ELEC 292 Project Report

Group Number: 69

Andrew Poyner, 20425805, 23tql@queensu.ca

Ben Malvern, 20424655, 23djk@queensu.ca

Clarke Needles, 20424709, 23gpb@queensu.ca

April 4, 2025

Table of Contents

List of Figures

List of Tables

1. Data Collection

Using the personal cell phones of two of the group members running the recommended Phyphox app, each member spent two minutes and thirty seconds walking, followed by the same amount of time spent jumping. One person held the phone in his hand, another placed it in his front pocket, and the third in his back pocket.

Commented [AP1]: Clarke, is it ok if I make some grammar/verb tense/sentence structure edits in section 4?

Commented [CN2R1]: yep

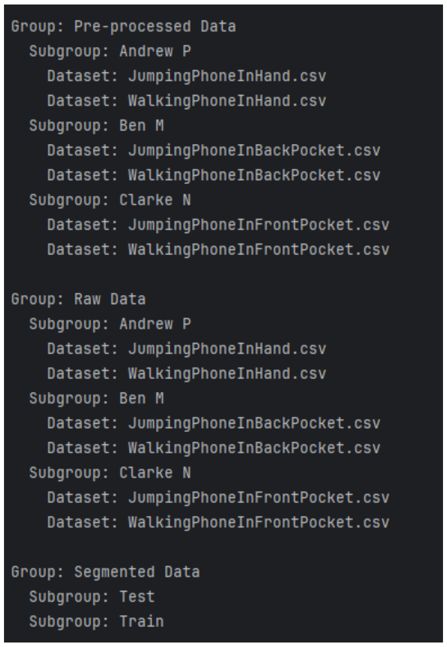
Commented [AP3R1]: Are you ready to submit now?

Each Phyphox output file was labelled immediately after collection by naming the file according to the phone position and the action performed. The data was then transferred into a .csv format to be sent to a laptop, however two of the files ended up as tab separated values formats, so the group had to alter the file reading section of the Python code to account for this.

2. Data Storage

As per the project instructions, the collected data was stored in an HDF5 file with three main branches, one for the raw, unprocessed data exactly as it was originally collected, one for the pre-processed data, and one branch for the segmented data that was split into five-second segments and into training and testing subsets. Shown in Figure 1 is the output representing the structure of the HDF5 file.

Commented [AP4]: Is the picture in section 2 the one you're referring to in the second section 2 bullet point?



```
Group: Pre-processed Data
  Subgroup: Andrew P
    Dataset: JumpingPhoneInHand.csv
    Dataset: WalkingPhoneInHand.csv
  Subgroup: Ben M
    Dataset: JumpingPhoneInBackPocket.csv
    Dataset: WalkingPhoneInBackPocket.csv
  Subgroup: Clarke N
    Dataset: JumpingPhoneInFrontPocket.csv
    Dataset: WalkingPhoneInFrontPocket.csv

Group: Raw Data
  Subgroup: Andrew P
    Dataset: JumpingPhoneInHand.csv
    Dataset: WalkingPhoneInHand.csv
  Subgroup: Ben M
    Dataset: JumpingPhoneInBackPocket.csv
    Dataset: WalkingPhoneInBackPocket.csv
  Subgroup: Clarke N
    Dataset: JumpingPhoneInFrontPocket.csv
    Dataset: WalkingPhoneInFrontPocket.csv

Group: Segmented Data
  Subgroup: Test
  Subgroup: Train
```

Figure 1: HDF5 File Structure

3. Visualization

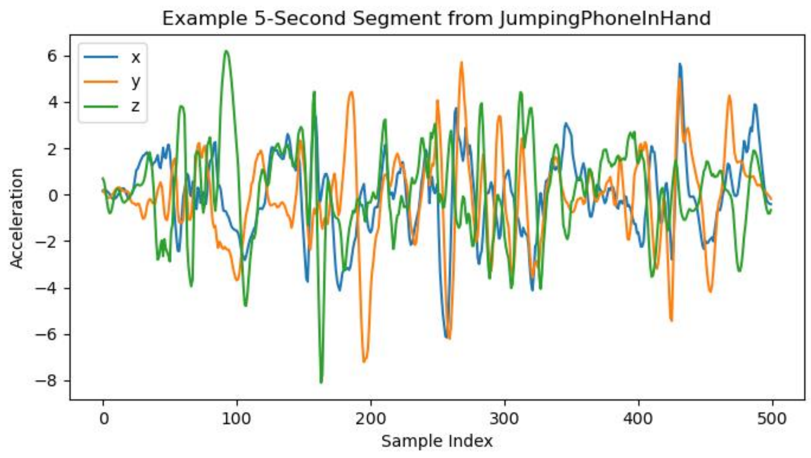


Figure 2: Segment of Data from Jumping While Holding the Phone

Figure 2 shows all three axes of motion for a five-second segment of data collection when the person was jumping with the phone held in his hand. As expected, it shows irregular spikes of acceleration, representing the beginnings and ends of jumps. The below plot shows the predictions made by the model, all of which were correct. If the data collection were to be repeated, it could be done with jumping interspersed within the walking, or with horizontal jumps or jumps over obstacles, to vary the data available to train and test the model, allowing it to function in a wider variety of environments. Another change if the data collection were to be repeated would be to collect data from more people than just the members of the group, to account for different gaits, walking speeds, and body structures.

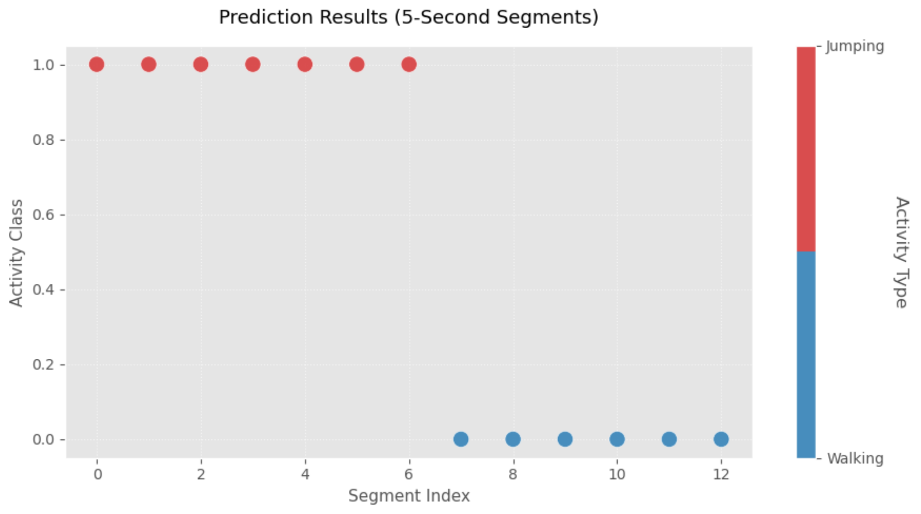


Figure 3: Prediction Results

4. Preprocessing

Firstly, the data was rendered easier to work with by using simpler standardized column names. Then it was ensured that each entry in the dataframe is a numeric value and not a string or other type. For the gaps in the data, linear interpolation as well as forward fill were used. Linear interpolation was chosen over quadratic interpolation because of the risk of overfitting. Furthermore, linear interpolation will work very well especially when walking at a consistent speed, and even when jumping, the jumps were consistent enough that the linear method would work sufficiently well. The quadratic method provided a better approximation for the jumping with gravity; however, the linear method would work better even with noise and thus was less risky. Linear interpolation was used first to fill the larger gaps in the data, then forward fill was used to clean up the remaining gaps in the data [1]. The forward fill essentially acted as padding for the data so that the missing gaps were not dropped, and valuable data was not lost [2]. For the moving average filter, a window of 6 was chosen to maintain the data peaks while still removing noise. This value was changed around a lot, however the final value of 6 seemed the best for reducing noise without too strong of a filter. Figure 4 and Figure 5 below show the effect of the preprocessing effects on the walking and jumping data while holding the phone in hand.

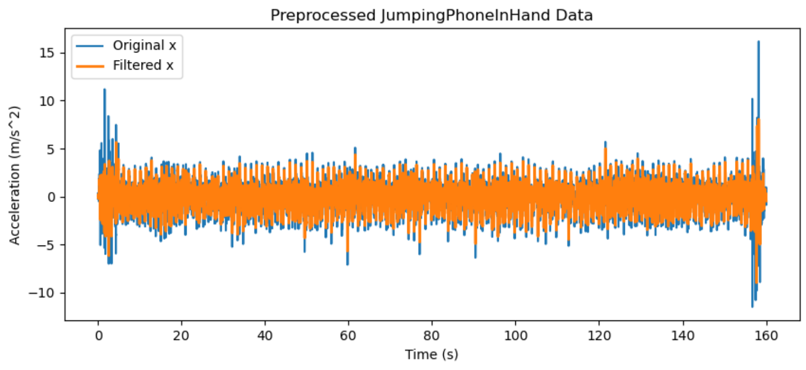


Figure 4: Preprocessing Applied on Jumping Data While Phone is in Hand

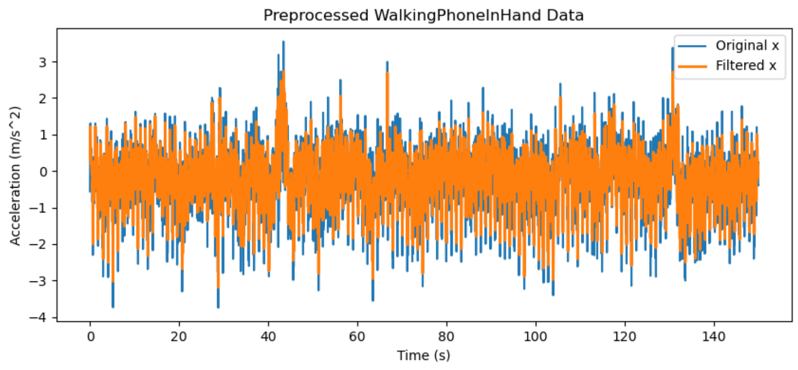


Figure 5: Preprocessing Applied on Jumping Data While Phone is in Hand

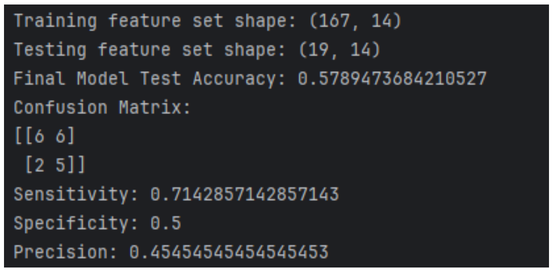
5. Feature Extraction & Normalization

When choosing the features to extract from the data, the considerations that were made were all based on differentiating jumping versus walking data. The key difference between the two motions are the effects on the z-axis. Therefore, things such as max acceleration, max jerk, max, min, mean, and standard deviation are things to consider. Acceleration is important because walking is very consistent, whereas jumping there is a lot of acceleration. Jerk is very similar to acceleration as it shows the effects of pushing off the ground and the jerk that the motion would put on the accelerometer when gathering data. Finding the max and min values is equally as important to detect when someone is standing still versus walking. Finally, in a study done [3], it was concluded that the common features between all axes for accelerometer data were mean and standard deviation. In the same study, they mention that the optimal subset of features was 18. In this situation, max, min, mean, and standard deviation are extracted from each axis. This with the max jerk and acceleration provides a total of 14 features. To try and reach 18 features, other things such as range were considered. However, in this case, range would be redundant because of the max and min features, and while there are many more features that could be included, there is always the curse of dimensionality that must be considered. The issue with too many features is that if it does not improve the normalization of the data, then it could be hurting the model performance.

6. Training and Testing the Classifier

We trained a logistic regression classifier to output a value of 0 to label “walking” or 1 to label “jumping” for a series of 5 second intervals. To ensure that we could both test and train the classifier, the data was split into two groups, “Testing” consisting of 10% of the total data and “Training” which consisted of 90% of the data split. All the data was randomized to ensure a more uniform spread of features while training. We used a fixed random seed of 42 so that the results were reproducible. Before training, a standard scaler was used to ensure all features contribute equally to the model.

Using Scikit-Learn's logistic regression function, we set the max iterations to 1000 to ensure that the model had enough iterations to converge and produce more reliable results. The classifier achieved a test accuracy of 57.9%.



```
Training feature set shape: (167, 14)
Testing feature set shape: (19, 14)
Final Model Test Accuracy: 0.5789473684210527
Confusion Matrix:
[[6 6]
 [2 5]]
Sensitivity: 0.7142857142857143
Specificity: 0.5
Precision: 0.45454545454545453
```

Figure 6: Model Accuracy Percentages

These results mean that while the model is reasonably good at detecting jumping, it struggles with detecting walking. This likely happens when the user starts walking fast or changes direction quickly.

7. Model Deployment

For the model deployment, the GUI was designed to be simplistic yet effective. The final application utilized file loading, file saving, and running the dataset through the training model. The final application was developed using Tkinter and its associated libraries. We ensured smooth workflow, while ensuring that the code was error trapped to stop the app from terminating. This includes things like file related errors, application resizing errors, button errors, and more. The application is split into two sections: load data and run analysis. The load data section is where the user must select a .csv file to be able to use the “Run Activity Prediction” button. Figure 7 depicts the original state of the app.

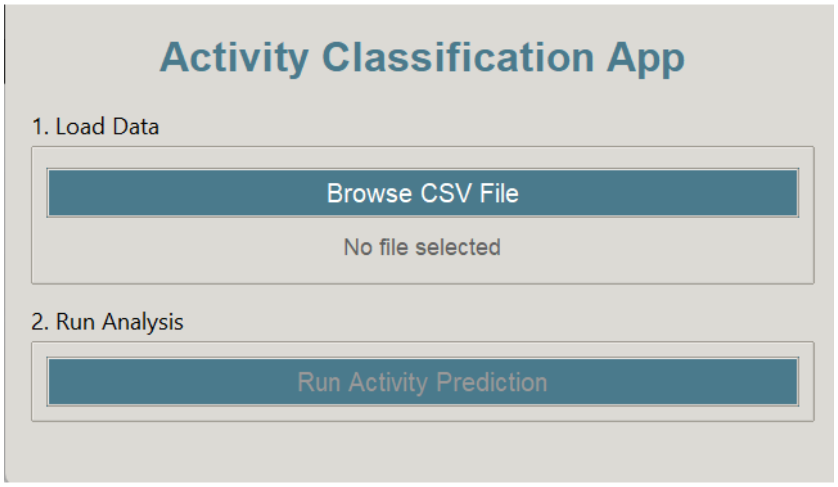


Figure 7: Default State of App

Once a file is selected, the file name is displayed under the "Browse CSB File" button. Now the user can run a prediction. Figure 8 displays this.

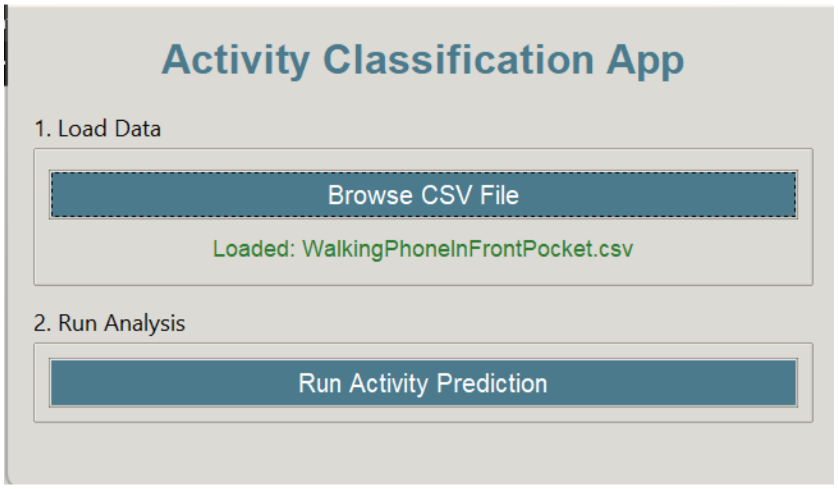


Figure 8: App state after a file has been selected

Once the prediction is run, the user is given the option to save the file as a .csv. Then after the user saves the file, the prediction results are represented graphically. Figure 9 shows an example of the graph style.

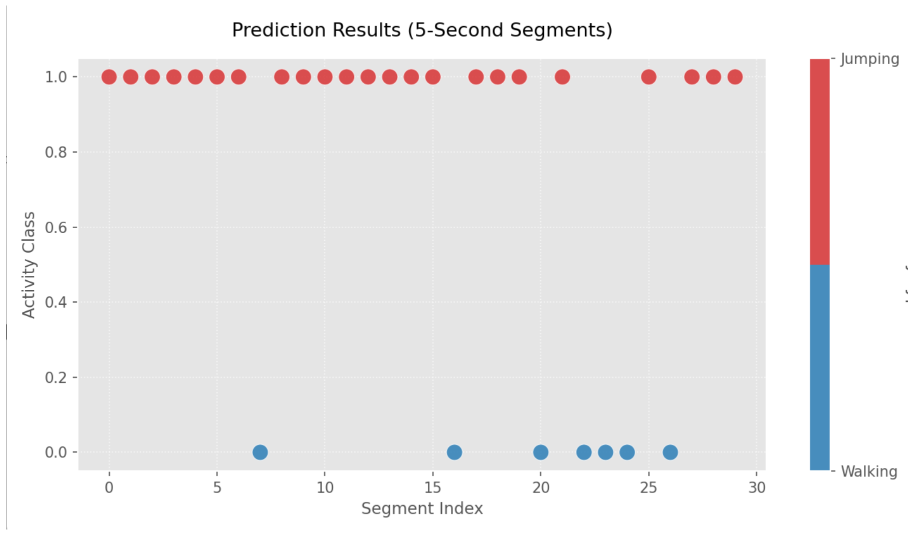


Figure 9: Output Graph for Prediction Results

For the graph, the idea was to once again make the data look as simple as possible. Using minimalist strategy makes the data easier to comprehend. Overall, the application provides the users an easy way to run predictions on their walking/jumping data and get quality results.

8. References

[1] "Developer," Apple, [Online]. Available: https://developer.apple.com/documentation/accelerate/using-linear-interpolation-to-construct-new-data-points. [Accessed 4 4 2025].

[2] whyamit404, "Medium," 12 2 2025. [Online]. Available: https://medium.com/@whyamit404/understanding-forward-fill-ffill-in-pandas-97a40fedab79. [Accessed 4 4 2025].

[3] N. Ahmed, . J. . I. Rafiq and M. R. Islam, "Enhanced Human Activity Recognition Based on Smartphone Sensor Data Using Hybrid Feature Selection Model," 10, 31, 2019.

9. Appendix I: Participation Report

Team member	Estimated Total time spent (hours)
Ben Malvern	(6 report)(12 code)
Andrew Poyner	(6 report)(12 code)
Clarke Needles	(6 report)(12 code)

Table 1: Work Distribution