***Group V Results***

**Thought Process**

Taking a single second of data to identify each person is impossible. If you look at the data, there are a lot of similar measurements, which are probably given through not interacting with the sensor at all. The heel, toe, and meta input combination of every person in a single second can overlap at some points, producing models with an accuracy below 30%.

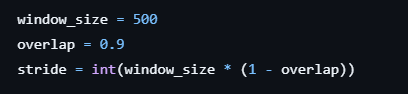
Multiple seconds of continuous input are considered to distinguish a person's trait. There are already ~5200 rows of data in the given data for a 10s time span. Approximately 520 voltages are detected in each second. So, the models use a sliding window approach to extract features from 1 s (~520 rows ) instead of 1 row. However, since the whole dataset is barely 10 seconds long, we will only have 10 data groups even before the T:V:T split. So, without data augmentation, the resulting models are doomed to be biased.

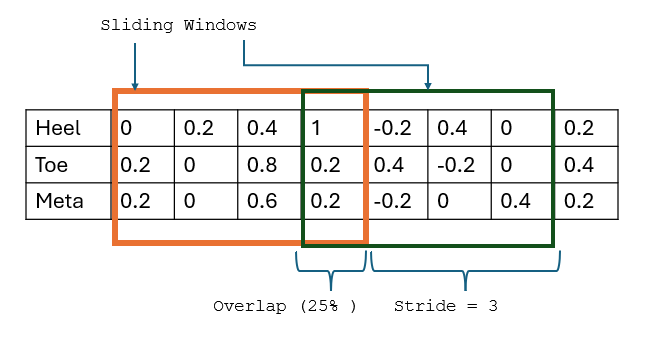
**Preprocessing**

First, we loaded and cleaned data since a few entries didn't have a timestamp. Afterwards, we generate training samples using sliding windows. We then split data into Train / Validate / Test (70, 15, 15) and prepare it for the learning. We then experimented with different values of window size and overlap by running it again and again.

**Sliding Window Method**

A window of fixed size (500 in the image) is moved across that data. The overlap of 0.9 (90%) ensures that each window shares 90% of its sample with the previous one in each move. Stride is a dependent variable based on overlap. After calculating the percentage of reusing data, how far the window is moved each time is calculated.

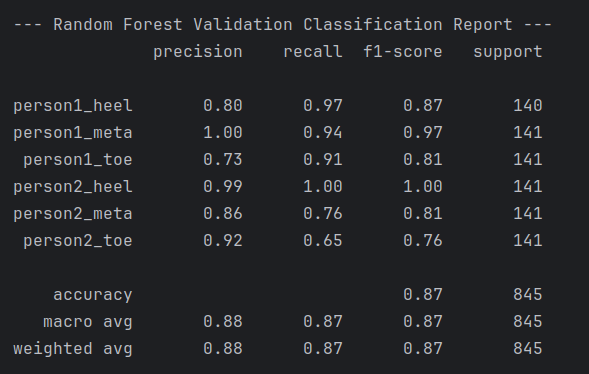


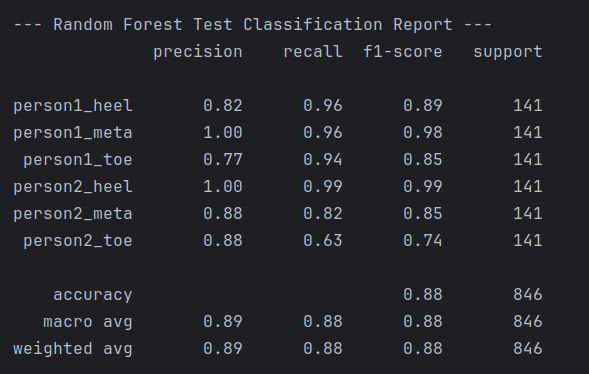
  
Example: Stride = 500 \* (1 - 0.9) = 50, so the window moves 50 steps forward each time.

***Results***

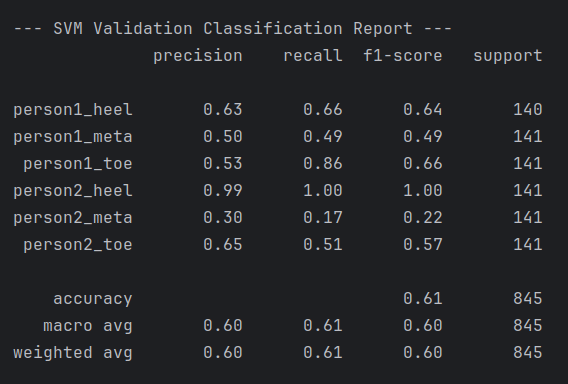
Following are the results of our code presented through their different confusion matrices! Moreover, we included our main.py output for each algorithm used individually! Therefore, one can see the exact accuracy and precision of the models!

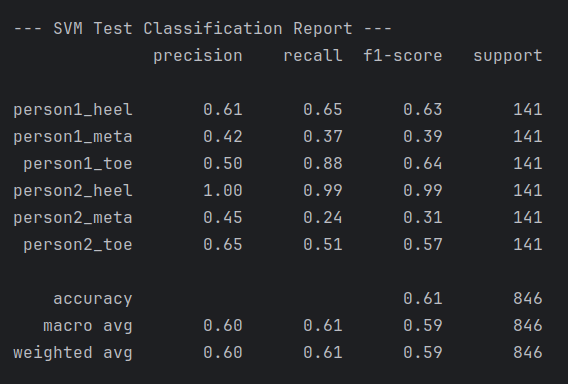
**Random Forest**



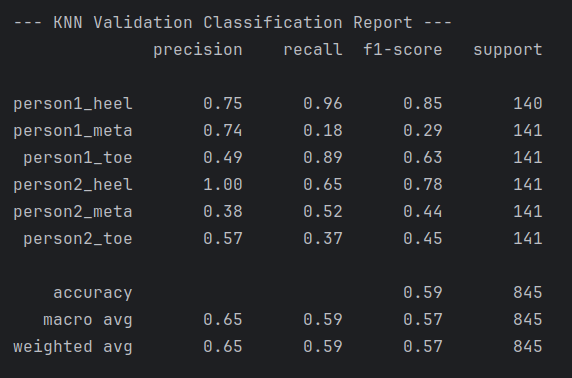


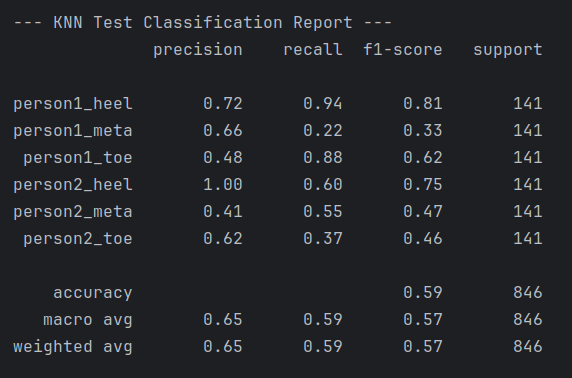
**Support Vector Machine**



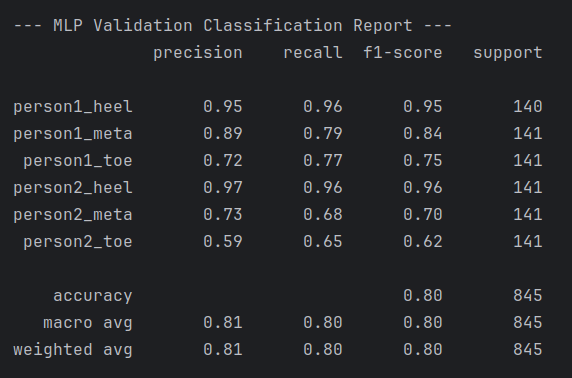


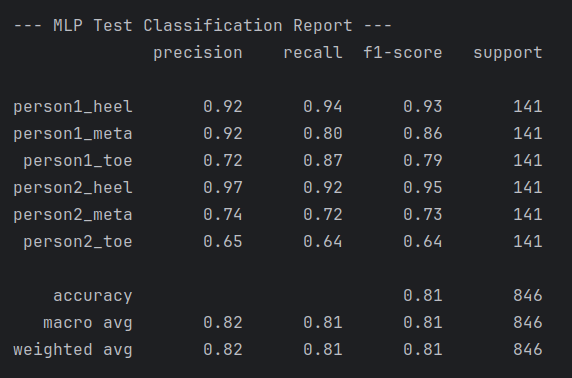
**K-Nearest-Neighbour**



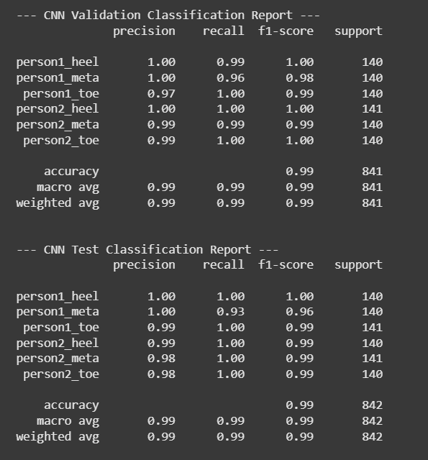


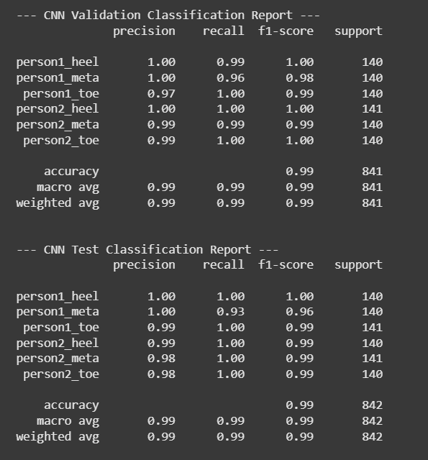
**Multi-Layer-Perceptron**

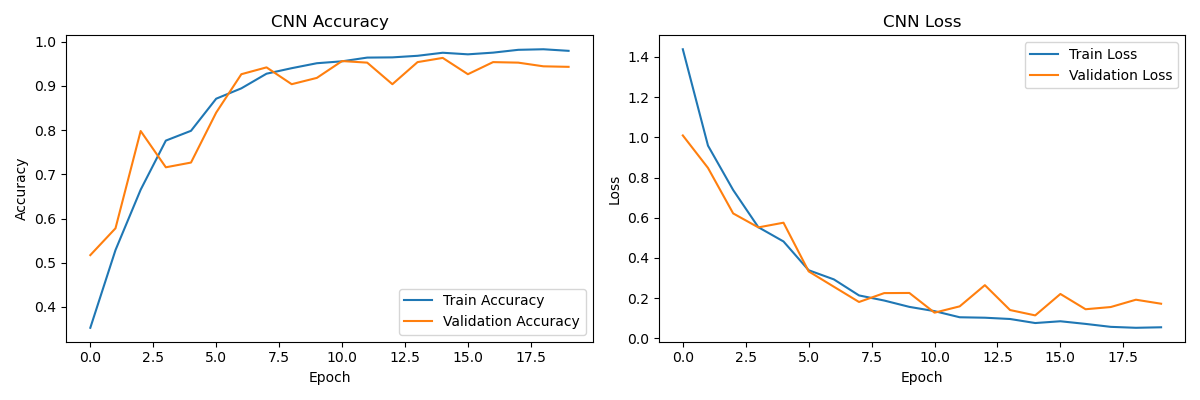
****

****

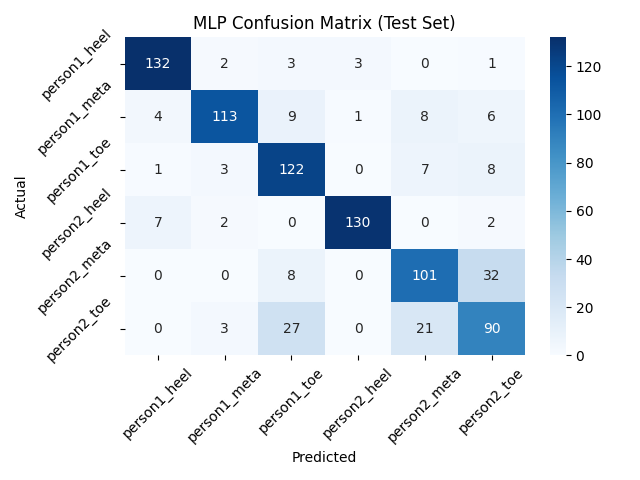
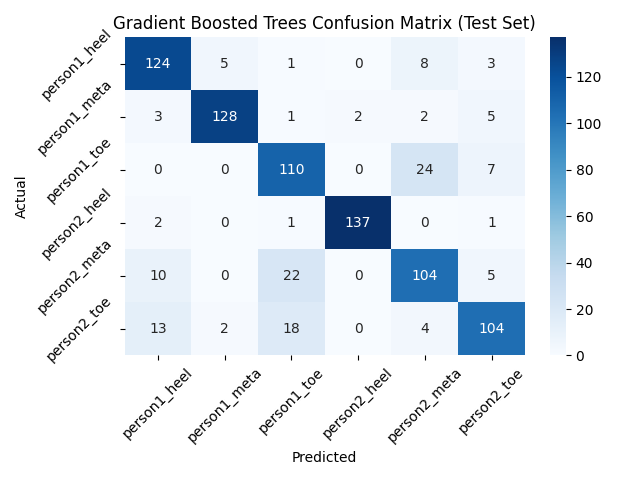
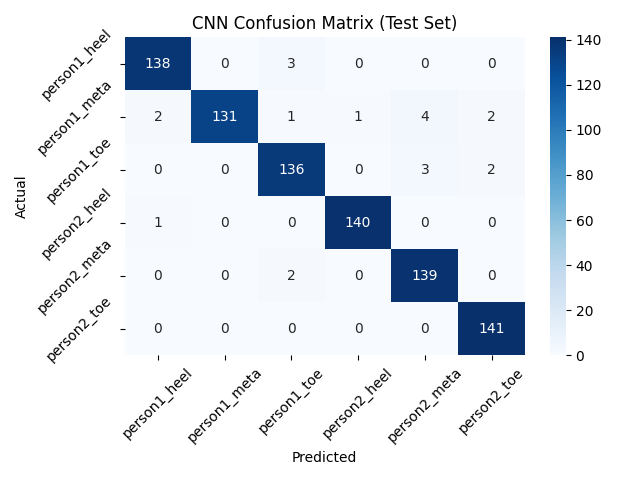
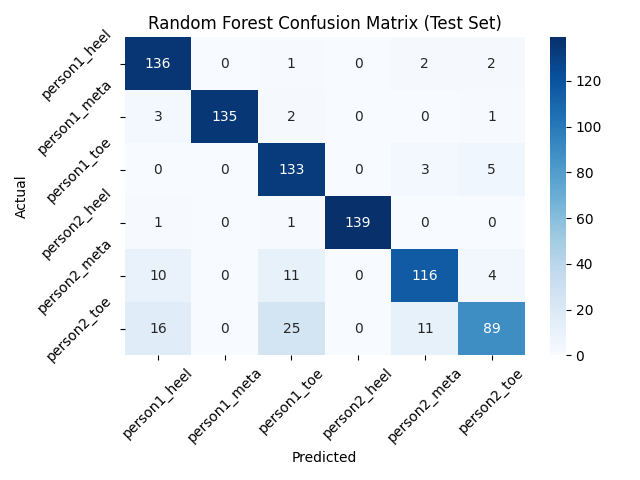
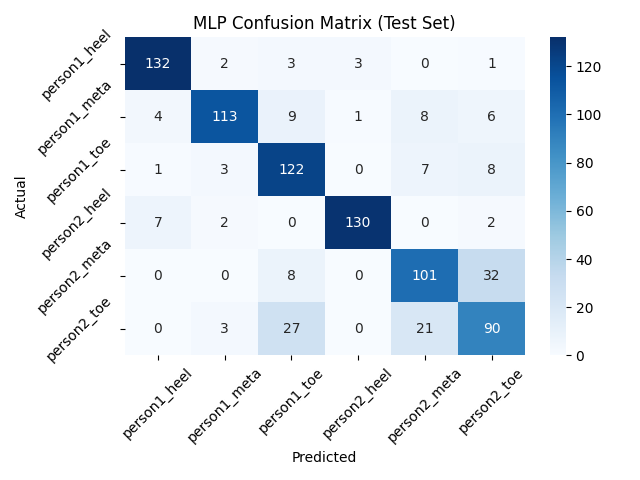
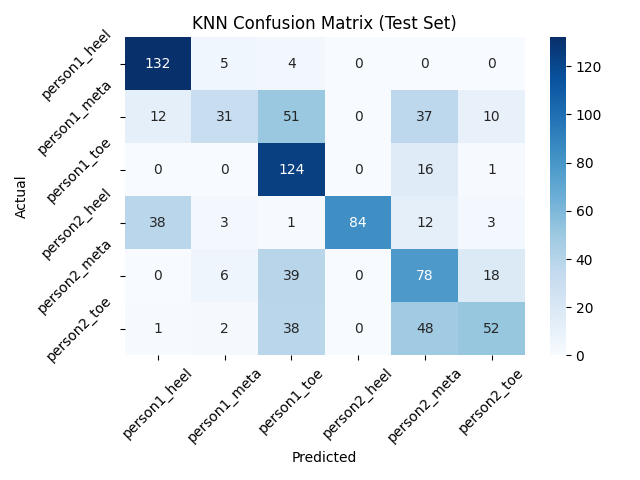
**Convolutional Neural Network**

****

****

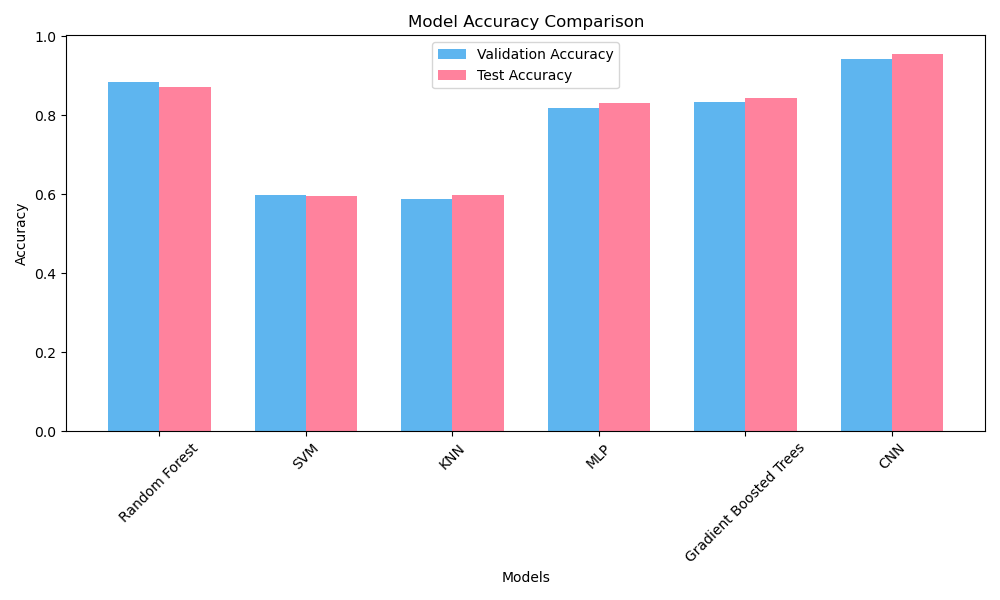
****

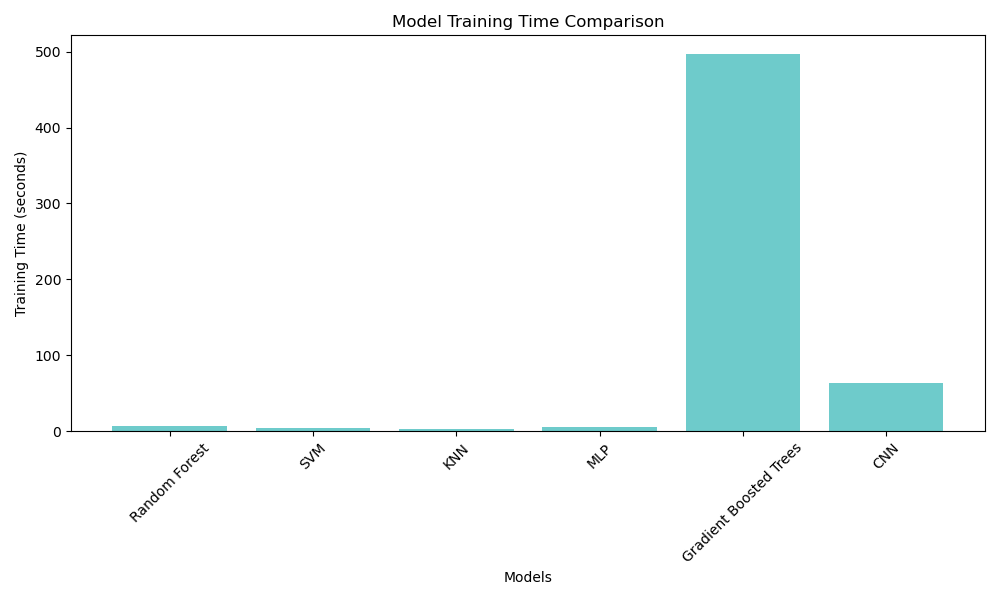
**Confusion Matrix**

****

Overall, you can see that the Random Forest and CNN perform pretty well, and this would be our choice of models and algorithms. Following is the accuracy and the training time comparision.

**Accuracy Comparison**

****

****