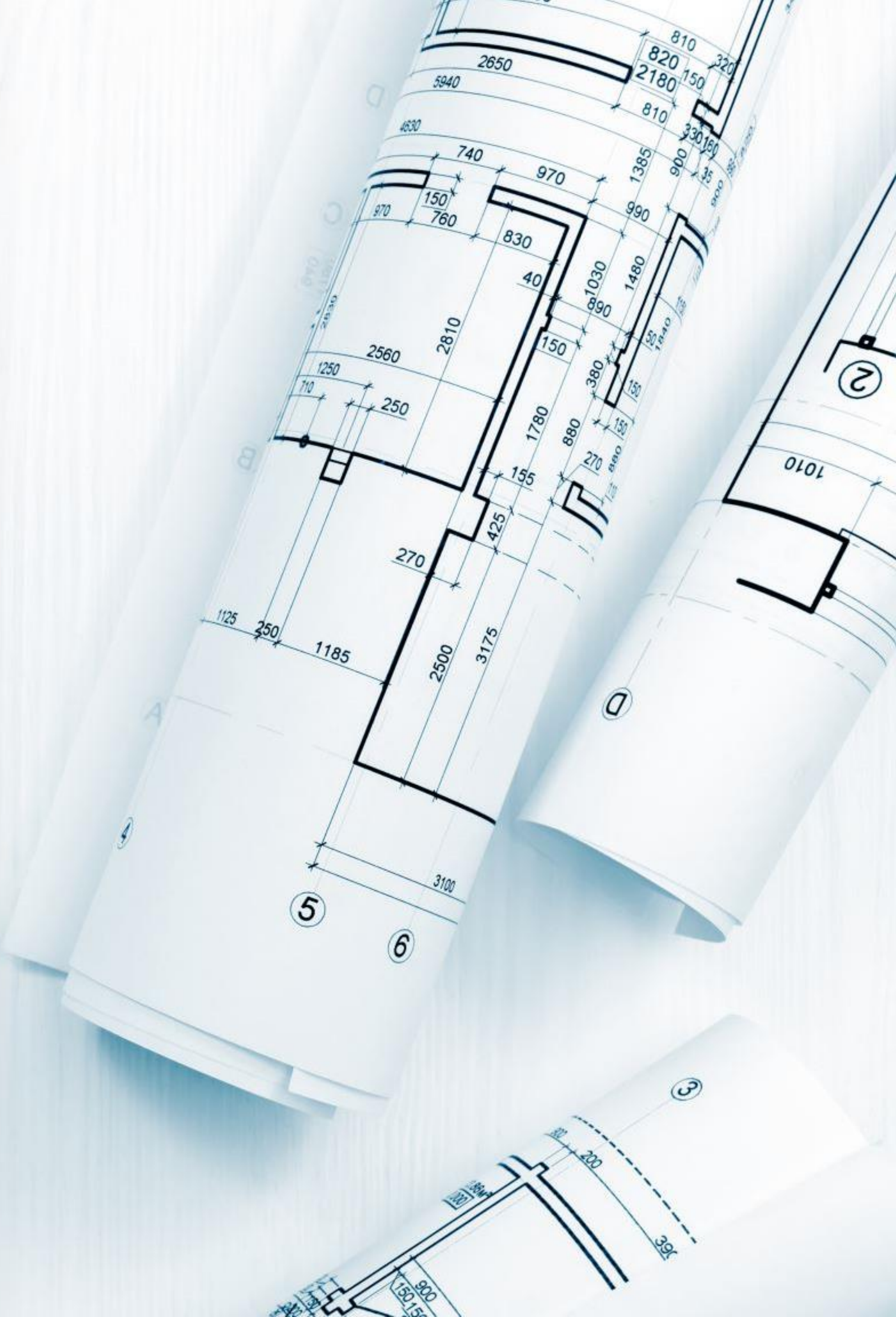
Nishant Mahajan

Prashant Tiwari

Vansh Gupta

Project name

UTILIZING SMARTPHONE SENSOR DATA FOR STUDENT'S LIFE ANALYSIS



Problem statement

Figuring out student day-to-day activities and their performance using smart devices.

Motivation

Early Intervention and support

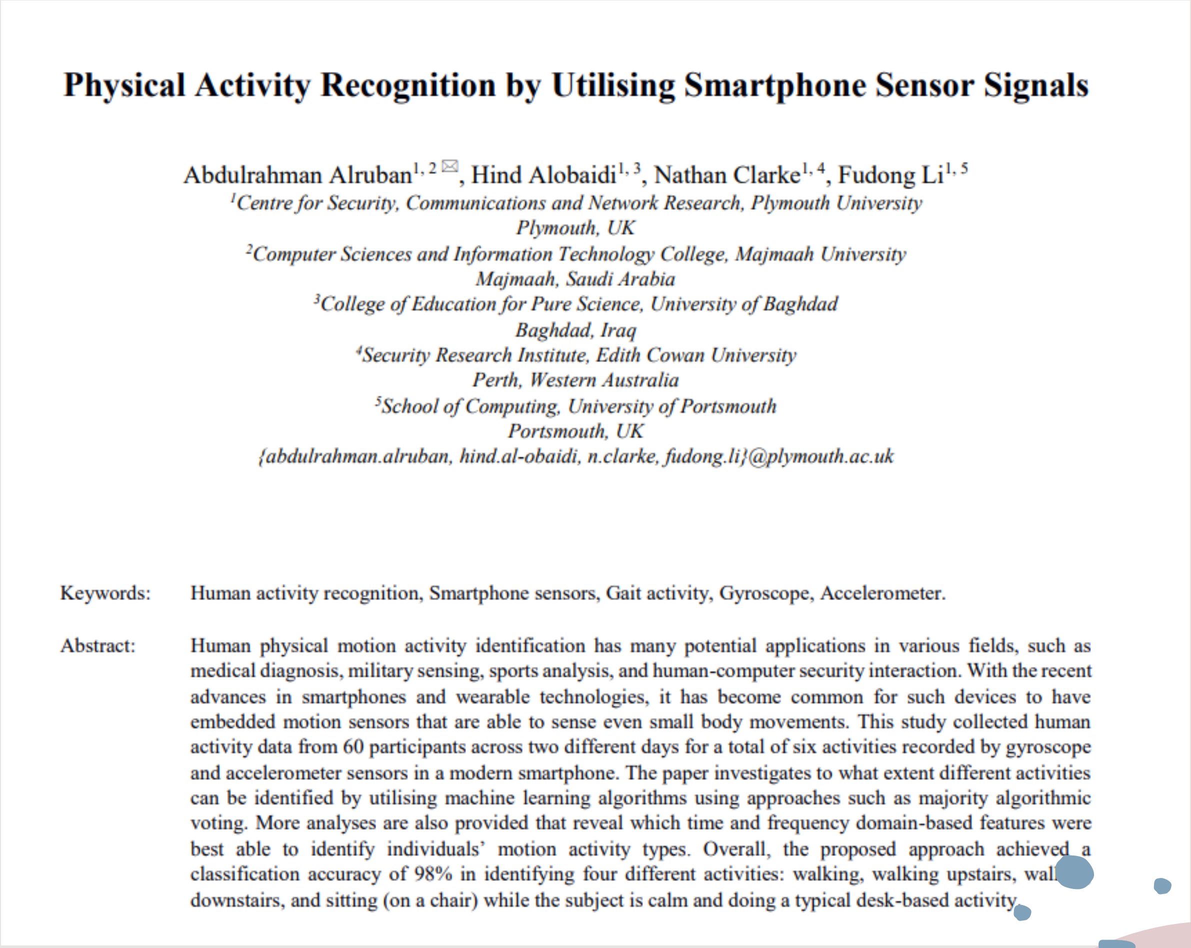
Proper Time Management

Improved Well-being

Background Research

- **Data Collection Methods:**
- Wearable sensors like smartwatches and smartphones were utilized.
- Data was collected during various physical activities: walking, running, and moving downstairs, and upstairs.
- **Classification Algorithms:**
- Employed three classification algorithms: eXtreme Gradient Boosting (XGB), feedforward neural network, and Support Vector Machine (SVM).
- **Accuracy:**
- Accuracy for identifying students' physical activity was 98%, based on smartphone-embedded gyroscope and accelerometer sensor signal gathered.

https://arxiv.org/ftp/arxiv/papers/2201/2201.08688.pdf



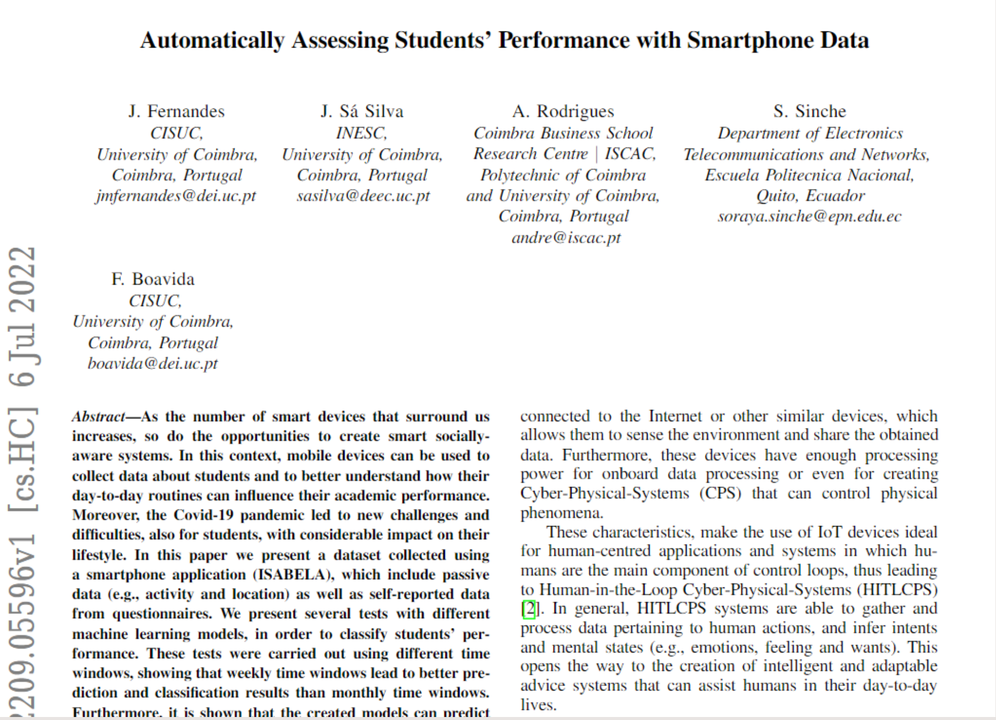
Physical Activity Recognition by Utilising Smartphone Sensor Signals

Abdulrahman Alruban[1, 2 ✉], Hind Alobaidi[1, 3], Nathan Clarke[1, 4], Fudong Li[1, 5]
[1]*Centre for Security, Communications and Network Research, Plymouth University*
Plymouth, UK
[2]*Computer Sciences and Information Technology College, Majmaah University*
Majmaah, Saudi Arabia
[3]*College of Education for Pure Science, University of Baghdad*
Baghdad, Iraq
[4]*Security Research Institute, Edith Cowan University*
Perth, Western Australia
[5]*School of Computing, University of Portsmouth*
Portsmouth, UK
{abdulrahman.alruban, hind.al-obaidi, n.clarke, fudong.li}@plymouth.ac.uk

Keywords: Human activity recognition, Smartphone sensors, Gait activity, Gyroscope, Accelerometer.

Abstract: Human physical motion activity identification has many potential applications in various fields, such as medical diagnosis, military sensing, sports analysis, and human-computer security interaction. With the recent advances in smartphones and wearable technologies, it has become common for such devices to have embedded motion sensors that are able to sense even small body movements. This study collected human activity data from 60 participants across two different days for a total of six activities recorded by gyroscope and accelerometer sensors in a modern smartphone. The paper investigates to what extent different activities can be identified by utilising machine learning algorithms using approaches such as majority algorithmic voting. More analyses are also provided that reveal which time and frequency domain-based features were best able to identify individuals' motion activity types. Overall, the proposed approach achieved a classification accuracy of 98% in identifying four different activities: walking, walking upstairs, wal[k]ing downstairs, and sitting (on a chair) while the subject is calm and doing a typical desk-based activity.

Background Research



Automatically Assessing Students' Performance with Smartphone Data

J. Fernandes
CISUC,
University of Coimbra,
Coimbra, Portugal
jmfernandes@dei.uc.pt

J. Sá Silva
INESC,
University of Coimbra,
Coimbra, Portugal
sasilva@deec.uc.pt

A. Rodrigues
Coimbra Business School
Research Centre | ISCAC,
Polytechnic of Coimbra
and University of Coimbra,
Coimbra, Portugal
andre@iscac.pt

S. Sinche
Department of Electronics
Telecommunications and Networks,
Escuela Politecnica Nacional,
Quito, Ecuador
soraya.sinche@epn.edu.ec

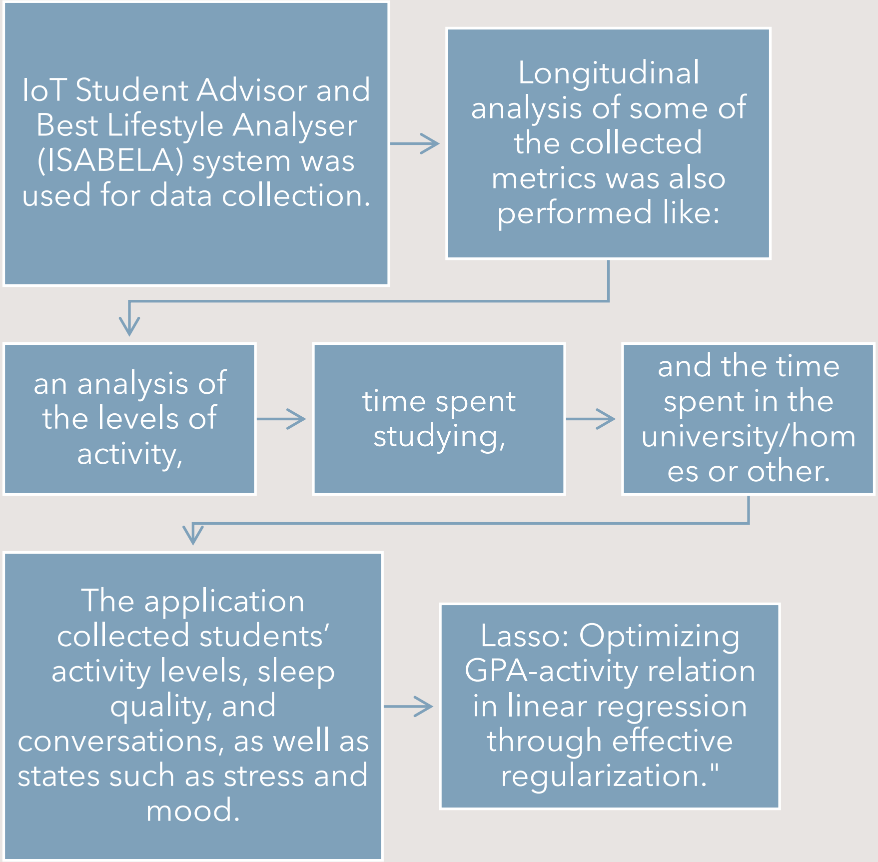
F. Boavida
CISUC,
University of Coimbra,
Coimbra, Portugal
boavida@dei.uc.pt

2209.05596v1 [cs.HC] 6 Jul 2022

***Abstract*—As the number of smart devices that surround us increases, so do the opportunities to create smart socially-aware systems. In this context, mobile devices can be used to collect data about students and to better understand how their day-to-day routines can influence their academic performance. Moreover, the Covid-19 pandemic led to new challenges and difficulties, also for students, with considerable impact on their lifestyle. In this paper we present a dataset collected using a smartphone application (ISABELA), which include passive data (e.g., activity and location) as well as self-reported data from questionnaires. We present several tests with different machine learning models, in order to classify students' performance. These tests were carried out using different time windows, showing that weekly time windows lead to better prediction and classification results than monthly time windows. Furthermore, it is shown that the created models can predict**

connected to the Internet or other similar devices, which allows them to sense the environment and share the obtained data. Furthermore, these devices have enough processing power for onboard data processing or even for creating Cyber-Physical-Systems (CPS) that can control physical phenomena.

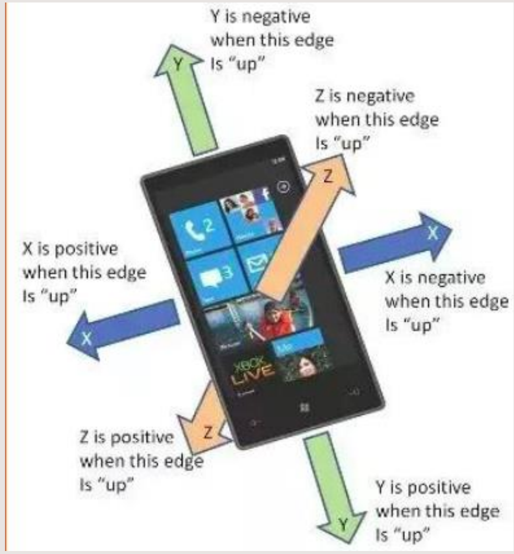
These characteristics, make the use of IoT devices ideal for human-centred applications and systems in which humans are the main component of control loops, thus leading to Human-in-the-Loop Cyber-Physical-Systems (HITLCPS) [2]. In general, HITLCPS systems are able to gather and process data pertaining to human actions, and infer intents and mental states (e.g., emotions, feeling and wants). This opens the way to the creation of intelligent and adaptable advice systems that can assist humans in their day-to-day lives.



https://www.researchgate.net/publication/363537610_Automatically_Assessing_Students_Performance_with_Smartphone_Data

Data collection:

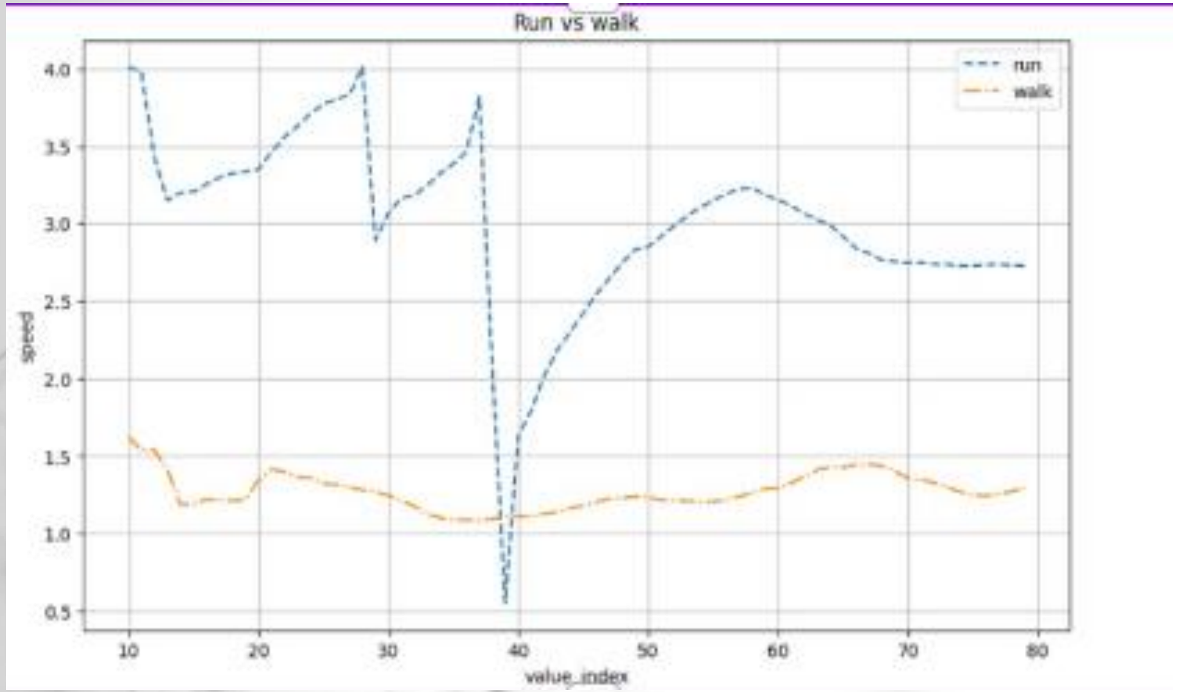
Dataset

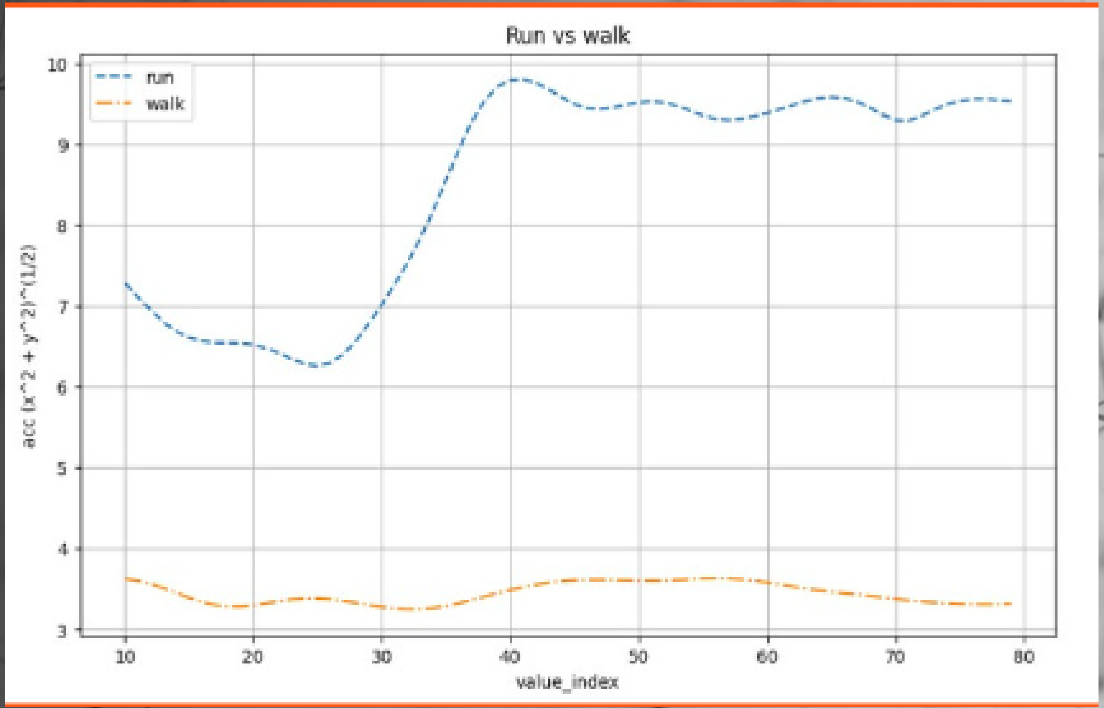


	time	seconds_elapsed	z	y	x
0	1697122414098249700	0.213250	-8.865	-4.153	0.568
1	1697122414108251400	0.223251	-8.948	-3.986	0.443
2	1697122414118253000	0.233253	-8.992	-3.891	0.406
3	1697122414128254700	0.243255	-9.030	-3.810	0.312
4	1697122414138256400	0.253256	-9.081	-3.673	0.442
...	...	...	...	...	...
25196	1697122666089584600	252.204585	-8.074	-5.506	0.805
25197	1697122666099586000	252.214586	-8.073	-5.500	0.855
25198	1697122666109587500	252.224587	-8.075	-5.490	0.900
25199	1697122666119588600	252.234589	-8.081	-5.474	0.938
25200	1697122666129590000	252.244590	-8.091	-5.455	0.967

- Collected mobile sensor data while doing various activities like walking, running, upstairs and downstairs, and rest for 2-3 min each.
- We used the Sensor Logger Android app for the data collection.
- Collected data from 10 students
- We got sensor data for gyroscope, Accelerometer, GPS, pedometer, etc.
- The sampling time for our data was 0.01 sec or a frequency of 100 Hz.
- Collected data type format: CSV.

Data Visualization





Kaggle Dataset

```
data.shape
```

(1215745, 14)

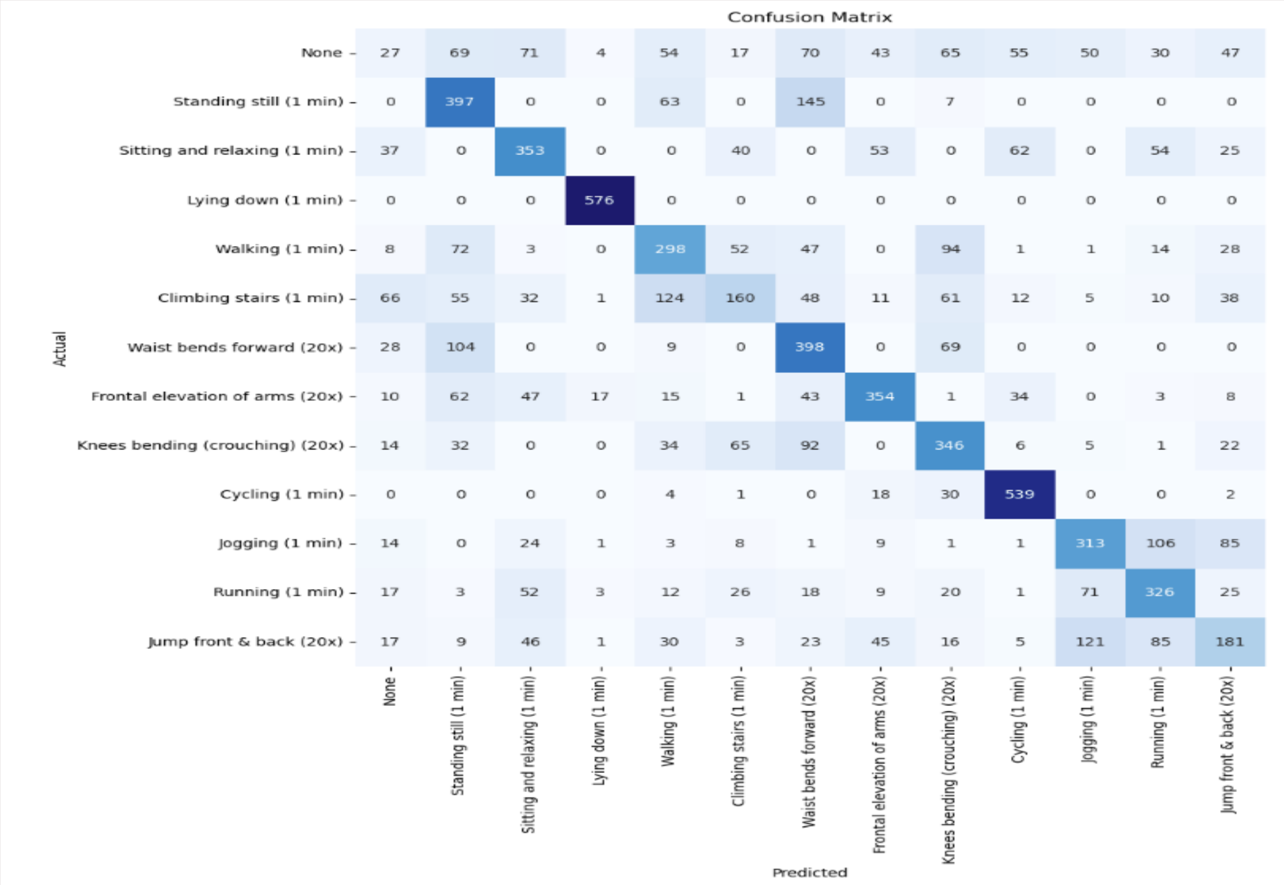
	alx	aly	alz	glx	gly	glz	arx	ary	arz	grx	gry	grz	Activity	subject
0	2.1849	-9.6967	0.63077	0.103900	-0.84053	-0.68762	-8.6499	-4.5781	0.187760	-0.44902	-1.0103	0.034483	0	subject1
1	2.3876	-9.5080	0.68389	0.085343	-0.83865	-0.68369	-8.6275	-4.3198	0.023595	-0.44902	-1.0103	0.034483	0	subject1
2	2.4086	-9.5674	0.68113	0.085343	-0.83865	-0.68369	-8.5055	-4.2772	0.275720	-0.44902	-1.0103	0.034483	0	subject1
3	2.1814	-9.4301	0.55031	0.085343	-0.83865	-0.68369	-8.6279	-4.3163	0.367520	-0.45686	-1.0082	0.025862	0	subject1
4	2.4173	-9.3889	0.71098	0.085343	-0.83865	-0.68369	-8.7008	-4.1459	0.407290	-0.45686	-1.0082	0.025862	0	subject1

alx: acceleration from the left-ankle sensor (X axis)
aly: acceleration from the left-ankle sensor (Y axis)
alz: acceleration from the left-ankle sensor (Z axis)
glx: gyro from the left-ankle sensor (X axis)
gly: gyro from the left-ankle sensor (Y axis)
glz: gyro from the left-ankle sensor (Z axis)
arx: acceleration from the right-lower-arm sensor (X axis)
ary: acceleration from the right-lower-arm sensor (Y axis)
arz: acceleration from the right-lower-arm sensor (Z axis)
grx: gyro from the right-lower-arm sensor (X axis)
gry: gyro from the right-lower-arm sensor (Y axis)
grz: gyro from the right-lower-arm sensor (Z axis)
subject: volunteer number

Feature Extraction (Data preprocessing)

1. Drop 'subject' column
2. Sample 2000 examples from each class
3. Split into X and y
4. Perform train-test split
5. Scale features using StandardScaler

Models Used :Logistic Regression:




```
Test Accuracy: 54.72%
```

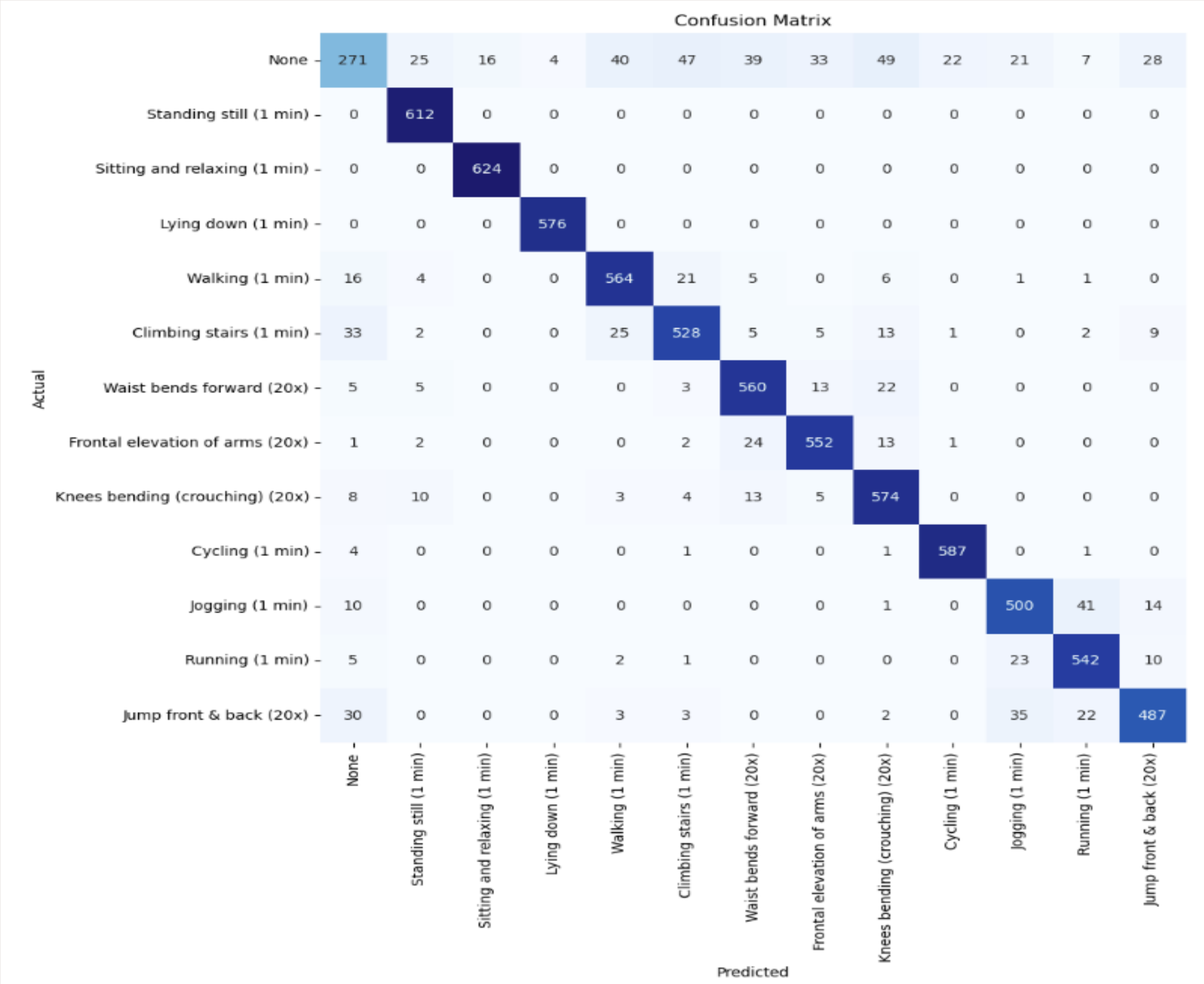
Classification Report:

----

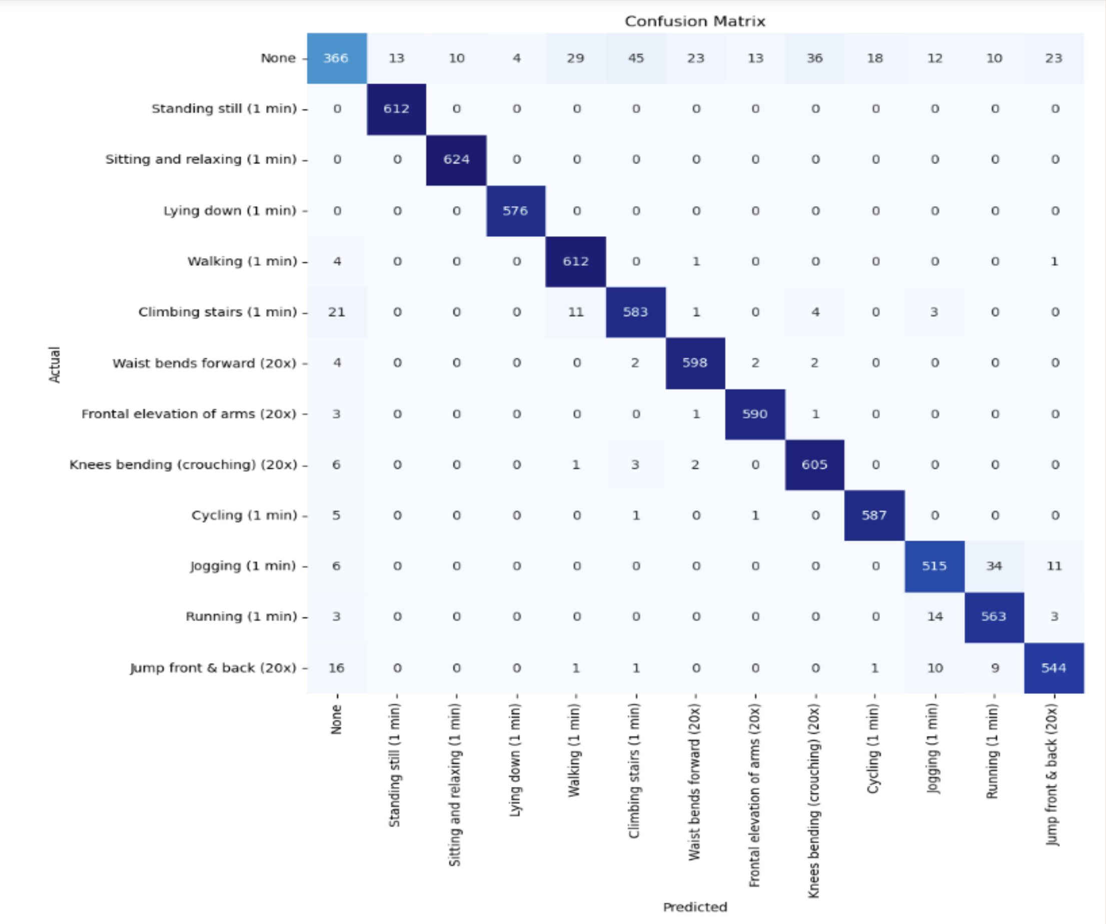
	precision	recall	f1-score	support
None	0.11	0.04	0.06	602
Standing still (1 min)	0.49	0.65	0.56	612
Sitting and relaxing (1 min)	0.56	0.57	0.56	624
Lying down (1 min)	0.96	1.00	0.98	576
Walking (1 min)	0.46	0.48	0.47	618
Climbing stairs (1 min)	0.43	0.26	0.32	623
Waist bends forward (20x)	0.45	0.65	0.53	608
Frontal elevation of arms (20x)	0.65	0.59	0.62	595
Knees bending (crouching) (20x)	0.49	0.56	0.52	617
Cycling (1 min)	0.75	0.91	0.82	594
Jogging (1 min)	0.55	0.55	0.55	566
Running (1 min)	0.52	0.56	0.54	583
Jump front & back (20x)	0.39	0.31	0.35	582
accuracy			0.55	7800
macro avg	0.52	0.55	0.53	7800
weighted avg	0.52	0.55	0.53	7800

Models Used : Long short-term memory (LSTM)

Test Accuracy: 89.45%



Models Used : XG-Boost Classifier




```
Test Accuracy: 94.55%
```

```
Classification Report:
----------------------
                                precision    recall  f1-score   support

                          None       0.84      0.61      0.71       602
        Standing still (1 min)       0.98      1.00      0.99       612
  Sitting and relaxing (1 min)       0.98      1.00      0.99       624
            Lying down (1 min)       0.99      1.00      1.00       576
               Walking (1 min)       0.94      0.99      0.96       618
       Climbing stairs (1 min)       0.92      0.94      0.93       623
     Waist bends forward (20x)       0.96      0.98      0.97       608
Frontal elevation of arms (20x)       0.97      0.99      0.98       595
Knees bending (crouching) (20x)       0.93      0.98      0.96       617
               Cycling (1 min)       0.97      0.99      0.98       594
               Jogging (1 min)       0.93      0.91      0.92       566
               Running (1 min)       0.91      0.97      0.94       583
       Jump front & back (20x)       0.93      0.93      0.93       582

                      accuracy                           0.95      7800
                     macro avg       0.94      0.95      0.94      7800
                  weighted avg       0.94      0.95      0.94      7800
```

Kaggle Dataset 2

	Unnamed: 0	X1	age	gender	height	weight	steps	hear_rate	calories	distance	entropy_heart	entropy_setps	resting_heart	corr_heart_steps
0	1	1	20	1	168.0	65.4	10.771429	78.531302	0.344533	0.008327	6.221612	6.116349	59.0	1.000000
1	2	2	20	1	168.0	65.4	11.475325	78.453390	3.287625	0.008896	6.221612	6.116349	59.0	1.000000
2	3	3	20	1	168.0	65.4	12.179221	78.540825	9.484000	0.009466	6.221612	6.116349	59.0	1.000000
3	4	4	20	1	168.0	65.4	12.883117	78.628260	10.154556	0.010035	6.221612	6.116349	59.0	1.000000
4	5	5	20	1	168.0	65.4	13.587013	78.715695	10.825111	0.010605	6.221612	6.116349	59.0	0.982816

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6264 entries, 0 to 6263
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Unnamed: 0            6264 non-null   int64
 1   X1                    6264 non-null   int64
 2   age                   6264 non-null   int64
 3   gender                6264 non-null   int64
 4   height                6264 non-null   float64
 5   weight                6264 non-null   float64
 6   steps                 6264 non-null   float64
 7   hear_rate             6264 non-null   float64
 8   calories              6264 non-null   float64
 9   distance              6264 non-null   float64
 10  entropy_heart         6264 non-null   float64
 11  entropy_setps         6264 non-null   float64
 12  resting_heart         6264 non-null   float64
 13  corr_heart_steps      6264 non-null   float64
 14  norm_heart            6264 non-null   float64
 15  intensity_karvonen    6264 non-null   float64
 16  sd_norm_heart         6264 non-null   float64
 17  steps_times_distance  6264 non-null   float64
 18  device                6264 non-null   object
 19  activity              6264 non-null   object
dtypes: float64(14), int64(4), object(2)
memory usage: 978.9+ KB
```

norm_heart	intensity_karvonen	sd_norm_heart	steps_times_distance	device	activity
19.531302	0.138520	1.000000	0.089692	apple watch	Lying
19.453390	0.137967	1.000000	0.102088	apple watch	Lying
19.540825	0.138587	1.000000	0.115287	apple watch	Lying
19.628260	0.139208	1.000000	0.129286	apple watch	Lying
19.715695	0.139828	0.241567	0.144088	apple watch	Lying

Feature Extraction (Data preprocessing)

```
df = data.drop(['device', 'Unnamed: 0', 'X1', 'age', 'gender', 'entropy_heart', 'entropy_setps',
                'corr_heart_steps', 'norm_heart', 'intensity_karvonen', 'sd_norm_heart', 'steps_times_distance',
                'hear_rate'], axis=1)
```

```
# Data Splitting
target = df['activity']
feature = df.drop(columns='activity')
X_train, X_test, y_train, y_test = train_test_split(feature, target, test_size=0.2, random_state=0)
```

Models for prediction we used

1. Decision Tree Model
2. K-NN Model
3. Naïve Bayes Model
4. Random forest Model
5. Logistic Regression Model
6. SVM Model
7. XGBoost Model

```
Training and Evaluating Decision Tree Model:
Accuracy: 0.7086991221069433

Training and Evaluating K-NN Model:
Accuracy: 0.6009577015163607

Training and Evaluating Naive Bayes Model:
Accuracy: 0.2641660015961692

Training and Evaluating Random Forest Model:
Accuracy: 0.7462090981644054

Training and Evaluating Logistic Regression Model:
Accuracy: 0.2649640861931365

Training and Evaluating SVM Model:
Accuracy: 0.2833200319233839

Training and Evaluating XGBoost Model:
Accuracy: 0.7438148443735035
```

conclusion

Challenges we faced

- Collection of real time student data
- Data Preprocessing was one of the challenge

Thank you