Introduction:
This report is a part of my Machine Learning course as a Master's student in Computer Science. The problem statement involves the prediction of accelerometer values of Google Glass using the linear accelerometer readings of two phones. The dataset includes data of 17 participants, where each participant walked for 85-120 seconds, and performed specific activities in the same sequence - opening a door, walking on a flat surface for 83 feet, opening another door, climbing up 2 flights of stairs with 12 steps per flight, climbing down the same 2 flights of stairs, opening another door, and walking on a flat surface for 83 feet. The data collected for each participant includes readings from linear accelerometers of Google Glass, two phones, and the time of the activity performed.

Problem Statement:
The problem at hand is to predict the accelerometer values of Google Glass using the linear accelerometer readings of two phones.

Solution:
The solution to the problem involved collecting the data from the linear accelerometers of two phones and Google Glass. We then processed the data, normalized the timestamp values, and split the data into input and output variables. The input variable was fed to the LSTM model and the output variable was used to train the model.

We then trained the LSTM model using the training dataset, which consisted of data from 17 participants. The model was trained using the sliding window technique, with the window size set to 100 milliseconds and stride set to 50 milliseconds. The number of features was set to 6, with two features for each device - phone, watch, and Google Glass.

The model was then evaluated using the test dataset, which consisted of data from one participant. The LSTM model predicted the accelerometer values of Google Glass using the linear accelerometer readings of two phones

Results:

Although my results were inconclusive, My pred looked like this:
Y_pred [[nan nan nan]
 [nan nan nan]
 [nan nan nan]
 ... [nan nan nan]]

But the objective was to train the model to learn the correlations between the linear accelerometer of google glass and the linear accelerometer of the two phones. And now give only phone data to predict glass data.

But attached is the code in which I tried to train the model using
tensorflow.keras.models - Sequential and tensorflow.keras.layers - Dense, LSTM, Dropout

Steps to execute the code:
conda create --name pythonTensorflowenv python=3.9
conda activate pythonTensorflowenv

pip install -r requirements.txt

Run files:
Mkdir RollingDataFinalTest
Python genarateRollingData.py
Python prediciton.py

Alternative Attempt - 1:
To classify data between walking and climbing - activityclassfication1.py (Code not complete)

Alternative Attemp - 2:
To read the linear accelerometer of glass and phones and generate more samples for all users.

File - genrate6.py (Code not complete)

The problem that I am trying to solve is to generate new data for each of the 17 users based on their linear accelerometer data. The data was collected when the participants performed a series of activities, including opening doors, walking on flat surfaces, climbing stairs, and descending stairs, while wearing Google Glass and carrying two phones. The data from each device's linear accelerometer was collected and combined into one DataFrame for each user.

The solution I have implemented involves concatenating the rolling window data for all users, splitting it into training and testing sets, and defining a model architecture to train on this data. The model architecture consists of a 1D convolutional neural network with two convolutional layers, one max pooling layer, and two dense layers. I want to use the "relu" activation function for the convolutional layers and the first dense layer and "softmax" activation for the output layer. compile the model using the "adam" optimizer, "sparse_categorical_crossentropy" loss function, and "accuracy" metric. Finally, train the model for 10 epochs with a batch size of 32 and validate it on the testing set.

To carry out the solution, we first read in the rolling data files for each user and concatenate the data into one DataFrame for each user. then reshape this data into 1D arrays of length 176 (which is the window size) and stack the data for all users to create a 3D array of shape (number of samples, window size, number of features). then split this data into training and testing sets and define the model architecture. Finally, compile and train the model on the training data and validate it on the testing data.

Here is the github link with all code attempts attempted:
https://github.com/vdhulappanavar/MLSecurityProject