Universität Augsburg
Fakultät für Angewandte
Informatik

Hypertension Detection

Feature Extraction and Selection for XBG Model

Victor Hug

16.09.2025

DSENS
Professur für
Diagnostische
Sensorik

Methodology

Feature Selection

200+ Feature Options: see feature_comp.py / optimization.ipynb

Representation

Most features return other arrays („signals“), for which different properties can be used as input features for the model.

Default property selection:

- mean
- std
- min
- max
- kurtosis
- skewness
- median
- var
- sum
- variance coefficient

Examples

Blood Pressure (ABP):

- Systolic Peaks, Diastolic Peaks, Distal Peaks
- Pulse Pressure
- Systolic–Diastolic Time Difference
- Pulse Transit Time (with PPG)
- Heart Rate (ABP-derived)

Electrocardiogram (ECG):

- Heart Rate (ECG-derived)
- Low-Frequency Power, High-Frequency Power, LF/HF Ratio
- LF/HF Ratio

Photoplethysmogram (PPG):

- Harmonic Ratio (H2/H1)

Spectral & Waveform

Raw Signals

- ECG (lead II, V)
- ABP
- PPG

No Capnogram used, due to lack of usage in other studies [1] [2]

Spectral Information:

- Wavelet Entropies (all)
- Spectral Centroid (all)

Filter:

Band-Pass was used for some signals. With both filtered and unfiltered signals used

Methodology

Feature Selection Optimization

see optimization.ipynb

Genetic Feature Selection

Evolutionary Optimization for Feature Selection (by sklearn_genetic)

Tested with different properties:

- max features
- goal metric
- estimators
- max depth

StandardScaler Pipeline was integrated into optimization at a later stage.

Outlier Removal as well, however this could not be used in the main training pipeline, due to fixed main.py

Possibilities & Limitations

Possibilities:

- Handles 200+ Features and Millions of Combinations
- Directly optimizing for different metrics

Limitations:

- Performance depends on configuration (genetic parameter, max_features etc.)
- May not fully represent the pipeline in model.py, so not directly optimizing for that
- Optimizer usually returns feature sets that are almost as big as the configured max_features

Results

(to change the version, replace “selected_features_select” in feature_comp.py)

These Features were often represented in the selections (Geneticly Optimized):

ABP Peaks (Syst, Dist), Pulse Transit Time (PTT), Wavelet Entropies, Pulse Pressure, Dist-Syst Time-Difference

Best Values: *AUPRC*: 0.491 (0.585*) *AUROC*: 0.989

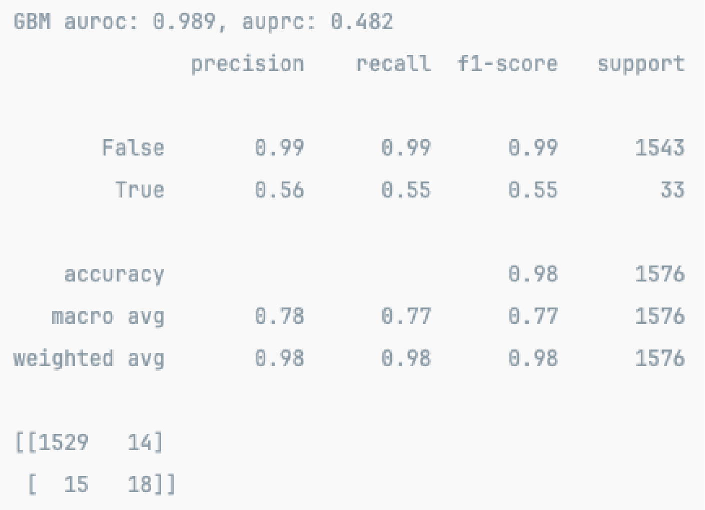
**after filtering for faulty data (about one dozen samples)*

Runs made by main.py (no modifications):

change feature selection version in feature_comp.py:514

V1: 73 Features *Tuned for Precision*

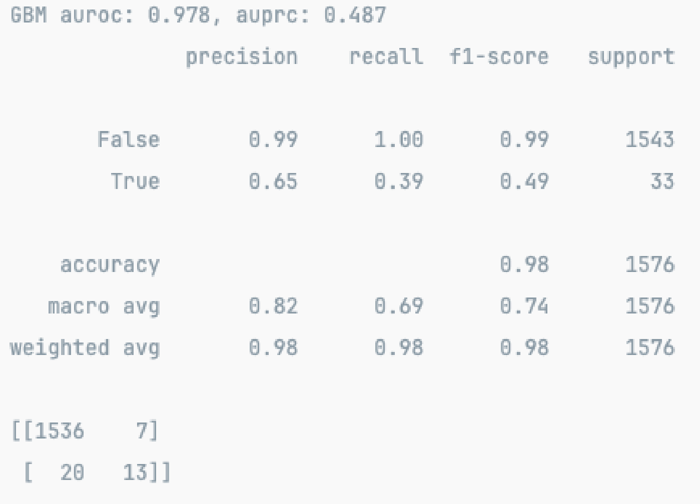
Max 75 Features



```
GBM auroc: 0.989, auprc: 0.482
              precision    recall  f1-score   support

       False       0.99      0.99      0.99      1543
        True       0.56      0.55      0.55        33

    accuracy                           0.98      1576
   macro avg       0.78      0.77      0.77      1576
weighted avg       0.98      0.98      0.98      1576

[[1529   14]
 [  15   18]]
```

v3: 19 Features *Tuned for AUROC*

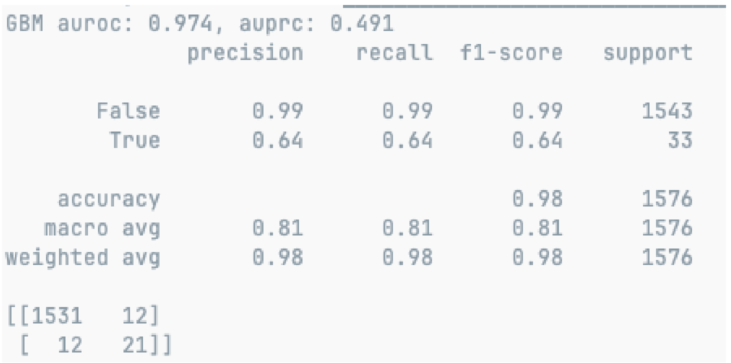
Max 30 Features



```
GBM auroc: 0.978, auprc: 0.487
              precision    recall  f1-score   support

       False       0.99      1.00      0.99      1543
        True       0.65      0.39      0.49        33

    accuracy                           0.98      1576
   macro avg       0.82      0.69      0.74      1576
weighted avg       0.98      0.98      0.98      1576

[[1536    7]
 [  20   13]]
```

V5: 37 Features *Tuned for AUPRC*

Somehow did not go for the .8 AUPRC with 0 TP in Confusion Matrix



```
GBM auroc: 0.974, auprc: 0.491
              precision    recall  f1-score   support

       False       0.99      0.99      0.99      1543
        True       0.64      0.64      0.64        33

    accuracy                           0.98      1576
   macro avg       0.81      0.81      0.81      1576
weighted avg       0.98      0.98      0.98      1576

[[1531   12]
 [  12   21]]
```

Discussion

Evaluation Metrics

AUROC:

- Value is always very high, but this is only due to imbalance

AUPRC:

- Highest achieved was 0.8, but the model does not return True Positives, only Negatives in the Confusion Matrix
- Due to it being calculated in the probability space, the auprc in some trials was high (>0.8), while the model never returns a probability >= 0.5

Faulty Data can not be filtered

No possibility to remove faulty data (like interrupted ABP)

Using the filtered dataset for training results in increased True positives (up to 27 TP) and higher AUPRC (+0.1)

You can see and test the removal of faulty data in „quick_start.py“ (line 29-35)

The data is highly imbalanced, and while Gradient Boosting mitigates this slightly, the forming of a generalized decision boundary is difficult

Fixed Model

The model / pipeline restricts the performance

State of the art studies use more sophisticated models to evaluate bio-signals like [3]

The importance of feature extraction and selection can be minimized by using a models that can extract data themselves from raw signal or general waveform transformations (Like CNNs or LSTMs)

Future Research

- Other Models
- Data Augmentation to fight imbalance in the dataset
- Data Inspection and cleansing, retrieving additional data
- Additional Visualizations, like extension of confusion matrix by showing different probability spaces
- Robust filters for accurate features (like Kalman for HR)

Sources

[1] Automated detection of severity of hypertension ECG signals using an optimal bi-orthogonal wavelet filter bank https://pubmed.ncbi.nlm.nih.gov/32768053/

[2] Systolic blood pressure estimation using ECG and PPG in patients undergoing surgery https://www.sciencedirect.com/science/article/pii/S1746809422005171

[3] Enhancing ECG classification with continuous wavelet transform and multi-branch transformer https://pubmed.ncbi.nlm.nih.gov/38434292/