ML4QS: Assignment 3

Martin Banchero and Dennis Dekker

VU, Amsterdam, NL

1 Introduction

In the last decade wearable devices became more popular and with this a lot of sensory data became available. Different application are being applied, from quantified energy resources in a house to quantified personal activities. This quantified self measurements lead to broad spectrum of problems that can be approached with machine learning algorithms. These algorithms are able to extract good predictions that can be useful to help the user of this wearable devices to improve the desire tasks.

Recently, different sensory databases are made publicly available. One of these datasets is the Wearable Stress and Affect Detection (WESAD) Data Set, which contains measurements of blood volume pulse (64 Hz), electrodermal activity (4 Hz), body temperature (4 Hz), heart rate sensor (1 Hz) and a three-axis accelerometer (32 Hz). The each sensor measured data with a different sampling rate, indicated in parentheses.

As a research question in the present project is proposed to predict heart rate using sensory data from the WESAD dataset [1], which contains 6 features next to the heart rate. Although this dataset contains the data of 15 different persons, only data belonging to one person is used, which is mainly due to the size of the complete dataset which is around 15 GB. Using one person the size is reduced to 1 GB making it more easy to handle. Next to this, including data from all persons would have led to combining all the data or using an algorithm that can cope with multiple time series, which is outside the scope of the course.

2 Methods

This section discusses the main preprocessing methods implemented and the applied temporal learning algorithms used.

2.1 Pre-process raw data

The data consisted of multiple single files, each containing measurement of one sensor. In order to make the data easier to handle, the datasets were combined into one single dataset. We used the *Pandas* package for python 3.7. As the sensory data were measured at different sampling rates, we used the highest sampling rate as timescale (64 Hz for blood volume pulse). For the other features we introduced NaNs on rows without measurements (due to lower sampling rate).

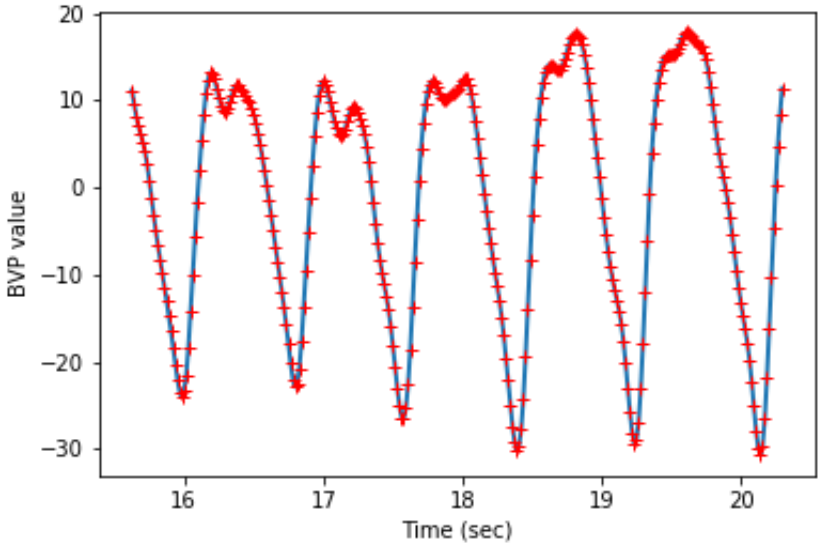
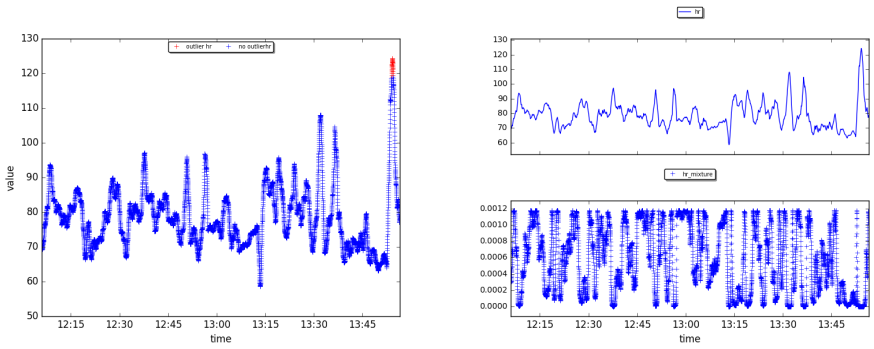


Fig. 1. Subsection of blood volume pulse (BVP). The data shows a high information density. The periodicity of the pulses lay around 800 ms.

The next step was to determine the granularity to be applied on the data. The heart rate attribute has an sampling rate of 1 Hz, so in order to used these values to score the prediction the granularity should not be greater than this. Also, all other measurements have a smaller sampling rate, so a large granularity would lead to a lot of information loss. To set a value for the granularity, we looked into the information within each feature. For the blood volume pulse (BVP), the information density was the highest. A subsection of the BVP data is displayed in Figure 1.

The figure shows that the periodicity of the data is around at least 800 ms. To reduce the amount of data, but still retail this cycle, we propose a granularity of 100 ms, which will result in 8-9 datapoints for each cycle, which should be enough to retail most of the information. For the other features the data showed that the periodicity of possible cycle was not smaller that the BVP data (data not shown).

Outlier detection method Different methods were applied to detect outliers. Several methods exist to deal with task, of which we explored 4: Chauvenet's criterion, Mixture models, simple distance base and local outlier factor (lof). We can categorize these as two distribution based methods, Chauvenent's and Mixture model, and two distance based methods, simple distance based and lof.



(a) Chauvenet's (b) Mixture model

Fig. 2. Results of outlier detection methods, (a) Chauvenent's and (b) Mixture model, on heart rate data. Outliers are indicated with red. For Chauvenet's method the increase in heart rate at the end was marked as outliers.

Here we show the results for heart rate for Chauvenet's criterion and Mixture model, see figures2a and 2b respectively. For Chauvenet's, a peak of the heart rate is marked as a outlier, while it is clearly not. The increase in heart rate is not a single point, which could be the result of a measurement error, and is also not an extreme value for heart rate. The mixture model shows a much better outlier detection, without removing the peak. As for the distance based methods (data not shown), the simple distance based methods also removed some heart rate datapoints. Local outlier factor performed similar to mixture model. Overall, mixture model and lof performed best, and mixture was applied to the data as lof required a lot more computational time.

Missing values handling In order to determine which method to use for handling missing values, we applied interpolation, Kalman filter, mean and median to impute the missing values. From these methods, interpolation was applied since the mean and median would not even come close to reasonable values and we observed the same result in comparison to the Kalman filter. The latter tries to detect ouliers and impute missing values. As we already did outlier detection and the interpolation of the missing values gave a good result, interpolation was applied to the dataset.

2.2 Data transformation

In order to filter out noise of the data, we applied the lowpass Butterworth filter. This filter remove periodic irrelevant data, which would result in a better regression model, as the model can not train on irrelevant data. The filter was applied on the accelerometer data, the heart rate, electrodermal activity and the temperature. The cutoff was set to 1.5 Hz, as higher frequencies would not

contribute to important information in these features. The blood volume pulse was not transformed, as the frequencies that are higher than 1.5 Hz could contain information about the heart rate The frequencies of the pulse will be around 1.5 Hz, as the periodicity of the data was around 800 ms, see section 2.1 ($f = \frac{1}{800/1000} \approx 1.3$ Hz).

While transformation can be applied to single features, information from multiple or all features combined can also result in information gain and useful features. A method that can combine features and extract the highest variance in the data is Principal Component Analysis (PCA). The advantage of PCA is that we can describe the data using only the a few features, while retaining the most important aspects of it (based on variance). However, the abstraction of the data will result in features that can not be interpreted easily as the features are expressed in highly dimensional space.

In order to express the variation in the data, PCA outputs multiple components, each with a lower amount of variation explained. Using all these component would result in no information loss, as all data is still expressed only in a different format. However, using less component lead to the extraction of data with the most information, or variance, while removing component that are less important, what could be the noise. The amount of components to be used can be determined by plotting the amount of variance explained against the index of the corresponding component, and then selecting the number of components where a 'break' in the graph can be seen. While there are more sophisticated methods to determine the amount of components, this rule of thumb will give us a relatively robust and simple way. The PCA resulted in the selection of 5 components (data not shown).

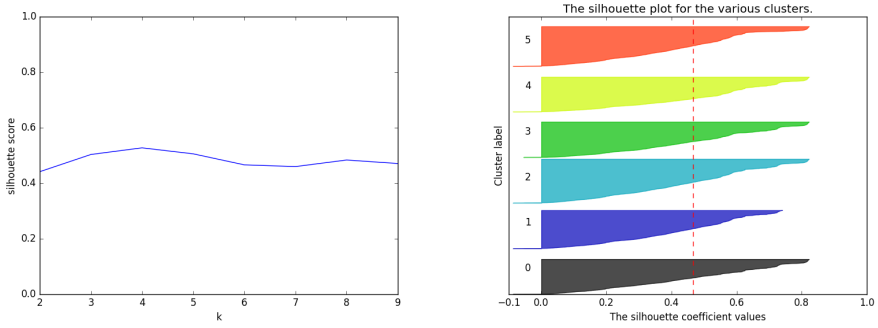
2.3 Feature engineering

As we are dealing with a time series, the information in the data can be correlated with earlier events in the data. As simple machine learning methods can not take advantage of these relation, so we should apply a method that can extract this information and present it to the algorithms. By using a sliding window over the data, we can extract the numerical relations and frequencies within these windows using the mean and standard deviation for numerical relations and Fourier transformation for frequencies. For each of the values, we can take different sizes of sliding windows, and calculate the import numerical values and frequencies within these domains. Each size will give different results, as small windows will give a result in more noise retention but will not loose too much information, while a large window retains removes the noise but will not have a lot of variation. In order to find a balance, different window sizes (5, 30 and 600 seconds) were applied and added as features to the data.

2.4 Clustering

While the dataset does not contain labels for certain activities or is based of measurement for different activities, we can use clustering to find structure in

the dataset and give the models a push in the right direction. Clustering can be done using different distance measures, but as we are only using numerical data a simple Euclidean distance measure can be applied. For the actual clustering, three different methods were applied: k-Means, k-Medoids and Agglomerative clustering. For each of the methods, a range (2 to 10) clusters were applied. The performance of the clustering methods was determined using the silhouette score. The best performance was achieved using k-Means with 4 clusters, see figures 3a and 3b.



(a) Silhouette score (b) Silhouette plots

Fig. 3. Plots showing the performance of k-Means clustering. (a) The silhouette score per k. For the final clustering k = 4 was chosen. (b) Silhouette plot per amount of clusters

2.5 Learning regression algorithms

In order to obtain the heart rate prediction, different temporal learning algorithms were used, such as Echo State Networks(ESN), rNN and Time series. This algorithms are able to incorporate time making it more robust for the prediction of the type of variable that we want to predict, heart rate. For ESN and rNN, different features sets were applied for training the models. For Time series, only the basic features were applied as not enough computational power was available. Each model was training using approximately 75% of the data, the other 25% was used to test the model. The error measure used was Mean Squared Error. The models were compared using this error function for each feature set and the final prediction graphs.

3 Results and Discussion

In order to get and idea of the accuracy of the temporal learning algorithms, the mean squared error (MSE) is plotted in figure 4. The highest MSE is given by

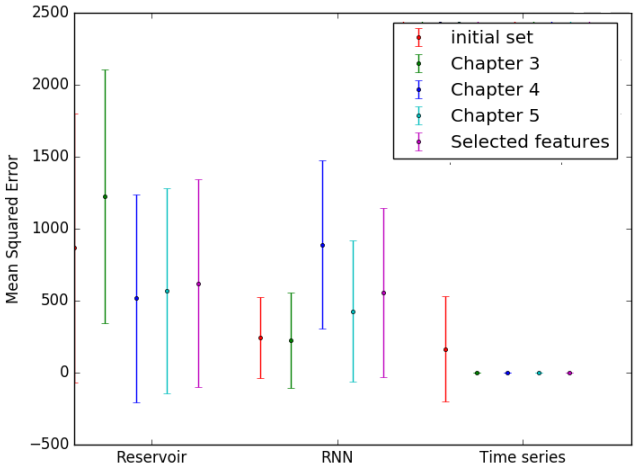
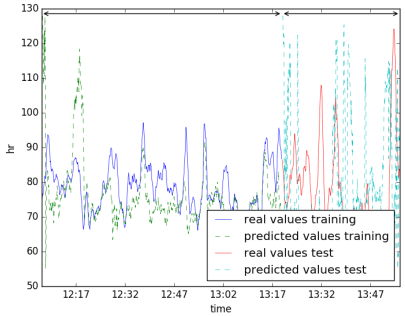


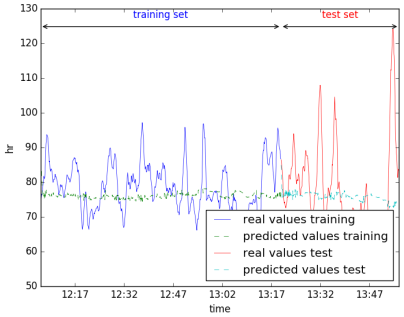
Fig. 4. Mean squared error for the three different algorithms that include time.

Reservoir for the features of Chapter 3 (PCA), similar to previous results, and here shown in figure 5a, this algorithm perform relatively good for the train set but not so well in the test set, losing generalizability. The time series presents the lowest MSE, but this is because it predicts the average, see figure 5b. Therefore, the MSE will not raise as much as the other algorithms, which are much more 'risc taking'. This characteristic of the Time series makes it, while very robust, undesired as a predictor as the computational cost could be immensely reduced by only taking the average of the heart rate data. Compared to the other two, rNN performs best in terms of prediction and accuracy. The MSE is low for the more basic feature sets, and the final plot shows a relatively good resemblance of the test set. However, the plot also shows that the output of the algorithm is very noise, completely different from the Time series. The performance of the rNN could maybe be increased by giving the algorithm more time to train, but this might also apply to the other algorithms.

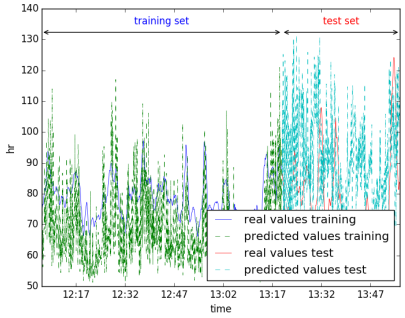
Regarding the overall result, the dataset seems difficult to predict. The heart rate data has some 'unpredictable' peaks, which might be difficult to pick up from the raw data. Similar to other studies, more data would increase the performance of the algorithm. While this could not be done for this dataset, as it was not ours, a future study could make use of the data that is available of other persons within the same study. This would increase the amount of data by 15-fold, as there is data available of 15 persons. However, this would require sophisticated preprocessing methods to combine these dataset to be used for the learning algorithms, or different learning algorithms that can learn on multiple multivariate time series data.



(a) Reservoir



(b) Time Series



(c) rNN

Fig. 5. Plots showing the performance of the different algorithms, Time Series, rNN and echo state.

4 Conclusion

This study assessed the usage of different temporal learning algorithms to predict the attribute heart rate. We found that all the methods present trade-offs, so in this case is not possible to ensure that rNN has a better performance than the rest. In this way the current study is inconclusive as to which of these is the best algorithm to predict heart rate using sensory data.

References

[1] Philip Schmidt et al. "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection". In: (). DOI: 10.1145/3242969.3242985. URL: https://doi.org/10.1145/3242969.3242985.