Goal

Predict "too tired to work" events from PPG data continuously monitored throughout their shifts and prevent fatigue-related accidents.

Backgrounds:

Workers' cognitive fatigue is the cause of 2/3 of accidents in the mining industry and there are similar dangers in road and air transportation and any shift-work environments where people are working 12-hour shifts (eg. hospitals).

The gold-standard for neuroscientists studying fatigue is an electroencephalogram (EEG) but it is hard to get good EEG measurements outside of a clinical setting. There has recently been more research using electrocardiogram (ECG or EKG) measurements, particularly using Heart Rate Variability (HRV) as a factor. Good ECG measurements are also not practical on people who are moving around in their working environment.
Photoplethysmograph (PPG) measurement on the ear is another technique to measure HRV and this can be packaged in a way that is accurate, comfortable and unobtrusive for people to wear all the way through their shift and on their drive home afterwards.

5-Gamer trial data

There were 5 participants, each attempting a 22 hour "shift" of computer gaming. For each participant there is:

- A PPG time-series (level of 660nm red light transmitted through participant's ear-lobe) sampled at approx 100Hz
- A diary of annotations including:
    - 'sleep-2-peak' reaction time each hour
    - caffeine and food ingress and egress
    - self-assessment Stanford sleepiness scale (1-7) each hour

Analysis

Factors to derive could include:

- every heartbeat (traditionally ECG "R-peak" but fastest-changing edge of the PPG curve may be a more accurate proxy)
- heart rate (HR) and heart rate variability (HRV)
- respiratory rate (RR)

All in order to estimate:

- Time to "too tired to work" epoch. This is the point at which you will lose the battle against falling asleep.

Consider:

- It's a "mean time to failure" problem

- "Cognitive fatigue" is *not* the same as the sleepiness you feel every day at bedtime
- Cleaning and annotating the data
- Dealing with gaps in the data (eg earpiece taken off)
- Dealing with noise in the data
- Factors to control or monitor better with subjects during future trials
- Possible sources of more 3rd-party fatigue data, for example Sleep Centres

Approaches

For each gamer:

1. Clean the PPG .CSV data files -> new raw PPG time-series file(s) (A)
2. Clean/normalise the annotations in the .CSV fatigue diary? -> new annotations file (B)
3. Heartbeat peak detection to obtain nanosecond timestamps for every heartbeat in the PPG data (A) -> new heartbeat timestamps file (C)
4. Calculate heart rate and heart rate variability from the heartbeat timestamps (C) -> new time-series file (D)
5. Explore trends and anomalies in the HR & HRV data (D). -> append new annotations to gamer's annotation file (B)
6. Document all we learn about baselines, trends, anomalies and correlations with the annotations and any ideas for further factors that move us towards a mean-time-to-failure prediction of their "too tired to work" epoch.
7. -> new time-series file of experimental attempts of time-to-"too tired to work"-epoch (F)
8. Breath (respiration) detection to obtain nanosecond timestamps for every breath from the PPG data (A) and/or the heartbeat timestamps (C) -> new breath timestamps file (E)
9. Calculate respiration rate from (E) -> append Respiratory Rate column to (D)
10. Revisit 5-7 with extra respiration factor
11. Revisit 5-7 for correlations in (D) with the new micro-sleep annotations

Classification algorithms

I evaluate multiple machine learning models (Decision Tree, Random Forest, KNN, Gaussian Naive Bayes, Multinomial Naive Bayes) on a dataset.

1. It initializes an empty DataFrame `accuracy_df` to store the results of the evaluation.

2. It iterates over each model, makes predictions on the test set, calculates various performance metrics (accuracy, precision, recall, F1-score) using the `sklearn` library, and prints the confusion matrix for each model.

3. For each model, it appends a row to the `accuracy_df` DataFrame with the model name and the calculated performance metrics.

4. Finally, it prints the `accuracy_df` DataFrame summarizing the performance of each model.

Neural Network

I define and train two neural network models using the Keras framework in Python.

1. The first model (`classifier`) is a simple feedforward neural network with three layers. It has an input layer with 10 units, a hidden layer with 6 units, and an output layer with 1 unit using a sigmoid activation function. The model is compiled using the Adam optimizer and binary cross-entropy loss function. It is then trained on the training data (`x_train` and `y_train`) with a batch size of 15, validation split of 0.2, and for 10 epochs. Finally, the model's accuracy is evaluated on the test data (`x_test` and `y_test`).

2. The second model (`classifier2`) is a more complex neural network with five layers. It has an input layer with 12 units and a hidden layer with 9 units using a tanh activation function, followed by three hidden layers with 6, 3, and 1 unit(s) respectively, using sigmoid activation functions. Like the first model, it is compiled using the Adam optimizer and binary cross-entropy loss function. It is trained on the training data with a batch size of 7 for 10 epochs. Finally, the model's accuracy is evaluated on the test data.

Bibliography

- Gang Li & Wan-Young Chung: Detection of Driver Drowsiness Using Wavelet Analysis of Heart Rate Variability and a Support Vector Machine Classifier

which refers to this seminal 1996 article on LF/HF HRV analysis:

- Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology: Heart rate variability: Standards of measurement, physiological interpretation and clinical use

Acknowledgements

This dataset is part of the Predicting Cognitive Fatigue with Photoplethysmography (PPG) project.