Assignment 3
Blood Pressure Prediction using Graphene Bioimpedance Tattoos

Name - Hugo Collins
Student ID - 203311486

Link to Repo and all my code here. REPO LINK

Abstract

In this report I detail the creation of my machine learning regression model that uses the data taken from continuous cuffless monitoring of arterial blood pressure via graphene bioimpedance tattoos. The project involves the analysis of data from seven subjects across various trials with the objective of predicting the Finapres data using bioimpedance data obtained from the tattoos. I explored a number of different approaches, employing various models and methodologies which are shown below. This project required substantial background research and a comprehensive understanding of the dataset. In this report, I explain why I chose to prioritise continuous data over beat-to-beat data. This decision was based on the belief that there were more interesting avenues to explore within the continuous dataset. I made this choice because it seemed that much of the beat-to-beat analysis and feature extraction had already been thoroughly investigated compared to only using the continuous data in its raw format.

Introduction

Before this assignment I had no knowledge in the world of health technology and in the area of heart rate monitoring and blood pressure. Therefore the first step was to study the paper published by Kireev et al regarding the study and the processes involved. The study looked at the effects of using graphene electronic tattoos to monitor blood pressure. The graphene electronic tattoos (GETs) were strategically positioned over the radial and ulnar arteries on the participants' left forearms, ensuring stable and intimate contact with the skin. This placement allowed precise bioimpedance measurement, leading to accurate blood pressure monitoring. Consequently, participants could undergo continuous blood pressure monitoring during different activities without experiencing discomfort or restriction of movement. In the data structure there were a number of different trials documented along with a number of setups. When I looked into the paper further I noticed how the study employed a meticulous approach to data collection, using a combination of trials to capture a picture of blood pressure. These trials included a HGCP , Valsalva manoeuvre, and cycling exercises, each designed to change the data outcome. Importantly, I noticed in the repo that beat-to-beat monitoring was employed throughout the experiments.

My goal was to develop a model that accurately predicts the finapres data using only features derived from raw Bio-Z and PPG data. I originally planned to format the data into heartbeats. However, after our discussion regarding the challenges posed by the data layout in their repository and the knowledge of the availability of already cleaned beat-to-beat data, I made a deliberate choice not to analyse the data in a beat-to-beat format. Instead, I wanted to explore how close I could get to replicating the beat-to-beat analysis outlined in the paper using only the continuous raw readings. I found this decision presented an intriguing challenge—one that I was genuinely interested in tackling and exploring the outcome.

Understanding the Data:

The first aspect of this assignment was to explore and understand the data. One complex part of the entire process was understanding how the data had been collected and how it was stored. Through reading the paper and the other articles and studying the storage layout, I was able to get an understanding of the data. For each subject there were roughly 10 setups and each setup had a number of trials - subject 7 was slightly different with

only one setup - cycling. Within each trail there were four csv's - one for the BioZ, one for the Finapres, one for PPG and one for PPG finapress. Bioimpedance data (BioZ) was collected using graphene electronic tattoos (GETs) placed on the wrist, measuring impedance changes. These tattoos inject an alternating current (AC) signal into the tissue and record corresponding changes in biopotentials. Finapres data was obtained using a medical-grade BP monitoring device known as Finapres NOVA to monitor blood pressure through a device on the finger. PPG data was recorded, capturing changes in blood volume in the microvascular bed of tissue. I discovered that PPG data can complement Bio-Z measurements by providing additional insights into blood pressure making it a possible feature for my regression model.

Data preprocessing

Due to the data being in a number of different csv's there was a lot of preprocessing required to get the data in the format necessary. The initial phase of my project involved preparing and formatting the data in a Google Colab notebook (here). This step was crucial as it involved merging multiple trials of finapres and BioZ data for all seven subjects. The notebook, accessible here, documents the process of merging these datasets.

Initially, I focused on building a machine learning model using only the raw bioimpedance data to predict the finapres data. This was carried out using data from one subject and one trial. I decided to do this so that I could test a number of different models on one data set and it would be a more computationally efficient way of assessing different model performances and hyperparameter tuning. In order to do this (as seen in the notebook) I concatenated the BioZ files with the Finapres files for each subject. As both had different timestamps, more data preprocessing was required. In order to sync up and align the timestamps I needed to interpolate the data. I created a function _ interpolate_and_shift() which I used in all my models to preprocess the data. The function involved filling in missing data points and standardising the time axis across all sensor readings. By creating a common time array (common_time) with evenly spaced intervals, I ensured uniformity across my dataset. The other preprocessing method used was that shifting was applied to adjust for any temporal misalignments between the readings. The idea behind this was to allow for more meaningful comparisons and analysis.

The final aspect of data preprocessing involved subject 7 who had no BioZ4 data. In order to combat this I attempted to use a mean value to fill in however I was unsure about this and the effect it was having on my results so I decided to stick with the official raw accurate data only. As a result, subject 7 was left out of the final model and any subsequent models after Bioz4 was incorporated as a feature. Throughout the process I decided to base my model on the baseline setup only. The decision to use data from the baseline setup exclusively was deliberate and driven by practical considerations. Given the time constraints, I chose to concentrate on developing an effective model for a single setup rather than incorporating data from multiple setups..

Machine Learning Model Development

Implement the machine learning pipeline described in the original study.

The first step I took in this project was running the code from the original study's gitlab and analysing the results in visual studio. In order to run this code I had to make a number of adjustments to the code, one of which involved concatenating the dataframe instead of appending. I also had to change a number of hyperparameters in the sklearn.ensemble.AdaBoostRegressor due to the version being outdated. This involved changing base_estimator_ to estimator_ and a few minor details like this. I then began analysing the output etc but following the class where we discussed the issues and problems surrounding the dataset I decided to focus on my efforts on the BioZ data and using that to predict the finapres data, rather than using a beat to beat approach and using the already cleaned data.

Experiment with alternative machine learning models and evaluate their performance.

As mentioned above I decided to build the original base of my model based on one subject in order to improve efficiency and make the most of my limited time. The initial stages involved exploring a number of different models - over the course of this stage I examined 4 key models - a BiLSTM model, a random forest model, an AdaBoost model (as used by study) and an XGBoost model. Each model offered distinct advantages and catered to specific characteristics of the data. I did some research into previous blood pressure prediction studies to get an estimate of what models could be useful. The thought process behind selecting these 4 models are listed below:

Random Forest: Firstly, the random forest model was chosen due to its robustness against overfitting and its ability to handle a large number of features. It served as an initial baseline model, providing insights into feature importance and overall predictive performance.

AdaBoost: The AdaBoost model was selected based on the fact that it had been used in the initial study and I wanted to explore how it would perform in contrast to these other models.

BiLSTM: During my research, I chose to try the BiLSTM method due to its ability to capture sequential dependencies in time-series data. I learned that its bidirectional architecture allows it to learn from both past and future contexts, making it well-suited for capturing both temporal and long-range dependencies present in signals like blood pressure and bioimpedance data.

XGBoost: Another model I wanted to experiment with was XGBoost. During my research I came across a paper where Attivissimo et al (link) found that "*XGBoost models are more accurate than NN models in both systolic and diastolic blood pressure measurements"*. An added bonus to this was that I was familiar with this ensemble method from the first assignment and thought it would be interesting to carry over my learnings from A1 into this assignment. XGBoost's ability to handle large datasets, capture complex interactions among features, and deliver state-of-the-art performance also made it an obvious choice to experiment with.

In summary, the selection of these models was not random. It was driven by their strengths and suitability for handling the complex, sequential, and nature of the data. Each model contributed uniquely to the iterative process of model development. In the end I decided to use XGBoost as it performed best during initial trials.

Feature Extraction

(Note - These Results Below were run using the optimised XGBoost Model using cross validation and optimised Parameters as detailed above)

Original Studies Use of Features

One area that required a lot of research was feature extraction. In order to gain some insight into the data I began looking into the features extracted by the original study. They carried out a beat to beat machine learning regression model and therefore used a number of different features that I wouldn't be using such as Inter-beat intervals and Pulse Transit Time. The data contained in the original study's repo was poorly labelled and not explained but I did some further research to try to understand the meaning of each column and measurement. The column names containing "IBI" I believe represent the inter beat interval. From research I discovered that this metric is valuable in understanding the rhythm and regularity of the pulse. For the PTT columns it is safe to assume that these refer to features using the pulse transit time - which refers to the time it takes for a pulse to travel between two locations in the body. Finally, moving on to the features P_D2M__TR, P_D2S__TR, and P_D2I__TR, these variables likely refer to different measurements associated with the "P" signal. In the paper it is described how this is crucial for understanding responses over time These could include parameters such as rise time, settling time, and interval duration. From reading the paper it is clear that a lot of these features were extracted from the beat to beat data, with up to 50 features per heartbeat. While it was a slow meticulous process

understanding these features and the data it was a very beneficial one and one I enjoyed. Below, I describe how I tried to follow this path of feature extraction using only the continuous measurements.

Bio-Z Measurements

As an initial step for my process I used only the Bio-Z measurements using the BioZ 1-4 measurements only. In order for my code to work I had to do a bit of data preprocessing on subject 7. As subject 7 had no Bio-Z 4 data present I experimented with different approaches to handle the missing data. Initially, I attempted imputing missing values using interpolation or median values, but when I created my final model, considering the potential impact on the analysis, I ultimately chose to exclude Subject 7 from the dataset. This allowed me to maintain data integrity and ensure consistency across the dataset. For the initial versions of my model I experimented using only Bio-z 1- 3 but when I added in Bio-Z 4 I found the median values to be less accurate and wanted to maintain accuracy of data. Subject 7 was used during testing for the majority of my earlier model versions however.

Optimising Shift of Finapres Data:

The first attempt I made to improve features and data preparation was to try to optimise the shift of the Finapres signal so that its peaks line up better with those of the bioimpedance readings. I created a for loop that shifted from -500ms to 500ms in 100ms increments and returned a RMSE score for each shift. The final output was the best shift and this was judged by the shift that yielded the lowest RMSE. The shift was 0.19999999999999984s and this is the value I used for the shifting of the Fianpress signals.

Results of Optimised Model using only Bio-Z Measurements and Optimised Shift:

```
Best Parameters: {'subsample': 0.8, 'reg_lambda': 3, 'n_estimators': 400, 'max_depth': 3, 'learning_rate': 0.01, 'gamma': 0.7, 'colsample_bytree': 1.0}
Best Score: 0.7548071992946804
Optimized Model Evaluation:
Mean Absolute Error (MAE): 5.46731716820409
Root Mean Squared Error (RMSE): 6.691067654949225
R^2 Score: 0.7966410962950174
```

Adding Derivatives of Bio-Z data:

The next part of feature extraction that I undertook was extracting the derivatives of the Bio-Z features. I felt that some of the short term changes were not being fully captured by the raw signals alone so I decided to add the first derivatives of the Bio-Z data. By doing this the model gains additional information about the rate of change in the bioimpedance signals over time. The derivatives look into the slope and gradients of the signals which gives the model clues into the speed of change in the bioimpedance data. This led to a significant increase in R^2 and improved all metrics. This was definitely a valuable feature addition.

```
Best Parameters: {'subsample': 0.6, 'reg_lambda': 3, 'n_estimators': 300, 'max_depth': 6, 'learning_rate': 0.05, 'gamma': 0.5, 'colsample_bytree': 1.0}
Best Score: 0.5442116516960422
Optimized Model Evaluation:
Mean Absolute Error (MAE): 3.9423231695683807
Root Mean Squared Error (RMSE): 5.086608813295501
R^2 Score: 0.8697252912469511
```

Moving Averages

While I attempted to implement moving averages, the time constraints caught up with me and I was forced to leave it out. However, during our time series module in CA4024 I learned about the importance of moving averages and how they can smooth out data fluctuations, revealing underlying trends. I was really disappointed to have to leave this out as integrating them could have enhanced the model's ability to capture subtle variations and potentially led to more accurate predictions.

Checking Shift again

```
Best Shift: 0.19999999999999984s, Best RMSE: 7.712763483175924
```

Updating shift - Still appears to be roughly the same measurement so I kept this

Adding Bioz - 4 Data - Removing Subject 7 and Experimenting More with changing shift

The next step involved inputting BioZ-4 data. As mentioned above I attempted using median values for subject 7 in order to include them in the study but found that
Results:

```
Best Parameters: {'subsample': 0.8, 'reg_lambda': 2, 'n_estimators': 500, 'max_depth': 6, 'learning_rate': 0.05, 'gamma': 0.5, 'colsample_bytree': 1.0}
Best Score: 0.5842631076098793
Optimized Model Evaluation:
Mean Absolute Error (MAE): 3.4306921869588676
Root Mean Squared Error (RMSE): 4.518345681493657
R^2 Score: 0.8972072842198744
```

Adding PPG

The next step in feature extraction I tried was actually adding a separate feature - the PPG readings. This decision was motivated by valuable insights that PPG signals can offer and the fact that there was PPG data available, it would have been silly to ignore.To integrate PPG data, I revisited my DataPrep notebook and merged the PPG readings with the existing BioZ and Finapres data for all baseline trials across subjects. I then had to adjust the interpolation function to also account for the PPG readings and then I was able to use these to test if it improved results. I observed improvements across all evaluation metrics. Encouraged by these enhancements, I retained the PPG readings as a feature in my model.

Results:

```
Best Parameters: {'subsample': 0.6, 'reg_lambda': 1.5, 'n_estimators': 500, 'max_depth': 6, 'learning_rate': 0.05, 'gamma': 0.1, 'colsample_bytree': 1.0}
Best Score: 0.6887468282310161
Optimized Model Evaluation:
Mean Absolute Error (MAE): 3.2888637740178894
Root Mean Squared Error (RMSE): 4.2833225688534675
R^2 Score: 0.9076227573077625
```

Adding PPG Derivatives and extracting second derivatives for both BioZ and PPG data

PPG Derivatives:

The first derivative PPG is also known as velocity plethysmogram (VPG) and this feature can help capture changes in blood volume over time. The first derivative of the PPG signal can provide insights into the slope or rate of change of blood volumes similar to the BioZ derivatives.Therefore, I decided to experiment with incorporating derivatives of the PPG signal into the feature set to further enhance the predictive capability of the model.

Second Derivatives:

The second derivative, known as the acceleration plethysmogram (APG), provides a refined analysis of the PPG waveform. By using the APG as a feature, this stabilised the baseline and enhanced the separation of waveform components more effectively than the first derivative. This allowed me to capture more detailed features of the PPG and Bioz signals. This approach allowed me to further remove noise from my data and I hoped it would potentially lead to improved accuracy in predicting blood pressure variations.

Results:

```
Best Parameters: {'subsample': 0.6, 'reg_lambda': 1.5, 'n_estimators': 500, 'max_depth': 6, 'learning_rate': 0.05, 'gamma': 0.1, 'colsample_bytree': 1.0}
Best Score: 0.8206250434060507
Optimized Model Evaluation:
Mean Absolute Error (MAE): 2.570482048742321
Root Mean Squared Error (RMSE): 3.3438082872541903
R^2 Score: 0.9437028808390097
```

Adding Average Bio-z and Cross Correlation between Bio-Z

Adding the average Bio-Z values involves calculating the mean of the Bio-Z measurements across all available channels for each time point. This feature offers a broader perspective on the physiological dynamics captured by the multiple channels, helping to identify overarching trends and patterns in the data.

Computing the cross-correlation between Bio-Z signals assesses the degree of similarity or correlation between pairs of Bio-Z channels over time. This measure indicates how closely the signals vary together, revealing relationships and synchronizations in physiological activity across multiple measurement sites. By examining these interrelationships, we gain insights into the coherence and coordination of physiological processes, potentially uncovering interactions between different regions of the cardiovascular system. Incorporating these features expands the scope of our analysis, considering both the collective behaviour of Bio-Z signals and the interactions between them, thereby enhancing the predictive capabilities of our model.

The Final Model:

Data pre-processing:

- The data is loaded for each subject and trial from CSV files.
- The data is passed through the following interpolation function to align the varying timestamps onto a common time axis.

```
# Define the interpolation and shifting function with subject ID
def interpolate_and_shift(data, shift_seconds, subject_id):
    start_time = max(min(data['time']), min(data['time.1']), min(data['time.2']))
    end_time = min(max(data['time']), max(data['time.1']), max(data['time.2']))
    common_time = np.linspace(start_time, end_time, num=1000)

    finapres_interpolator = interp1d(data['time'], data['FinapresBP'], bounds_error=False, fill_value="extrapolate")
    bioz1_interpolator = interp1d(data['time.1'], data['BioZ1'], bounds_error=False, fill_value="extrapolate")
    bioz2_interpolator = interp1d(data['time.1'], data['BioZ2'], bounds_error=False, fill_value="extrapolate")
    bioz3_interpolator = interp1d(data['time.1'], data['BioZ3'], bounds_error=False, fill_value="extrapolate")
    bioz4_interpolator = interp1d(data['time.1'], data['BioZ4'], bounds_error=False, fill_value="extrapolate")
    ppg_interpolator = interp1d(data['time.2'], data['PPG'], bounds_error=False, fill_value="extrapolate")

    finapres_common = finapres_interpolator(common_time)
    bioz1_common = bioz1_interpolator(common_time)
    bioz2_common = bioz2_interpolator(common_time)
    bioz3_common = bioz3_interpolator(common_time)
    bioz4_common = bioz4_interpolator(common_time)
    ppg_common = ppg_interpolator(common_time)

    time_difference = np.diff(common_time)
    average_delta_time = np.mean(time_difference)
    shift_indices = int(shift_seconds / average_delta_time)
    finapres_shifted = np.roll(finapres_common, shift_indices)

    return common_time, finapres_shifted, bioz1_common, bioz2_common, bioz3_common, bioz4_common, ppg_common, np.full_like(common_time, subject_id)
```

Features -

This model highlights the evolution of my feature extraction. Over the entire process I built up my list of features starting from the standard Bio-Z features and building up to a total of 18 separate features which are detailed above in the feature extraction section. See below the list of features:

```
features_train = np.vstack((all_data['bioz1_common'], all_data['bioz2_common'], all_data['bioz3_common'], all_data['bioz4_common'],
                            all_data['bioz1_derivative'], all_data['bioz2_derivative'], all_data['bioz3_derivative'],
                            all_data['bioz4_derivative'], all_data['bioz1_derivative_2'], all_data['bioz2_derivative_2'],
                            all_data['bioz3_derivative_2'], all_data['bioz4_derivative_2'], all_data['ppg_common'],
                            all_data['ppg_derivative_1'], all_data['ppg_derivative_2'], all_data['trial_number'],
                            all_data['avg_bioz'], all_data['cross_corr_bioz_ppg'])).T
target_train = all_data['finapres_shifted']
```

Model is trained on all the data from the subjects excluding subject 4 who is used for testing only. This allows me to test my model on an unseen subject and evaluate results. Features are scaled using MinMaxScaler to ensure normalisation and further ensure that all features are on a common scale and have similar ranges. Then hyperparameters for the XGBoost regressor are optimised using RandomizedSearchCV with cross-validation of 15 folds. The model is evaluated using the metrics Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) score. Performance is then tested specifically for a test subject (Subject 4) by comparing actual and predicted FinapresBP values. The results of my final model are seen below.

Results:

```
Best Parameters: {'subsample': 0.6, 'reg_lambda': 1.5, 'n_estimators': 500, 'max_depth': 6, 'learning_rate': 0.05, 'gamma': 0.1, 'colsample_bytree': 1.0}
Best Score: 0.8192653934802723
Optimized Model Evaluation:
Mean Absolute Error (MAE): 1.734461925389452
Root Mean Squared Error (RMSE): 2.397215279616611
R^2 Score: 0.9762365841660995
```

The model's performance saw a remarkable improvement from its earlier stages, with the Mean Absolute Error (MAE) dropping from 9 to 1.73 and the R-squared (R^2) score climbing from 0.4 to 0.976. This progress reflects a careful and iterative tweaking process, resulting in a highly satisfying outcome. With an MAE of 1.74 and RMSE of 2.39, the model demonstrates its ability to make accurate predictions with a small margin of error. The high R^2 score indicates a strong correlation between predicted and actual values, reaffirming the model's effectiveness in making precise predictions. In order to further improve this model without looking at beat to beat I would've liked to look further into the moving averages feature. However, overall I am pleased with the results of my model. The most beneficial aspect of this process was having to spend significant amounts of time understanding the data and watching it improve slowly over a number of iterations was a great learning experience. Below are some visualisations of my results:

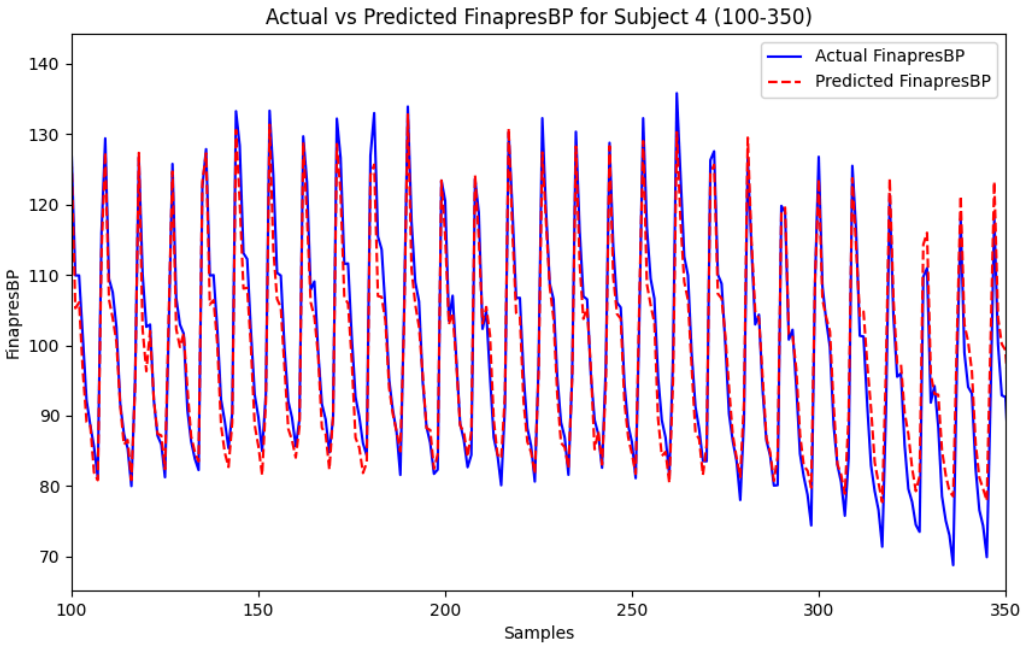

Fig 1: Actual Vs Predicted between 100 - 250

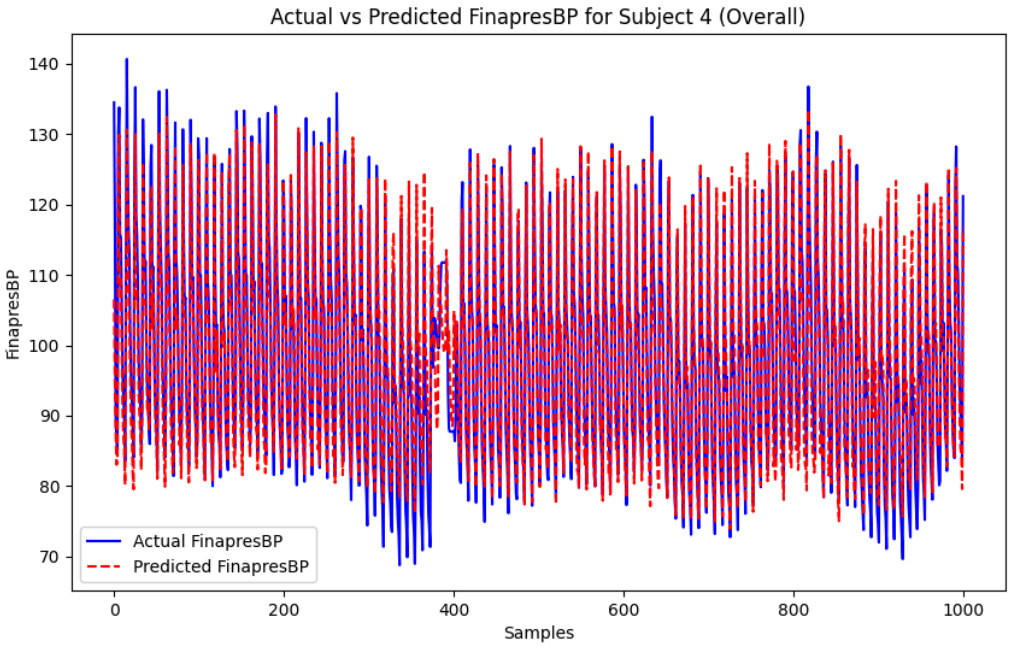

Fig 2 Overall Actual vs Predicted for Subject 4

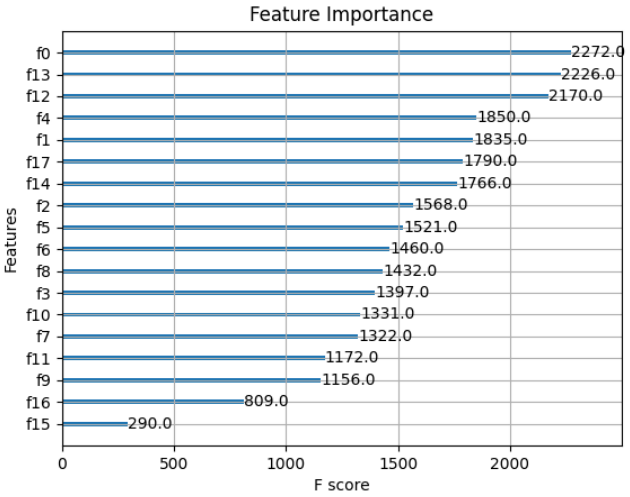

Fig 3 - Feature Importance

Here we can see the importance of each feature - Interestingly two of the 3 top features refer to the PPG derivative values which is very interesting. Also BioZ 1 seems to be the main feature. However, we can see that most features are contributing a significant amount to my model.

Comparison of Results with Original

Comparing the results of both models was quite a difficult task as I took a slightly different approach to the original study by predicting Finapres data and not breaking the data into beat to beat format and using that to predict SBP and DBP. My thought process behind avoiding using the already formatted beat to beat data was that I personally wanted to see how I could improve a basic Finapres prediction model through an iterative approach. This was the approach that appealed to me the most and the one I decided to follow. Theres no denying that the original model created in the study obtained impressive accuracies, with DBP and SBP

estimation errors of 0.2±4.5mmHg and 0.2±5.8mmHg, respectively. This categorised it as comparable to Grade A accuracy which was very impressive. However, my model showcases progress, with a Mean Absolute Error (MAE) dropping from 9 to 1.73 and the R-squared (R^2) score climbing from 0.4 to 0.976. This MAE of 1.74 and RMSE of 2.39 further reinforce the model's ability to make accurate predictions with a small margin of error also. After running the code provided by the study as mentioned above on the data stored in the repo I was able to gain access to the testing results of the study's model on Subject 4 as seen below:

Original Model Testing Results on Subject 4

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		training_n	feature_n	model nan	Subject ID	testing DBP N	CC	ME	RMSE	testing SBI	CC	ME	RMSE
2	3	SingleTrair	WithIBI	ada	4	347	0.829289	6.184595	8.917418	347	0.899135	10.52235	12.19686

My XGBoost Model Results on Subject 4

```
Best Parameters: {'subsample': 0.6, 'reg_lambda': 1.5, 'n_estimators': 500, 'max_depth': 6, 'learning_rate': 0.05, 'gamma': 0.1, 'colsample_bytree': 1.0}
Best Score: 0.8192653934802723
Optimized Model Evaluation:
Mean Absolute Error (MAE): 1.734461925389452
Root Mean Squared Error (RMSE): 2.397215279616611
R^2 Score: 0.9762365841660995
```

Here two separate models were tested each with different window lengths of 10 and 20. It is clear that my model has a far superior performance on RMSE for subject 4 with an RMSE of 2.39 in comparison to 8.917 and 12.196 as seen above. This demonstrates my model's predictions are closer to the actual values, indicating higher accuracy and highlights the effectiveness of my model in generating predictions that closely align with the actual values. Overall, I was very pleased with these results and while I note it is not comparing completely like with like, given the short time frame I am very pleased with the results.

Conclusion

Overall this was a very interesting assignment. This assignment differed from previous assignments given that there was a lot of background research and reading required which I thought was very beneficial and more realistic. While parts of this assignment were quite stressful and there was a tight enough deadline following the meeting where we discussed progress a week before the deadline, I felt this project was incredibly beneficial. It gave me hands-on experience studying another papers' work and trying to take my own approach to their research and apply my data science learnings to an area where I had no previous experience or knowledge. Overall I was pleased with my work and results. I took an iterative approach and tested a number of different models and methods. I experimented with different approaches varying from random forests to ADABoost. Ultimately I ended up going with XGBoost using k-fold cross validation, paired with some interpolation function and lots and lots of experimenting with feature extraction. In conclusion, I was pleased with the processes I followed and was disciplined with my approach. There are a few further ideas I would like to follow further but these can maybe be used for future research perhaps. It was a very interesting but challenging project, one that required a lot of code (I used 6 separate notebooks) but overall I am pleased with my results.