Human Activity Recognition using Smartphone Data

Minh Vo

MSCA 31009 - Machine Learning & Predictive Analytics

Final Project

AGENDA

- Problem Statement
- Data Source
- EDA
- Assumptions
- Data Preprocessing
- Modeling
- Model Selection
- Future Work

PROBLEM STATEMENTS

Human Activity Recognition (HAR)

- HAR collects data from smart devices' sensors, allowing for human movement observation using machine learning/deep learning methods.
- It plays an essential role in many aspects of people's life, including healthcare, sports, lifestyle monitoring, etc.
- The challenge lies in effectively analyzing and interpreting the vast amount of sensor data collected from smartphones to identify different activities.

Project Objectives

- Using data collected from smartphone sensors to classify 6 primary human activities.
- Experimenting various deep learning models to identify the best model that can accurately perform activity classification.

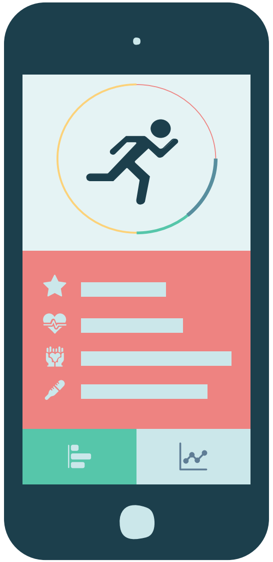
DATA DESCRIPTION

Data Source

- UCI Machine Learning Repository.
- Recorded from 30 volunteers performing daily-living activities while wearing a waist-mounted smartphone equipped with an embedded accelerometer and gyroscope.
- Train data: 7352 records (70%)
  Test data: 2947 records (30%)

Target Variable: ACTIVITY

- WALKING
- WALKING_UPSTAIRS, WALKING_DOWNSTAIRS
- SITTING
- STANDING
- LAYING.



Features

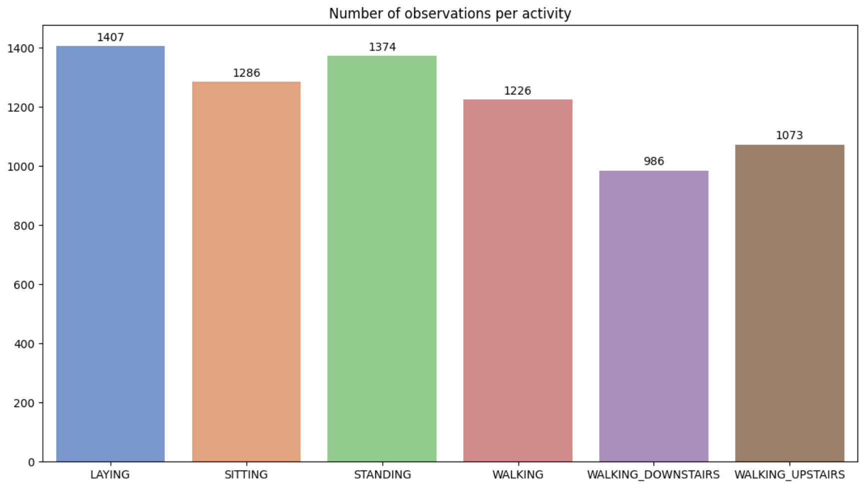
- 561-feature vector that combines time and frequency domain variables are from the sensor signals.
- Provide a comprehensive representation of the data and serve as input for the activity classification task.

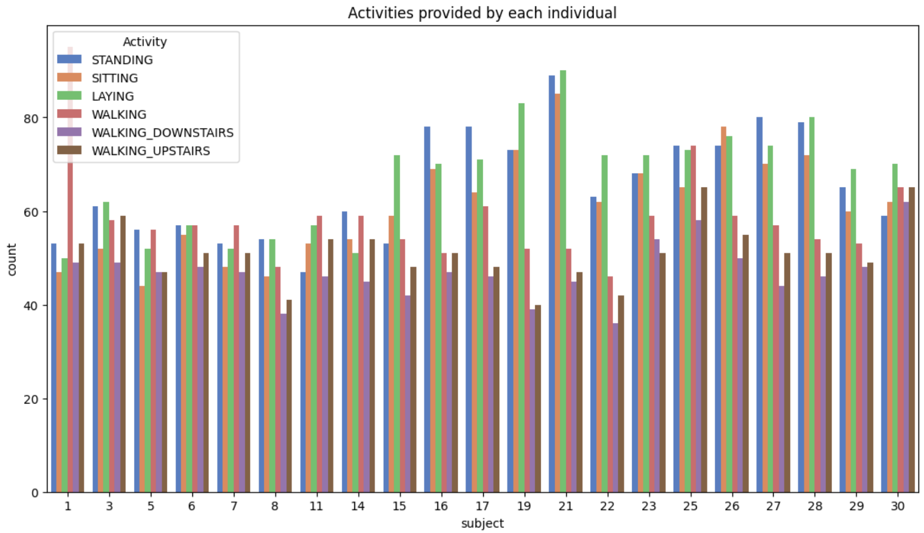
Generating Process

- The sensors captured 3-axial linear acceleration (accelerometer data) & 3-axial angular velocity (gyroscopic data) at a constant rate of 50Hz.
- Jerk signals are for *BodyAcceleration* readings.
- By combining the measurements from the accelerometer and gyroscope, the dataset provides both the total acceleration and the estimated body acceleration.

https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

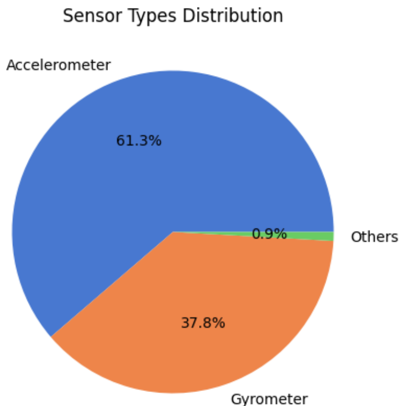
EXPLORATORY DATA ANALYSIS



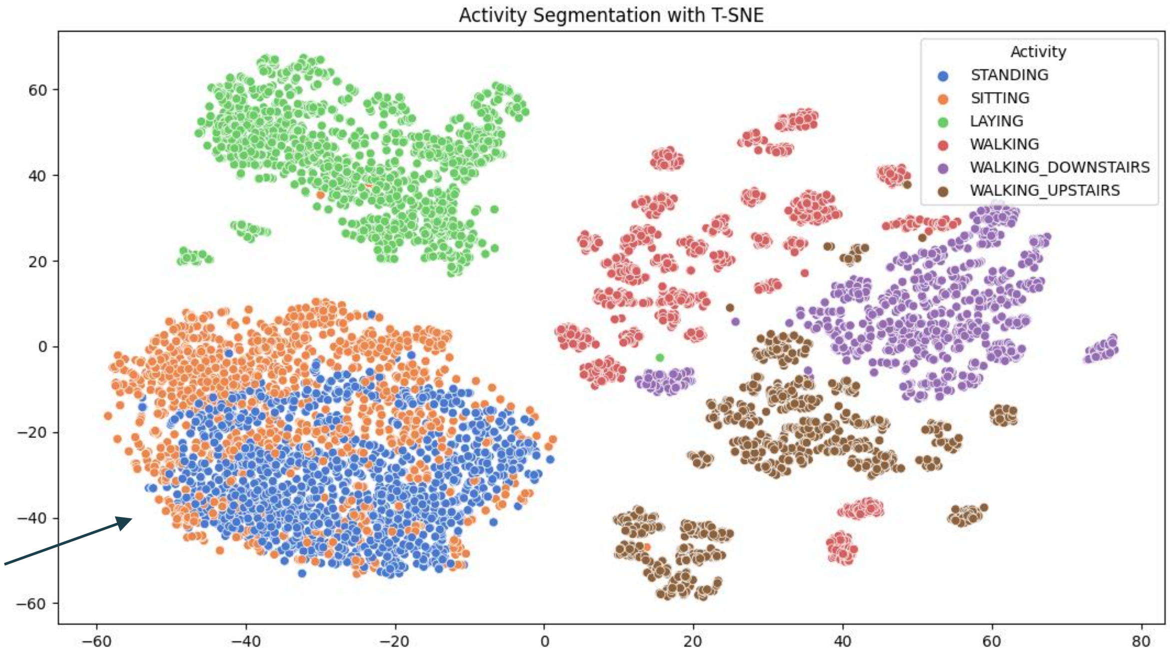


- <u>Static Activities:</u> SITTING, STANDING, LAYING
- <u>Dynamic Activities:</u> WALKING, WALKING_DOWNSTAIRS, WALKING_UPSTAIRS
- Static Activities (*Laying* and *Standing*) are performed the most.
- Most of activities/movements were recorded from the accelerometer sensors.

There seems to be some class imbalance. This may affect the model's classification performance.



EDA (cont.)



Six activities tend to be mostly separable between static activities (Standing, Sitting, Laying) and dynamic activities (Walking, Walking Downstairs & Upstairs), **except Standing and Sitting.**

DATA & MODEL ASSUMPTIONS

DATA ASSUMPTIONS

- The dataset represents a diverse range of participants in terms of age and activity patterns.
- The participants perform the activities as instructed and in a consistent manner.
- The gravitational force is assumed to have only low frequency components, so a filter with 0.3 Hz cutoff frequency was used to capture sensor data.*
- The 561 selected features are relevant to the classification task and capture important patterns of the activities.

MODEL ASSUMPTIONS

- Deep learning models, such as MLP, CNNs or LSTMs, have demonstrated strong performance in various pattern recognition tasks.
- Given the complexity and sequential nature of the sensor data, it is hypothesized that deep learning models will capture intricate patterns and dependencies.

*https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

DATA PREPROCESSING

1. Train – Test

- Target: “Activity”
- Training data size: (7352, 561)
- Test data size: (2947, 561)

2. Data Encoding

Encode 6 activities in “Activity” column:

- **0:** LAYING
- **1:** SITTING
- **2:** STANDING
- **3:** WALKING
- **4:** WALKING_DOWNSTAIRS
- **5:** WALKING_UPSTAIRS

3. Data Scaling

- Many input features have varying scales and ranges.
- Use MinMaxScaler() to scale the features to a consistent range.

MODELING

01 BASE MODEL - DNN

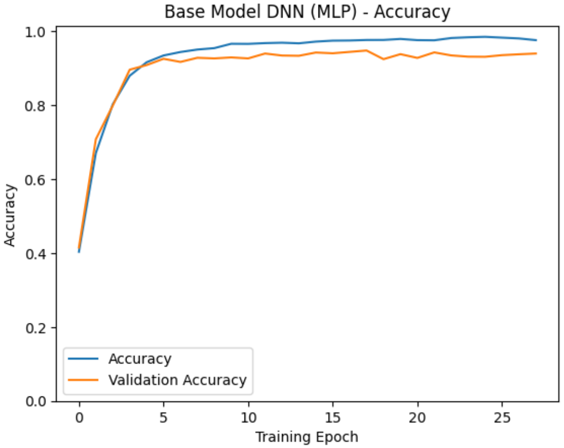
02 CNN MODEL

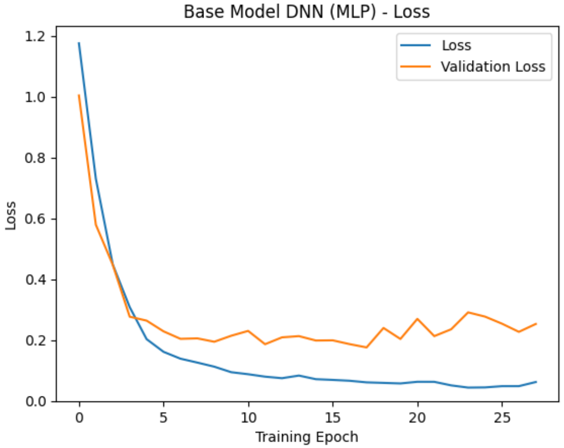
03 LSTM MODEL

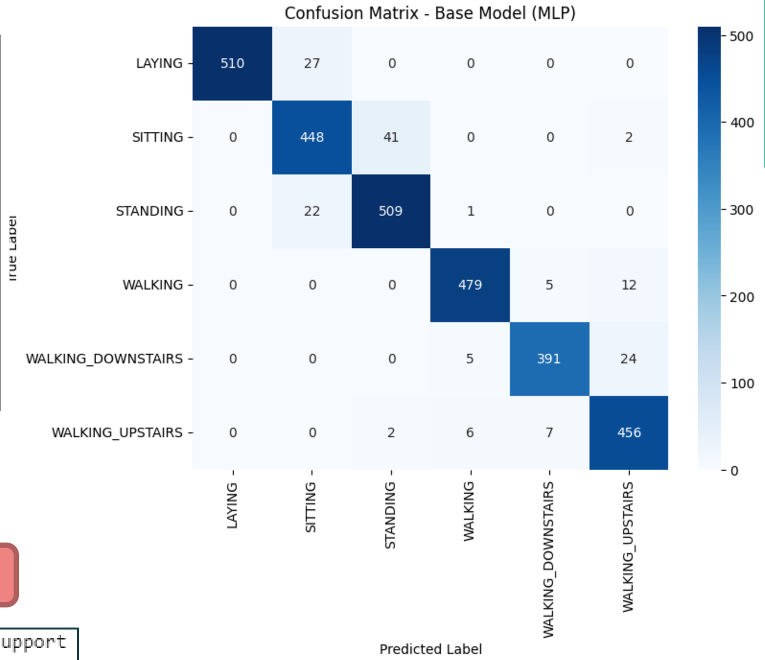
04 CNN + LSTM MODEL

- Apply EarlyStopping at every model to reduce overfitting.
- Reshape data before fitting CNN, LSTM, and CNN + LSTM models.

BASE MODEL – MLP







Accuracy: 0.9477 ; Loss: 0.176

Classification Report - Test

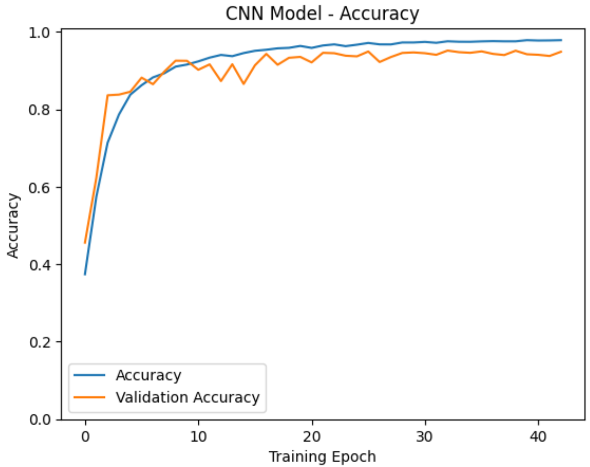
	precision	recall	f1-score	support
0	1.00	0.95	0.97	537
1	0.90	0.91	0.91	491
2	0.92	0.96	0.94	532
3	0.98	0.97	0.97	496
4	0.97	0.93	0.95	420
5	0.92	0.97	0.95	471
accuracy			0.95	2947
macro avg	0.95	0.95	0.95	2947
weighted avg	0.95	0.95	0.95	2947

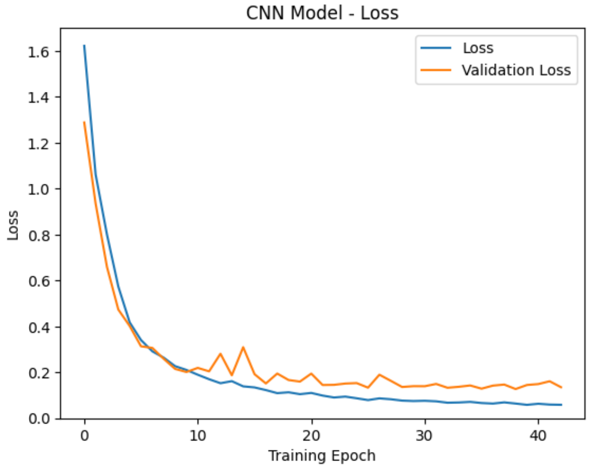
Classification Report - Train

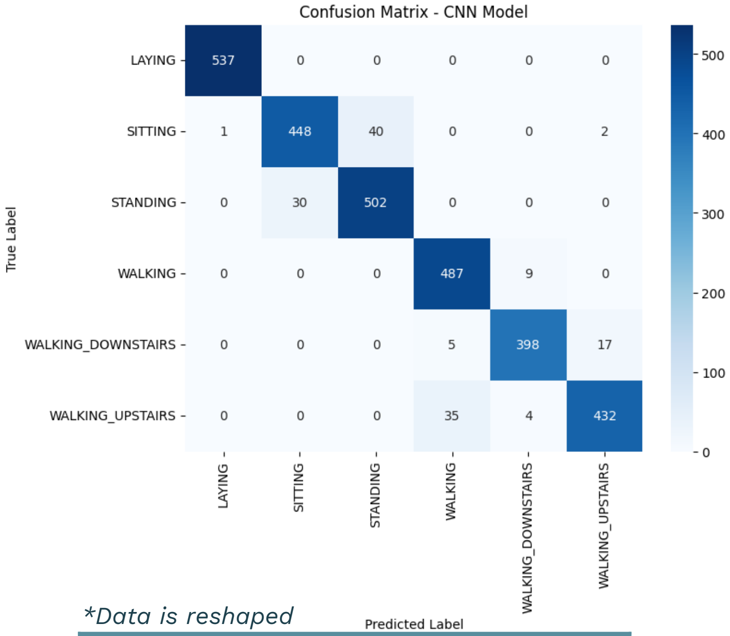
	precision	recall	f1-score	support
0	0.99	1.00	1.00	1407
1	0.95	0.96	0.96	1286
2	0.97	0.96	0.96	1374
3	1.00	0.99	1.00	1226
4	0.98	1.00	0.99	986
5	0.99	0.99	0.99	1073
accuracy			0.98	7352
macro avg	0.98	0.98	0.98	7352
weighted avg	0.98	0.98	0.98	7352

Layers	Output Shape
Dense	(None, 64)
Dense	(None, 64)
Dense	(None, 64)
Dense	(None, 6)

CNN MODEL







Accuracy: 0.9515 ; Loss: 0.1328

Data is reshaped

Classification Report - Test

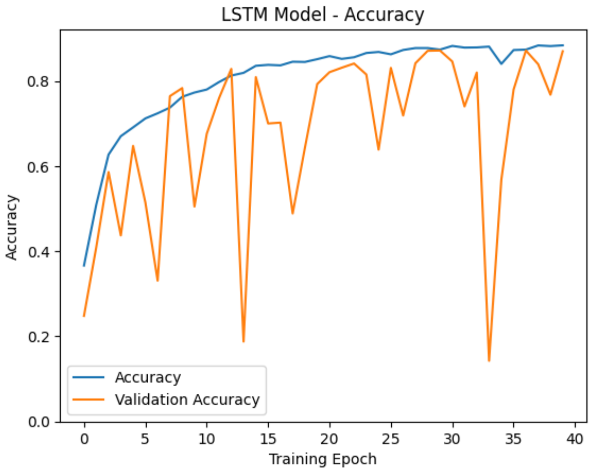
	precision	recall	f1-score	support
0	1.00	1.00	1.00	537
1	0.94	0.91	0.92	491
2	0.93	0.94	0.93	532
3	0.92	0.98	0.95	496
4	0.97	0.95	0.96	420
5	0.96	0.92	0.94	471
accuracy			0.95	2947
macro avg	0.95	0.95	0.95	2947
weighted avg	0.95	0.95	0.95	2947

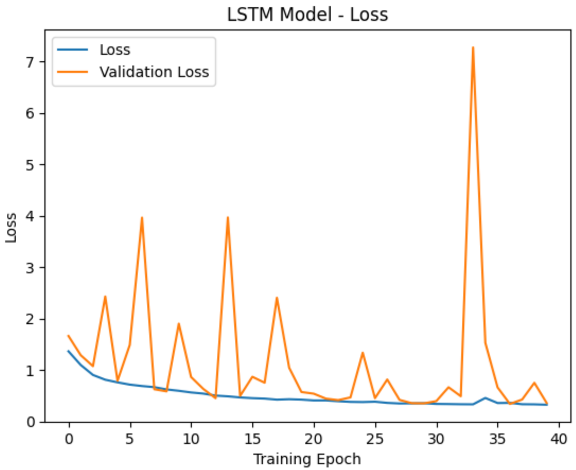
Classification Report - Train

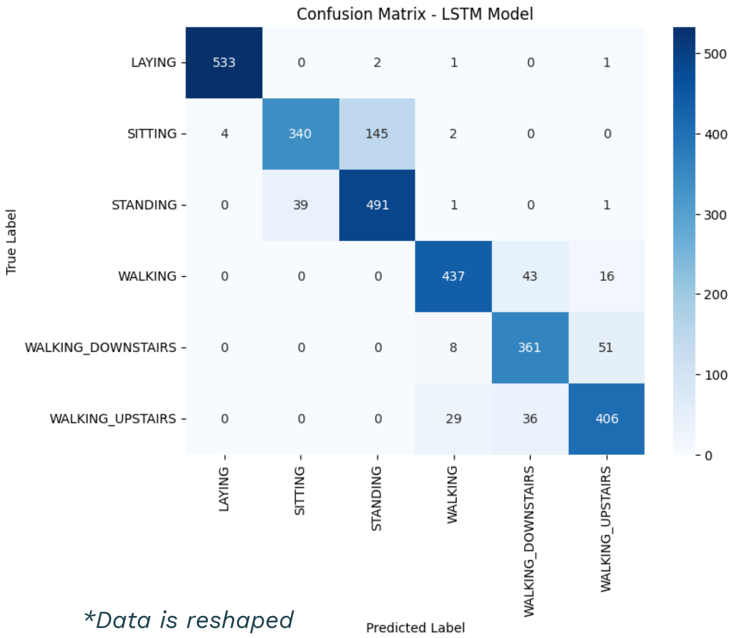
	precision	recall	f1-score	support
0	1.00	1.00	1.00	1407
1	0.95	0.96	0.95	1286
2	0.96	0.95	0.96	1374
3	0.99	1.00	1.00	1226
4	1.00	1.00	1.00	986
5	1.00	0.99	1.00	1073
accuracy			0.98	7352
macro avg	0.98	0.98	0.98	7352
weighted avg	0.98	0.98	0.98	7352

Layers	Output Shape
Conv1D	(None, 559, 64)
Conv1D	(None, 557, 32)
Dropout	(None, 557, 32)
Flatten	(None, 17824)
Dense	(None, 64)
Dense	(None, 128)
Dense	(None, 6)

LSTM MODEL







Accuracy: 0.8714 ; Loss: 0.353

Data is reshaped

Classification Report - Test

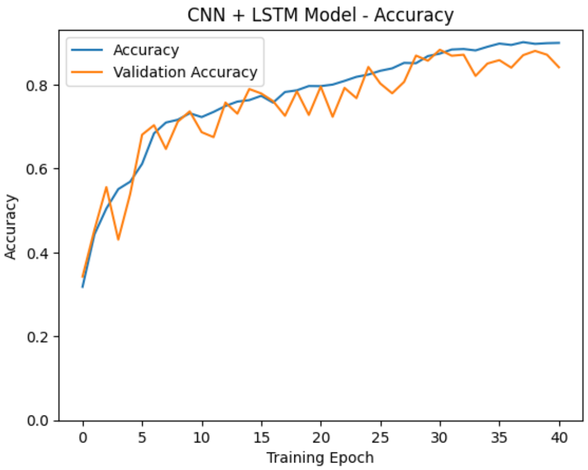
	precision	recall	f1-score	support
0	0.99	0.99	0.99	537
1	0.90	0.69	0.78	491
2	0.77	0.92	0.84	532
3	0.91	0.88	0.90	496
4	0.82	0.86	0.84	420
5	0.85	0.86	0.86	471
accuracy			0.87	2947
macro avg	0.87	0.87	0.87	2947
weighted avg	0.88	0.87	0.87	2947

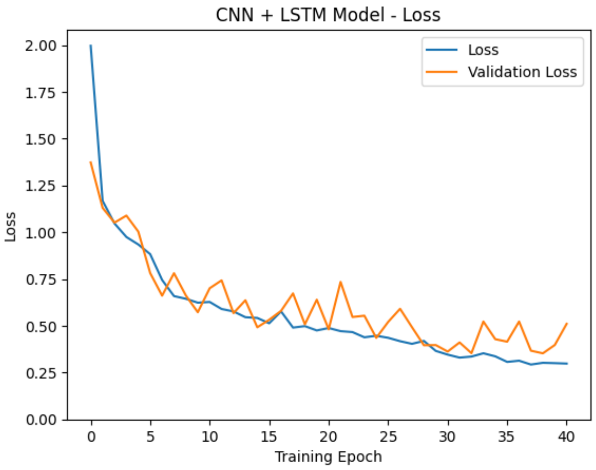
Classification Report - Train

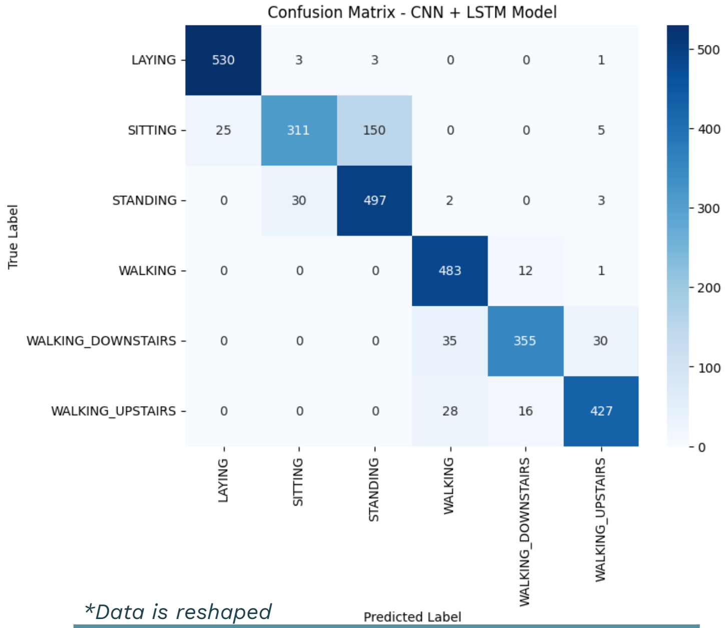
	precision	recall	f1-score	support
0	0.99	0.99	0.99	1407
1	0.91	0.77	0.83	1286
2	0.82	0.93	0.87	1374
3	0.96	0.89	0.93	1226
4	0.84	0.95	0.89	986
5	0.88	0.87	0.88	1073
accuracy			0.90	7352
macro avg	0.90	0.90	0.90	7352
weighted avg	0.90	0.90	0.90	7352

Layers	Output Shape
LSTM	(None, 561, 48)
BatchNormalization	(None, 561, 48)
Dropout	(None, 561, 48)
LSTM	(None, 32)
Dense	(None, 32)
Dense	(None, 6)

CNN + LSTM MODEL







Accuracy: 0.8833 ; Loss: 0.3622

Data is reshaped

Classification Report - Test

	precision	recall	f1-score	support
0	0.95	0.99	0.97	537
1	0.90	0.63	0.74	491
2	0.76	0.93	0.84	532
3	0.88	0.97	0.93	496
4	0.93	0.85	0.88	420
5	0.91	0.91	0.91	471
accuracy			0.88	2947
macro avg	0.89	0.88	0.88	2947
weighted avg	0.89	0.88	0.88	2947

Classification Report - Train

	precision	recall	f1-score	support
0	0.96	0.99	0.98	1407
1	0.92	0.75	0.83	1286
2	0.83	0.94	0.88	1374
3	0.96	0.98	0.97	1226
4	0.97	0.91	0.94	986
5	0.92	0.96	0.94	1073
accuracy			0.92	7352
macro avg	0.93	0.92	0.92	7352
weighted avg	0.93	0.92	0.92	7352

Layers	Output Shape
Conv1D	(None, 559, 64)
Conv1D	(None, 557, 128)
Dropout	(None, 557, 128)
MaxPooling1D	(None, 278, 128)
LSTM	(None, 64)
Dropout	(None, 64)
Dense	(None, 128)
Dense	(None, 6)

MODEL COMPARISON

Models' Accuracy & Loss

#	MODEL	ACCURACY	LOSS
1	Dense (MLP)	0.9477	0.176
2	CNN	0.9515	0.1328
3	LSTM	0.8714	0.353
4	CNN + LSTM	0.8833	0.3622

Models' F-1 Score by Activities

#	ACTIVITY	Dense (MLP)	CNN	LSTM	CNN + LSTM
0	**LAYING**	0.97	1.00	0.99	0.97
1	SITTING	0.91	0.92	0.78	0.74
2	STANDING	0.94	0.93	0.84	0.84
3	WALKING	0.97	0.95	0.90	0.93
4	WALKING_ DOWNSTAIRS	0.95	0.96	0.84	0.88
5	WALKING_ UPSTAIRS	0.95	0.94	0.86	0.91

CONCLUSION & FUTURE WORK

CONCLUSION and LESSONS LEARNED:

- CNN model appears to be the most effective for this human activity recognition project.
- LSTM and CNN+LSTM models need further optimization to improve their training convergence.
- Models tend to perform well in predicting instances of *Laying* activity. This is probably because the observations of Laying is the highest among 6 activities.
- All models still encounter overfitting despite having relatively high accuracy.
- Fitting LSTM model takes the longest time.

FUTURE WORK

- Focus on optimizing and fine-tuning the CNN architecture to improve the model's performance.
- Experiment with more different batch size, number of epochs, and size of the filters in the layers.
- Perform hyperparameter tunning (using Random Search or Bayesian Optimization) to find the best combination of hyperparameters for each model.
- Apply data augmentation to increase the size of training dataset and enhance the models' generalization.
- Explore other types of models.

THANK YOU

REFERENCES

1. https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones
2. https://machinelearningmastery.com/how-to-model-human-activity-from-smartphone-data/
3. https://github.com/tam-ng/Human_Activity_Recognition/blob/master/Presentation_HAR.pdf
4. https://www.kaggle.com/code/vikashrajluhaniwal/eda-all-classification-algorithms-with-96-acc
5. https://www.kaggle.com/code/morrisb/what-does-your-smartphone-know-about-you/notebook
6. https://www.kaggle.com/code/rakibulnahin/visualization-and-96-accuracy-on-human-activity#Deep-Learning
7. https://towardsdatascience.com/human-activity-recognition-har-tutorial-with-keras-and-core-ml-part-1-8c05e365dfa0