ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

data analytics lab

Forecasting intracranial hypertension using time series and waveform features

Master thesis

Matthias Hüser

Department of Computer Science, ETH Zürich

Advisors: Dr. Martin Jaggi
Data Analytics Lab, ETH Zürich
Dr. Valeria De Luca
Computer Vision Lab, ETH Zürich
Supervisor: Prof. Dr. Thomas Hofmann
Data Analytics Lab, ETH Zürich

Wednesday 15th April, 2015

Abstract

Intracranial hypertension is an important risk factor of secondary brain damage after traumatic brain injury. In current clinical practice, the intracranial pressure (ICP) signal is invasively monitored and episodes of elevated ICP are manually identified on the waveform trace. As a result, clinicians diagnose intracranial hypertension *reactively*, and valuable time is lost as the condition is identified and treatment is instated. Instead, a more *pro-active* approach is desirable, in which the hypertension onset is forecasted, yielding an additional time window to prepare and deliver the intervention. Furthermore, because of the risks associated with the invasive measurement of ICP, a continuous ICP signal might not available in the intensive care unit. This further complicates the diagnosis and treatment of intracranial hypertension.

This work provides machine learning models to overcome these two limitations of current practice:

- A forecasting model allowing to *predict the onset of intracranial hypertension* from 5 up to 20 minutes in advance. This model relies on invasively measured ICP.
- A prediction model allowing to *non-invasively estimate the current ICP mean* from other channels (Blood pressures, Heart rate,...) whose collection carries a significantly smaller risk of endangering the patient. We are also investigating how this model can be exploited in forecasting of the ICP mean and thereby complementing the first forecasting model.

Both models rely on features extracted every 30 seconds from a multi-scale history (summarizing the last 30 seconds up to 256 minutes) of physiological channels, captured at high resolution (125 Hz), low resolution (1 Hz) or minute-by-minute. The features include statistical summaries, spectral summaries and morphological metrics of Arterial blood pressure and ICP pulses. We validate the generalization performance of our models using k=10 randomized fold validation on the publicly available MIMIC-II data-set.

Our results show that the two models surpass recently reported predictive models regarding classification and regression scores (Area under the ROC curve: 0.83, Mean absolute error: 3.84 mmHg respectively) on the MIMIC-II data-set. Furthermore we show that they generalize robustly to other patients and even data sets with different retrieval conditions, as we have confirmed on the Brain-IT dataset.

Additionally, we quantify the impact of including/removing different channels and feature types from the feature generation process with respect to the predictive performance of the two models. This allows us to draw conclusions about the economic design of an ICP forecasting framework.

Acknowledgements

I whole-heartedly thank my main advisors Dr. Martin Jaggi and Dr. Valeria De Luca for their strong involvement and key ideas that lead to new insights into the problem and their useful feedback given during our weekly meetings as well as during the write-up phase. Many thanks also go to Prof. Dr. Emanuela Keller for her feedback during the consultation in December and for introducing me to the related ICU cockpit project.

Moreover I want to express my gratitude to Prof. Dr. Thomas Hofmann, for giving me the opportunity to work on a topic that I found very interesting both from the machine learning and medical perspective.

I acknowledge the help of Dr. Rob Donald who made the Brain-IT data-set available, which was instrumental in validating the generalization capacity of our models.

Most importantly, I thank my parents for always supporting me and enabling my studies at ETH Zürich. I also acknowledge the support I have received from 'Studienstiftung des deutschen Volkes' throughout my MSc. studies.

Table of contents

Chapter 1

Introduction

1.1 Background and motivation

Traumatic brain injury

With an estimated 10 million cases traumatic brain injury (TBI) is the leading cause of death and permanent disability after an injury worldwide. It is usually caused by an insult to the head – most commonly during a fall or traffic accident.

After admission to the intensive care unit (ICU) and initial brain imaging diagnosis the primary injury usually leaves the focal point of attention.

Secondary brain injury

Rather, the major risk factor to be managed by the ICU treatment protocol is *secondary brain injury*, the permanent brain damage caused by e.g.

- Cerebral ischemia (Decrease of blood flow to the brain),
- Cerebral hypoxia (Decrease of energy substrate/oxygen flow to the brain),
- Brain herniation (Swelling leading to compression of brain structures)

and related conditions that lead to worse neurological outcomes or death.

It has been of great interest to researchers to understand how secondary brain injury comes about. The key variable explaining all three of the above conditions is *intracranial pressure* (ICP). ICP is determined by the overall volume of the cranial components: neural tissue, blood and the cerebospinal fluid. According to the Monro-Kellie doctrine [1] the total volume of the cranial system is constant and an increase of volume in one of its components can potentially increase ICP. One of the major concepts that emerged in the study of ICP auto-regulation is *intracranial adaptive capacity* (IAC) [2]. IAC is informally defined as the ability of the brain to maintain the blood- and energy substrate flow by holding cerebral pressure constant against slight volume changes of the cranial components.

Intracranial hypertension

The ICP value of a healthy adult is maintained by the IAC system in the range 7-15 mmHg. However, if the regulatory capacity is reduced rapid non-linear ICP elevations can occur. If such an elevation of ICP over 20-25 mmHg is sustained over more than 5 minutes it is defined as *intracranial hypertension* (ICH). If not immediately treated by cerebospinal fluid drain it can lead to all of the secondary injuries mentioned above [3]. Empirically, a direct association of ICH with clinical outcome has been established: the area under the mean ICP curve in the first 48 hours of ICU treatment is an independent predictor of in-hospital mortality [4]. Accordingly, it is consensus in the intensive care literature that intracranial hypertension must be avoided to optimize patient outcome in the ICU [5].

Shortcomings of current ICP monitoring

Invasive, intraparenchymal ICP monitoring combined with surgical interventions is the gold standard to control and maintain ICP in the physiological range below 15-20 mmHg and ensure adequate IAC [6]. Recent advances in monitoring and signal processing technology have allowed to take high-frequency ICP traces and analyze them in real-time [7]. However there are several limitations:

- Recorded ICP signals are corrupted by several types of noises and artifacts: High-frequency noise caused by measurement and amplifier devices as well as signal quantization; low-frequency noises due to patient movement, speech and coughing.
- Raw data is presented to the clinician and is not automatically translated into treatment recommendations. This can lead to information overload and over-consumption of human attention. A recent study reports that clinicians are not confident that effort spent on inspection of ICP traces is redeemed by improving outcome after TBI [8].
- Clinical expert systems with insufficient predictive performance, such as a too high false alarm rate, can desensitize staff to actually dangerous hypertension events.

Most crucially, interventions are only delivered after onset of intracranial hypertension, when negative effects might have already occured.

Forecasting of ICP elevation

To alleviate these problems, automatic *forecasting* of ICH onset could augment the current ICU protocol which is overly manual, reactive and prone to errors. The attention of clinicians could then be shifted away from tedious analysis of raw data to the monitoring of high-level patient state. Such a system would be able to identify the *pre-cursors to ICP elevation* automatically and robustly forecast impeding ICP elevations. While various forecasting models for ICH have been proposed, it is not clear if the error rates are sufficiently low for clinical implementation and how such systems fare under varying data quality and retrieval conditions.

Non-invasive ICP mean estimation

As mentioned earlier, ICP can only be reliably measured using an intraparenchymal probe that has to be inserted through a hole drilled into the skull. Often such an invasive procedure is not feasible because of the associated risks to the damaged brain. In these cases, where ICP measurement is impossible, one would still like to infer the current ICP mean from other channels, whose collection carries a substantially lower risk for the patient. Examples are the systolic and diastolic arterial blood pressures (ABP) and time series like the heart- and pulse rates. Having a machine learning tool that performs the estimation step in a fully-automated way is very desirable. Besides estimating the *current* ICP mean one could aspire to forecast the *future* ICP mean. Such a model could then be combined with the ICH forecasting model to give insight into the future development of ICP.

In this thesis we are treating the two problems separatedly, mostly because their general forms (classification/regression) lead to different formulations of machine learning models. However the features that are input to the statistical learning models could be the same – with the exception that no ICP-related features can be constructed in the case of the non-invasive ICP estimation problem.

Let us now proceed to define the two problems formally:

1.2 Problem definition

Let us assume that we have monitored and recorded a collection of physiological signals and time series for a certain amount of time. We are able to store summaries of this history (or the raw values) in some kind of memory.

Forecasting of ICP elevation

The *ICH forecasting problem* is then to predict, based on this stored history:

Is there an **onset of intracranial hypertension** in exactly t minutes?

The length of the forecasting horizon t has to be in the range 5-30 minutes for the system to be clinically significant. Alternatively we would like to maximize t while bounding the forecasting error below a clinically acceptable value.

Here we define the onset of intracranial hypertension as follows: In the 2 minutes preceding the reference point (t minutes into the future) the ICP mean is $\geqslant$ 20mm Hg for *all* 30 second windows in this range. In addition, in the 3 minutes preceding these 2 minutes the ICP mean has to be *below* 20 mmHg. According to this definition intra-hypertension episodes receive a negative label and are discriminated from pre-hypertension episodes.

Non-invasive ICP mean estimation/forecasting

Analogously, the *non-invasive ICP mean estimation problem* is to predict, on the basis of the stored history (which does not contain the ICP channel, or any of its derived channels, like CPP):

What is the **exact current ICP mean**?

When an additional forecasting horizon $t > 0$ minutes is introduced, we have the *non-invasive ICP mean forecasting problem*. One could then imagine a suitable horizon of e.g. $t = 10$ minutes.

1.3 Thesis contributions

Our main contributions are:

- A forecasting model for intracranial hypertension onset with an horizon of e.g. $t = 5, 10, 20$ minutes, which achieves an area under the ROC curve in the range $0.80 - 0.85$. It surpasses the generalization scores of other recently proposed models on the publicly available MIMIC-II data-set. It also generalizes to data-sets with different retrieval conditions, as tested using the alternative Brain-IT data-set.
- A non-invasive ICP mean estimation model which achieves a mean-absolute error of ≈ 3.80 mmHg on unseen patients in the publicly available MIMIC-II data-set.
- An implementation of statistical, spectral, complexity and morphological features defined on a collection of ICP and other neuro-physiological signals (ABP, Heart rate, Pulse rate,...)
- A software framework that can be used for online processing of ICP and other signals/time series. It is capable of automatic signal cleaning, feature generation and partial fitting of a stochastic gradient descent classifier/regressor.
- A detailed evaluation of the importances of different subsets of features (along different categories: sampling rate, underlying channel, feature type) for the generalization ability of the ICH forecasting model, allowing us to draw conclusions about the economic design of ICH forecasting systems in general.

1.4 Thesis organization

The thesis is structured as follows:

- In **Related work** we explain and contrast previously proposed ICH forecasting / ICP mean estimation methods with our models.

- **Data sets** presents the MIMIC-II data-set that was the primary source of samples for feature generation, and the Brain-IT data-set which was used to evaluate generalization capacity between different data-sets.
- In **Methods** we
  1. Present the software architecture and algorithms that were used to process signal segments from the MIMIC-II data-set and construct features from them.
  2. Explain the algorithms that were used to clean the signals, increase their signal-to-noise ratio and prepare them for feature extraction.
  3. Give an overview of the various types of features that were extracted from the cleaned signals and time series.
  4. Re-define the forecasting/estimation problems from the perspective of machine learning and introduce the Stochastic Gradient Descent method that was used to turn the constructed features into predictive models.
- In **Experiments and results** we quantify the forecasting/estimation performance of our models using two randomized fold validation schemes, and analyze the "importances" of subsets of the full set of constructed features.
- Finally, in the **Conclusion**, we summarize the main findings of the preceding chapter, discuss our methods and point out possible future extensions.

Chapter 2

Related work

Below we are surveying 3 different research directions related to forecasting of ICH, and contrast our methods/evaluation/results with these related works.

2.1 Forecasting of intracranial hypertension: Morphological features

Forecasting intracranial pressure elevation using pulse waveform morphology

Hamilton et al. present a method to forecast ICP elevations using morphological features [9]:
Data: A proprietary data-set from the UCLA medical center was used. As the MIMIC-II dataset it does not specifically contain patients that were admitted because of traumatic brain injury. It is not clear from the description if any patients with TBI were included in the data-set.
Problem: Forecasting horizons of 5,20 but also 35 minutes were considered. This is a longer horizon than we have tested in our experiments. The definition of intracranial hypertension is slightly stricter than ours: Whereas we define ICH onset as 3 minutes of ICP mean below the hypertensive threshold (20mm Hg) followed by 2 minutes above the hypertensive threshold, the authors define it as a 5 minute segment with ICP consistently $\geqslant$ 20 mmHg.
Methods: Their method uses one type of features (morphological) and is not considering the mean or diastolic ICP and other statistical summaries. Their method can be interpreted as an attempt to test the predictivity of the MOCAIP metrics and is not aiming for the most comprehensive feature set possible. For segmentation of ICP pulses and designation of sub-peaks the MOCAIP algorithm [10] was used. This algorithm also underlies the morphological features that we have constructed on the MIMIC-II data-set. However their method uses a more sophisticated version with reference libraries of legitimate ICP pulses. In contrast, we only filter individual pulses using heuristics such as expected intervals for pulse lengths. The scheme of extracting an averaged pulse is different to our approach: Whereas we average pulses point-wise to extract a representative pulse, their method clusters pulses and then extracts the centroid pulse as the representative pulse. The processing interval is also slightly different: Whereas our method extracts an averaged ICP pulse every 30 seconds, theirs uses a window size of 1 minute. The method to designate the 3 sub-peaks is more sophisticated than ours: Whereas we are using a completely derivative-based method, their algorithm searches in the set of all possible designations using a loss function. A quadratic discriminant classifier was used as the statistical model. In contrast, in our experiments we have used variants of generalized linear models with stochastic gradient descent (SGD) fitting. It is not clear whether a full predictive model was fitted, as it is only reported that the distribution of individual MOCAIP metrics was compared between episodes preceding intracranial hypertension and control segments. Their approach for searching for the optimal set of metrics is quite different from ours: They are solving an auxiliary problem using a Particle Swarm Optimization to find the optimal set of metrics. Our approach differs to this, as we are using the SGD classifier for ranking of features (using the absolute coefficients in the $\boldsymbol{w}$ vector)
Evaluation: To approximate the generalization performance of their model, the "Bootstrap" has been used. We are using 10-fold randomized fold evaluation which is a related re-sampling method. For the bootstrap sampling they are using an approach akin to ours, which preserves the initial distribution of

positive and negative examples in training and test sets.
Results: As classification scores they report a sensitivity of 90% at a specificity of 75% for a forecasting horizon of 5 minutes. Our tests with non-linear kernel SVMs yielded a similar operating point (in terms of sensitivity and specificity) for a forecasting horizon of t=10 minutes.

Forecasting ICP elevation based on prescient changes of intracranial pressure waveform morphology

In a related approach Hu et al. perform statistical tests to determine whether pre-intracranial hypertension segments can be discriminated from segments that are not associated with ICH (control segments) [11]:
Data: Their data-set contains only non-TBI patients with idiopathic hypertension. They obtain a similarly low number of positive instances as we have from the MIMIC-II data-set: 70 number of pre-hypertensive episodes were identified. With roughly 400 control segments they have sampled a similar number of negative instances compared to our experiments (924).
Problem: Their definition of "pre-ICH" segment ranges back up to 1 hour before the onset of the ICH. It is not clear if the model is able to forecast exactly when the hypertensive onset is going to occur, since only controls and different kinds of pre-ICH segments are discriminated.
Methods: In contrast to our work they are only using morphological metrics of the ICP pulse, so their model can be seen as a subset of our full model. A differential evolution algorithm was used to find the optimal set of MOCAIP metrics that minimizes a custom loss function. Our approach, in contrast, always uses the full set of feature columns and achieves feature selection using either L1-regularization or manual selection of subsets of feature columns. Parts of their MOCAIP feature generation process resembles the steps we have used to create ICP morphological features: Whereas the pulse segmentation step is almost identical, they use a more complicated scheme to designate subpeaks (defining an auxiliary optimization problem on all possible assignments). Our method is searching for relative extrema on the delineated pulse. Their heuristic to designate peaks when not all 3 are found was re-used in our work. As a machine learning model they have used a quadratic discriminant classifier which is passed as input feature vector an optimal subset of the MOCAIP metrics. In contrast, in our work we are using stochastic gradient descent with integrated feature selection.
Evaluation: Cross-validation was used, which is closely related to the randomized fold validation used in our experiments.
Results: They report the following operating point: 99.9 % specificity at 37 % sensitivity 5 minutes prior to onset of intracranial hypertension. We chose the opposite tradeoff between the two rates and have obtained a specificity of $\approx$ 55 % at perfect retrieval rates (sensitivity: 100 %). They do not state what their model's tradeoff is for higher sensitivities.

Intracranial hypertension prediction using extremely randomized decision trees

Most recently, Scalzo et al. have presented a model that predicts sustained intracranial hypertension using a set of morphological metrics and ensembles of randomized trees. [12]
Data: Their data-set is of similar size to ours: 30 patients from the UCLA Medical center, and is as generic as MIMIC-II: The patients have different ICH-related conditions and are not specifically treated for TBI. Biased sampling was used to avoid imbalance between classes. We have tried to address the same problem using a combination of biased sampling and adjusting the class weights in the loss function of the classifier.
Problem: The definition of intracranial hypertension is the same as in previous works by the same authors: 20mm Hg sustained over 5 minutes. Their experiments use times-to-onset between 1 and 20 minutes, as ours do as well.
Methods: Their method focuses on morphological features, whereas we have used a broader range of time series and waveform features. All features are based on the ICP channel, whereas our method also uses other physiological channels collected in the ICU. The formulation of feature vectors is similar to our history scheme: feature vectors are extracted from successive ICP segments. Our scheme has a more multi-scale flavour because the signal is resampled at various sampling rates. The MOCAIP

algorithm has been used to extract 24 morphological features. A subset of these metrics also forms the morphological ICP features of our models. The ICP beat-by-beat analysis is similar to ours: They are first extracting the QRS complex from the ECG, and are then using hierarchical clustering, which we did not attempt. Instead we are using a simple point-wise pulse averaging scheme. Also, we use a potentially less robust method for sub-peak designation compared to the Kernel Spectral Regression. Their signal pre-processing is similar to ours: They use a low-pass filter with a cutoff-frequency of 40 Hz – we are using a sharper cutoff at 20 Hz. As machine learning models their method uses ensembles of extremely randomized decision trees. While we have also checked performance of this type of model, it was not the focus of our experiments. Instead, we have mainly used stochastic gradient descent and generalized linear models.
Evaluation: They compare generalization performance between linear and non-linear methods, similar to what we have attempted in one of ours experiments. No tests were carried out with a longer horizon of > 10 minutes.
Results: They report AUROC scores (using the ensemble of randomized trees) of $\approx$ 0.85-0.89 for forecasting horizons of up to 10 minutes, which is slightly higher than what we have obtained on the MIMIC-II data-set.

2.2 Forecasting of ICH: Statistical summaries on minute-by-minute data

Novel methods to predict increased intracranial pressure during intensive care

Guiza et al. present a machine learning model that is capable of predicting ICH episodes up to 30 minutes in advance [13]:
Data: Their data-set consists of 264 TBI patients, while the primary data-set in our study (MIMIC-II) contains patients with different ICH-related conditions. They report that they have complete data available for 239 patients, which contrasts with the MIMIC-II data-set which is not complete and has missing values due to signal contamination.
Problem: Their definition of intracranial hypertension is more restrictive than ours: They define it as a ICP elevation over 30mm Hg *over 10 minutes*, whereas our definition includes shorter-lasting episodes of 2 minutes with ICP mean >20 mmHg.
Methods: For the feature generation they use minute-by-minute samples from the ICP, Mean Arterial Pressure (MAP) and Cerebral Perfusion Pressure (CPP) channels. Our method, in contrast, uses waveforms and signals (sampled at 1-125 Hz) besides minute-by-minute data. As statistical learning methods they are using logistic regression and Gaussian processes (similarity-based); the former is identical to the SGD method paired with the logistic loss, which was used for all our classification experiments. While constructing features they partition the time series into non-overlapping windows and compute summary statistics on them. In contrast, we use a multi-scale scheme, that resamples the history with different sampling rates and then computes features on these derived signals. The maximum length of the history that is used for feature generation is 4 hours, roughly the same as the 256 minutes, which was the coarsest scale of the multi-scale history in our prediction framework
Evaluation: Their validation data-set is patient-disjoint from the training data-set and hence, their evaluation method could be compared to the patient-stratified randomized fold validation, which we are using.
Results: They report a AUROC score of 0.87 ($t = 30$) in the validation set which is slightly higher than the AUROC of 0.84 we have obtained for $t = 20$ on the MIMIC-II data-set.
Summary: Overall, the work of Guiza et al. shows that even with relatively few channels a good discriminative ability can be obtained. Our work complements theirs as we present an analysis of other channels besides ICP, MAP and CPP and features built on high-frequency signals and 1 Hz time series.

2.3 Invasive ICP mean forecasting

Artificial neural network based intracranial pressure mean forecast algorithm for medical decision support

Zhang et al. present a non-linear auto-regressive neural network (ANN) model to forecast the future ICP mean [14].
Data: Their data-set consists of 53 patients that have been monitored for at least 24 hours each. Similar to our work, they also consider high-frequency signals sampled at 100 Hz.
Problem: A crucial difference to our approach is that they rely on ICP-derived input features, whereas we are solving the harder non-invasive ICP mean estimation/forecasting problems.
Methods: Their approach of using a history of windows and segmented sub-windows is similar to ours, where each window is sampled at a different rate. In general the feature generation process is related to the multi-scale history approach that was used in our work: k previous windows are defined and each of them is then sub-divided into a potentially different number of sub-windows. In contrast, our method does not sub-divide windows repeatedly, to avoid having a too large number of features. Their pre-processing consists in replacing artifacts/spikes by imputed data. We have used a similar technique with hard-coded bounds for the acceptable range of values. Compared to our work the range of calculated features is more limited: they only report mean, standard deviation and the regression line slope.
Evaluation: Their evaluation process does not test generalization between patients: their ANN is an online model in time to be used on individual patients.
Results: Similar to the results that we have obtained, they report that removing features such as variance/slopes has increased the predictive performance of the model. They speculate that this is due to the insignificant statistical association between mean and these two variables. They report a mean squared error of 0.88 for their best model. This is lower than the one we have obtained for the non-invasive ICP mean estimation problem. This is to be expected, because they are also using the ICP channel as a source of features. Thus the two scores are not directly comparable.
Summary: On the whole, their model performs well in the invasive ICP forecasting setting.

Conclusion

In comparison to the previous approaches explored above, our work contains the following innovations:

- For the first time a ICH forecasting model is constructed that combines several different feature types: Statistical summaries, spectral features, measures of time series complexity and morphological features extracted from different channels available in the ICU.
- Previously, synchronized waveforms and time series have not been concurrently exploited. Instead, either minute-by-minute or high-frequency waveforms were used. We are using waveforms and time series simultaneously.
- To the best of our knowledge, we present the first ICH forecasting model that has been trained and evaluated on the publicly available MIMIC-II data-set.
- We quantify the importance of individual feature categories for obtaining high discrimination, as measured by the AUROC score. Previously, various feature selection methods have been used (e.g. differential evolution) that rank the features and then re-train the model on the optimal set of metrics. No fine-grained analysis of the worth of individual feature categories was conducted.

Chapter 3

Data sets

3.1 MIMIC-II data-set

The MIMIC-II (multiparameter intelligent monitoring in intensive care II) database [15] is an ICU research database that contains high-resolution waveforms, time-series of vital signs as well as static clinical records. It is freely available to download from the web to support epidemiologic research and evaluation of clinical decision-support systems in critical-care medicine.

The entire data-set consists of 25,328 adult patient stays at the tertiary hospital 'Beth Israel Deaconess Medical Center', including cardiac-, surgical and general intensive care cases, in the years between 2001 and 2007. The data are retrospective: no interventions were part of the study design. They have been completely de-identified to comply with health act standards. In case that a patient has been admitted to the same ICU several times with a separation of at least 24 hours, these stays are recorded under different segment IDs.

Waveforms

The physiological signals include the different leads of the electrocardiogram, heart rate, arterial blood pressure (diastolic/systolic), oxygen saturation and respiration rate. They are sampled at 125 Hz using bedside monitor 'Component Monitoring System Intellivue by Philips Healthcare. Only $\approx$ 15 % of ICU stays include waveforms because of collection or archiving failures; no attempt was made by the designers of the study to assure that data containing waveforms are a representative sample of the population.

Time series

Time series are either updated at 1 Hz, are minute-by-minute trends that are constant over intervals of 60 seconds or have interpolated values.

Segment selection

Together with the data the authors have released the PhysioNet suite of data processing tools [16]. We have used the command-line utility `rdsamp` to extract individual segments as CSV files.

Only a small fraction of the data-segments contains ICP signals. Accordingly, as a first step of our data analysis we have discarded all segments that do not contain an ICP signal over their entire range. The remaining segments yielded roughly 43 GB of raw CSV text data which corresponds to 67 days of ICP signal measured at sampling rates 125 Hz or 1 Hz.

The selected subset of the database, which was used to generate features and conduct all experiments, consists of the following segments:

```
3142868, 3148126, 3160820, 3169632, 3189000, 3270954, 3270980, 3309132, 3319401,
```

```
3365681, 3453290, 3487247, 3543187, 3562822, 3624651, 3629298, 3642023, 3655233,
3656395, 3668415, 3688532, 3693937, 3700665, 3774557, 3938777
```

The segments have been cross-checked manually with the static patient information contained in the MIMIC-II database to ensure that no two segments correspond to different ICU stays of the same patient. In this way we can perform tests for patient-to-patient generalization of our machine learning methods.

Channel selection

In the next step, other channels (signals and time series) besides ICP were identified. The primary criterion for including a channel in the feature generation process was that the already selected segments have a large ratio of valid samples for the respective channel. This is motivated by our expectation that many invalid samples will lead to invalid features and these uninformative feature columns will "confuse" the machine learning models.

The next figure shows the ratios of valid samples by channel (pooled over all selected segments).

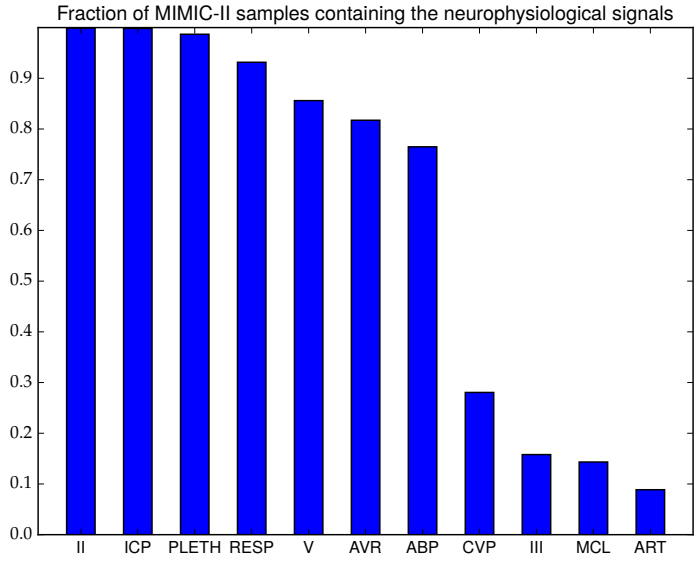


Figure 3.1: Fraction of samples of data-set that contain a valid value per signal

Based on this information, we have decided to use the `II` channel as the only lead for ECG analysis. Also the channels `PLETH`, `RESP` and `ABP` were deemed frequent enough to be useful as a feature source. In contrast, for example, the channels `CVP` and `ART` were discarded.

Target value distribution

Recall that our prediction problems of interest are:

- `onset_icp`: Existence of an intracranial hypertension onset in `t` minutes: A hypertensive onset is a 5 minute window of which the last 2 minutes are hypertensive and the preceding 3 minutes *not* hypertensive.
- `mean_icp`: Current ICP mean

We define intracranial hypertension as an elevation of the ICP over 20mm Hg.

Using these definitions, as a first data exploration step, we have looked at the distribution of the target values in the MIMIC-II data-set:

Segment	# positive `onseticp` instances	Mean(ICP) in mmHg	Std(ICP) in mmHg	Length in days
3142868	(0/16397)=0.00%	7.976	4.396	5.69
3148126	(3/5266)=0.06%	16.106	4.873	1.83
3160820	(0/2096)=0.00%	7.045	3.445	0.73
3169632	(0/2639)=0.00%	6.233	3.842	0.92
3189000	(1/3564)=0.03%	9.273	3.863	1.24
3270954	(1/4661)=0.02%	4.750	3.940	1.62
3270980	(3/19153)=0.02%	6.887	3.771	6.65
3309132	(0/5976)=0.00%	6.602	2.555	2.07
3319401	(0/7122)=0.00%	8.945	3.516	2.47
3365681	(14/16262)=0.09%	14.032	5.542	5.65
3453290	(8/8422)=0.09%	14.969	2.900	2.92
3487247	(1/1460)=0.07%	7.152	6.070	0.51
3543187	(6/8017)=0.07%	13.984	3.369	2.78
3562822	(0/3645)=0.00%	5.256	3.311	1.27
3624651	(0/2614)=0.00%	10.335	6.065	0.91
3629298	(4/8311)=0.05%	10.359	7.826	2.89
3642023	(0/3613)=0.00%	7.935	3.163	1.25
3655233	(16/20962)=0.08%	11.471	4.477	7.28
3656395	(0/1558)=0.00%	4.269	6.039	0.55
3668415	(0/6403)=0.00%	4.902	5.953	2.22
3688532	(26/11187)=0.23%	14.465	6.421	3.88
3693937	(0/2316)=0.00%	9.772	2.169	0.80
3700665	(1/2579)=0.04%	6.683	3.570	0.90
3774557	(0/2577)=0.00%	7.060	4.522	0.89
3938777	(3/2410)=0.12%	14.735	4.906	0.84

Table 3.1: Frequencies of positive `onset_icp` instances and summary measures of ICP mean (MIMIC-II data-set)

It is obvious from the data that the `onset_icp` classes are extremely unbalanced on the MIMIC-II data-set: There are only 86 positive labels among $\approx$ 170,000 instances. This has to be taken account when training and evaluating the classifier.

We also note from the Mean(ICP) and Std(ICP) summaries that

- These metrics vary significantly between patient stays: Therefore we expect that the non-invasive ICP mean regressor might have "trouble" generalizing beyond the patients contained in the training set.
- The standard deviation of the ICP mean is upper bounded by ≈ 6. This means that a naive mean regressor, which justs predicts the pooled global mean, will have an expected mean absolute error no greater than 6. Hence a non-invasive mean prediction, to be of any utility, needs to have a substantially lower error than this reference value.

Besides the MIMIC-II data-set we have used a second data-set to evaluate how a regression/classification model trained on the MIMIC-II data-set fares in out-of-data-set generalization:

3.2 Brain-IT data-set

The BrainIT data-set [17] originates in a multi-hospital study across 22 clinics in Europe and contains three types of information: static patient information (demographics) and physiological time series. We have obtained a subset of the data-base that contained only the following fields:

```
ICPs: Systolic ICP
ICPd: Diastolic ICP
ICPm: Mean ICP
```

and some static information (including sex and age) of the patient as well as information about the type and severity of the accident. As our models do not use static information but only physiological information we discarded all columns except `ICPm` and the time stamp. Our data-set contained 172 patients of which 85% are male and which have a median age of 36.2 years.

Segment selection

As we have only used the data-set as a test set for evaluation of our model's generalization performance we have decided to select the first 3 out of the available data segments, yielding $\approx$ 36,000 samples.

```
15026161,15026261,15027262
```

In the next table some statistical summaries about these segments are shown:

Segment	# positive `onseticp` instances	Mean(ICP) in mmHg	Std(ICP) in mmHg	Length in days
15026161	(6/5344)=0.11%	11.434	5.199	3.71
15026261	(0/1534)=0.00%	7.332	4.285	1.07
15027262	(66/28519)=0.23%	12.554	6.785	19.80

Table 3.2: Frequency of positive `onset_icp` instances and summary measures of ICP mean (BrainIT data-set)

Notably with 66 positive instances we obtain almost as many cases of intracranial hypertension onsets as in the much larger subset of the MIMIC-II database. The relatively "large" number of positive labels ensures that the scores of the generalization performance are robust and not unduly influenced by individual instances. We can also see that the values for the Mean(ICP) and the Std(ICP) are in the same range as for the MIMIC-II data-set, which means that we have not selected any "unusual cases" from the BrainIT data-set.

Chapter 4

Methods

For constructing the features from the MIMIC-II data we have implemented a medium-sized feature generation framework in Python ($\approx$ 14000 lines of code). We will present it here in a concise manner but note that it has been devised in a very general way: For instance it could be re-used for feature- and label generation from arbitrary physiological waveforms/time series.

4.1 Software framework

During feature generation the signals and time series are processed sequentially. One could also imagine the same process taking place as physiological signals are streamed in online from a collection of sensors in an ICU.

This sequential streaming process uses the following data structures:

- `Header`: The header holds information about the signals contained in a particular segment. It is only created once at the beginning of the streaming process and then never modified.
- `Window`: A buffer holding the last 30 seconds of the available channels. It is the synchronization unit of the entire prediction process. Once it is full, a new set of features is emitted. The chosen length of 30 seconds represents a tradeoff between the number of generated features and required processing time for feature generation.
- `PreProcessingPipelines`: This is a collection of individual pre-processing pipelines which are specialized for each type of channel. Each pre-processing pipeline consists of small stages (such as low-pass filters and interpolation algorithms) which can be readily plugged in/out in a modular way.
- `LabelMatrix`: A buffer for the labels emitted in the entire training process, i.e possibly spanning multiple segments. After the streaming process it is written out to a CSV file.
- `DesignMatrix`: A buffer for the design matrix (features) emitted during the entire training process. Like the `LabelMatrix` it spans multiple segments and is written to a CSV file after the feature generation process.
- `IcpMeanHistory`: It stores the history of the ICP mean up to some maximum history length (we set it to 600 minutes) length of the prediction horizon.
- `MultiScaleHistory`: Retains the history of all available channels at multiple scales. Each "level" of this history is a queue. Once a window is full, its samples are re-sampled at various sampling rates and then pushed to the appropriate queue. Each queue has associated a capacity which corresponds to the maximum size of the history that is stored at that scale.
- `ValueHistory`: The value history is the central computational unit of the feature generation process. As several features might depend on the same intermediate value (classical example: FFT of a segment) it is important that these are computed once and then retrieved from a cache. The value history ensures this by exposing a dictionary in which any intermediate value can be stored and later retrieved. As a special case, it also stores the entire history of computed features.

- `Values`: This is a container of functors that contain algorithms for generation of individual features and intermediate values. For example one of the `Values` is the Cooley-Tukey algorithm computing the DFT. It is accessed by the `ValueHistory` when a value, that has no yet been computed, is requested.

An overview of the classes used in the streaming architecture and their relations are shown in the figure below:

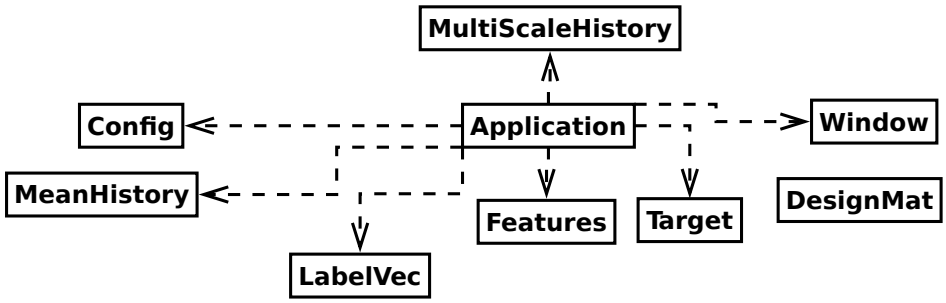


Figure 4.1: Overview of classes used for feature generation.

`Signal` and `TimeSeries`

To make the system easy to extend we have introduced the following classes for the signals and time series (only the ones used to construct the features are listed). All of these channels are stored in individual `Windows`, then stored in parts of the `MultiScaleHistory` and later used for feature construction:

Signals:

- `IcpSignal`: Intracranial pressure waveform
- `IISignal`: Main ECG lead waveform
- `AbpSignal`: Arterial blood pressure waveform (First artery)
- `RespSignal`: Respiration waveform
- `PlethSignal`: Fingertip oxygen saturation waveform

Time series:

- `ABPMean`: Mean arterial pressure
- `ABPSys`: Systolic arterial pressure
- `ABPDias`: Diastolic arterial pressure
- `CPP`: Cerebral perfusion pressure
- `ICP`: Intracranial pressure
- `HR`: Heart rate
- `PULSE`: Pulse rate
- `RESP`: Respiration rate
- `SpO2`: Oxygen saturation at finger tip
- `PVCRatePerMinute`: Rate of premature ventricular contractions

which are organized into a class hierarchy according to the type of the signal. For example `IISignal` is an sub-class of the `EcgSignal` and `IcpSignal` and `AbpSignal` are sub-classes of the general `BpSignal`. This hierarchy of signal classes is mirrored in a parallel hierarchy of `PreProcessingPipelines`, which are specialized to implement the data cleaning which should be run for a particular channel. This is justified, since for two different channels, rarely the same filtering stages should be applied (some channels are noisier than others, etc.)

4.2 Feature construction algorithm

The input to the feature construction algorithm is the current `MultiScaleHistory` which contains signals/time series of all channels for the last 256 minutes. The two outputs of the feature construction are a design matrix X and a label matrix *Y*, which are stored column-by-column into separate CSV files.

Features are computed in a sliding-window fashion over stretches of past data sub-sampled to yield a view on the signal "at different scales". There are 4 levels where the lowest level comprises the exact 4096 samples (30 seconds samples at 125 Hz) of a segment, but the highest level contains a sub-sampled version with an equivalent sampling rate of 0.1 Hz. Each level of this history might have a different length depending on available storage capacity.

The different scales that form the basis to feature construction are visualized in the next figure:

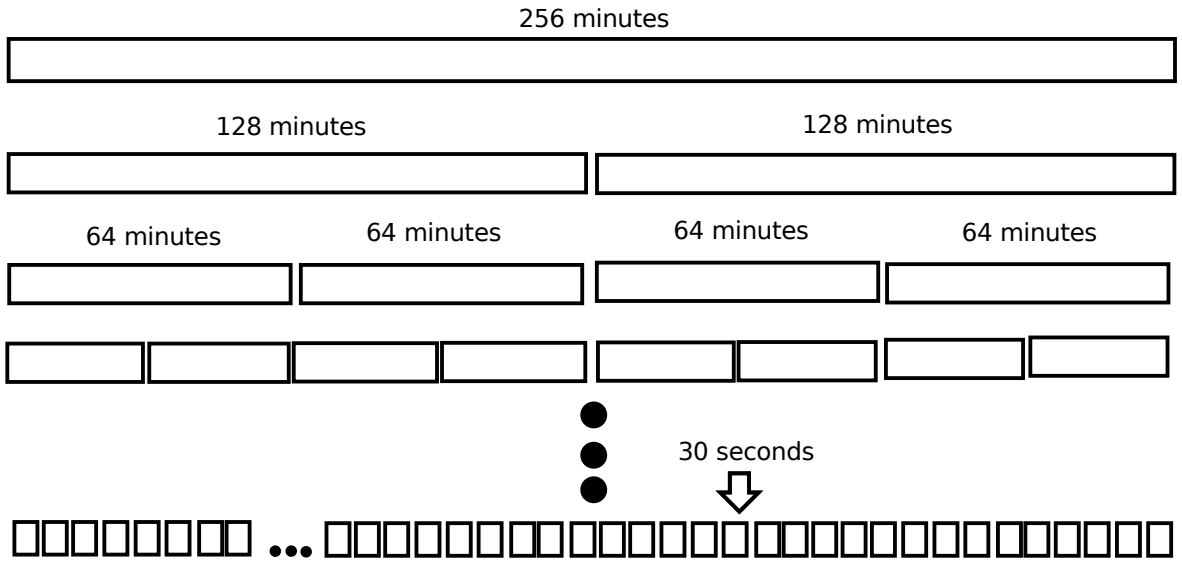


Figure 4.2: Overview of scales at which features are computed

At a high level the feature generation procedure works as follows:

1. Construct the multi-channel history, ICP mean value history and the pre-processing pipelines for the input signals/time-series
2. Construct an empty design matrix and an empty label matrix
3. Initialize an empty window that holds 30s of history
4. `FOR` every segment `DO`
   - Reset the multi-scale, ICP mean and value histories
   - Reset the design and label matrices containers
   - `FOR` all signal points of the segment `DO`
     a) Add the signal point to the current window
     b) `IF` window is full
        – `FOR` all time series points in the range up to the last fetched signal time point
          i. Add the time series point to the current window
        – `FOR` every signal `s` in {ABP, ICP, II, PLETH, RESP}:
          i. Get the array corresponding to signal `s` from the current window

ii. `IF` signal is not contained in the window: Store an invalid portion into the multi-scale history; and in case of an ICP signal store an invalid portion to the ICP mean history and continue with next loop iteration
iii. `ELSE:` Clean the signal `s` by passing it through its preprocessing pipeline
iv. `IF` the signal is marked `INVALID` after being passed through the preprocessing pipeline: Store an invalid portion to the multi-scale history; and in case of an ICP signal store an invalid portion to the ICP mean history
v. `ELSE`: Store the cleaned signal to the multiscale history and if it is an ICP signal store the mean ICP value to the ICP mean history.

– `FOR` every time series `t` in {MAP, ICP, CPP, ABP Dias, ABP Sys, SpO2, HR, PULSE}:
i. Get the array corresponding to time series `t` from the current window
ii. `IF t` is not contained in the window: Store an invalid portion into the multi-scale history and continue with next loop iteration
iii. `ELSE:` Clean the time series `t` by passing it through its preprocessing pipeline
iv. `IF` the time series is marked `INVALID` after being passed through the preprocessing pipeline: Store an invalid portion to the multi-scale history.
v. `ELSE`: Store the cleaned time series to the multiscale history.
– Reset the value history to the `DIRTY` state (no values have yet been computed)
– Generate the features in the `features` container by turn and store them along with a time-stamp into the design matrix
– Generate the labels in the `labels` container by turn and store them along with a time-stamp into the label matrix.
– Reset the current window to the empty state

5. Save the design matrix to the output CSV files, one file per feature contained in the design matrix.
6. Save the label matrix to the output CSV files, one file per label.

4.3 Signal cleaning

One of the problems in analyzing physiological signals is their often low signal-to-noise ratio and pollution with artifacts: In case of the ICP signal high-frequency noise is introduced from the sensors or the analogue-digital conversion process. Low-frequency noise comes from the patient itself, for instance through body movement and coughing. Lastly the sensor might be detached for a period of time but still outputting a signal.

4.4 Artifact removal

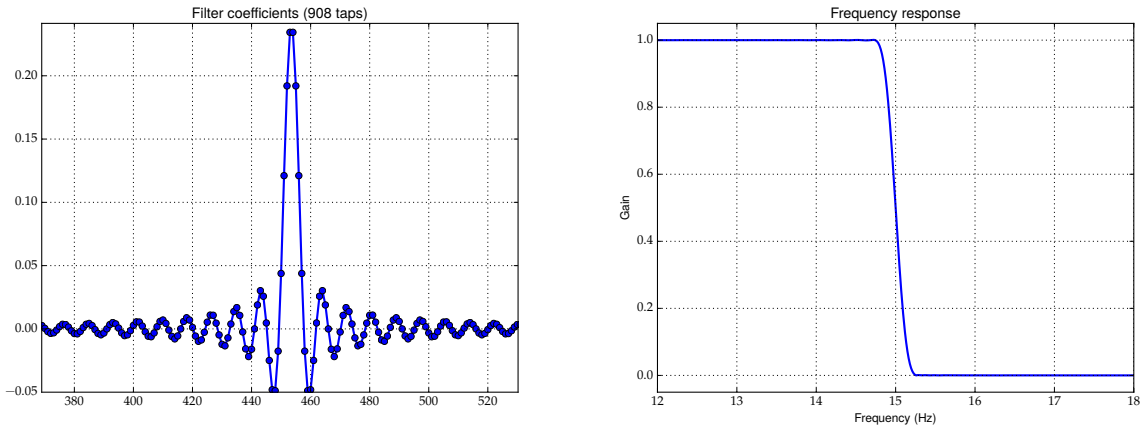
The preprocessing happens as a full window is processed by the `PreProcessingPipelines` class. We will explain this process for the example of the ICP signal:

1. Firstly it is decided if the ICP window should be discarded at the outset. Our criterion is as follows: If an ICP signal has more than 50% of its samples missing or is obviously corrupted (outside of the physiologically plausible range) it is marked “invalid”
2. Otherwise the missing samples are reconstructed from the neighbours via linear interpolation. If the neighbourhood is too sparse and no meaningful interpolation is possible, missing samples are set equal to the global window mean.

In a prototype of our system we have also experimented with other interpolation methods such as RBF kernels or spline fits but they were found to introduce rapid oscillations that are undesirable because they “pollute” feature generation further down the processing pipeline.

4.5 Low-pass filtering

To remove high-frequency noise originating in the ICP measurement system we apply a forward-backward zero-phase delay FIR filter. The filter coefficients and the magnitude response of this filter are shown below:



(a) Filter coefficients of FIR filter used for ICP preprocessing

(b) Frequency response of FIR filter used for ICP preprocessing

Figure 4.3

The end-product of this process is a signal window cleaned from artifacts and high-frequency noise. In the next figure an example of a 2-second segment of such a cleaned ICP signal window is shown. One can identify the characteristic 3 subpeaks and the following decay to the diastolic level of 10 mmHg

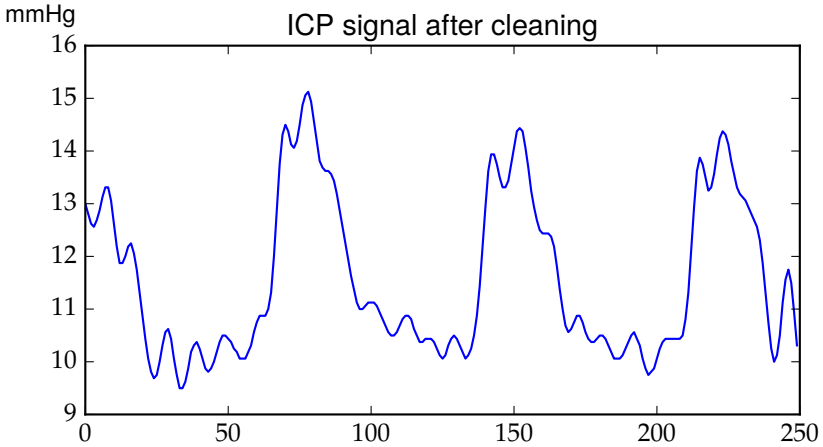


Figure 4.4: Cleaned ICP signal window with 200 samples, the x-axis is labeled by the sample numbers

4.6 Waveform and time series features

We will now present a small subset of the features that have been integrated as functors into the `Values` class. Features were either implemented based on their definition in the literature, based on library functions from SciKit-Learn [18], or on similar methods implemented in the PyEEG library [19]. We will concisely define individual features mathematically. Further properties and information about the features are available in the classical literature on statistics and signal processing.

4.7 Statistical summaries

In the following we denote a signal vector by $\mathbf{x}$. Typically its length is 1-256 minutes sampled at $0.1 - 125$ Hz. The length of the signal will be referred to as $|x| = n$. Some features could only be computed on smaller segments because of their high computational cost.

We start with some classical measures of signal location:

Measures of location

Maximum / Minimum

$$x_{MAX} = \max_{1 \leqslant i \leqslant n} x_i$$

$$x_{MIN} = \min_{1 \leqslant i \leqslant n} x_i$$

Mean

$$\mu_{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

which is also known as the first uncentralized moment of the signal.

We also obtain a range of order statistics of the signal. In general we denote by $x_{(i)}$ the i-th order statistic, for $1 \leqslant i \leqslant n$, such that there exist exactly $i - 1$ elements in $\mathbf{x}$ less or equal than $x_{(i)}$ and similarly there exist $n - i + 1$ elements in $\mathbf{x}$ greater or equal than $x_{(i)}$.

Median

$$M = \begin{cases} x_{(\lceil n/2 \rceil)} & \text{if } n \text{ is odd} \\ 1/2 \left[x_{(n/2)} + x_{(n/2+1)} \right] & \text{if } n \text{ is even.} \end{cases}$$

Additional level features that were constructed are:

- Geometric mean
- Euclidean norm (`sqrt`-transformation of the 2nd uncentralized moment)
- Average energy per sample (the 2nd uncentralized moment divided by n).

We will now come to measures of signal variability:

Measures of dispersion

Variance

Define the k-th uncentralized moment of the signal by

$$\mathbf{x}^{(k)} = \frac{1}{n} \sum_{i=1}^{n} x_i^k$$

Then the variance of the signal is

$$\sigma^2(\mathbf{x}) = \mathbf{x}^{(2)} - \left[\mathbf{x}^{(1)} \right]^2$$

Standard deviation

We also apply the SQRT transformation to the variance to obtain the standard deviation of the signal

$$\sigma(\mathbf{x}) = \sqrt{\sigma^2(\mathbf{x})}$$

Additional dispersion features that were used are:

- The standard deviation of the energy contained in sub-windows of the window, giving a measure of the medium-scale variability of the signal inside the window.
- The coefficient of variation: The ratio of the mean μ and the standard deviation σ. (also called signal-to-noise ratio)

The next features characterize the shape of the distribution of signal values in $\mathbf{x}$:

Measures of distribution

Skewness

Using the definition of the uncentralized moments from above, the skewness of the signal is defined by

$$g_1 = \frac{\mathbf{x}^{(3)}}{\left[\mathbf{x}^{(2)}\right]^{3/2}}$$

Its interpretation is complicated but for unimodal distributions a high skewness suggests that the left and right tails of the distribution have different "weights". A low skewness suggests a more symmetric distribution.

Kurtosis

Correspondingly the kurtosis is defined as

$$g_2 = \frac{\mathbf{x}^{(4)}}{\left[\mathbf{x}^{(2)}\right]^2}$$

It quantifies the "fatness" of the tails of the distribution compared to the energy in the values close to its peak (for unimodal, roughly symmetric distributions)

Measures of trend

We define 3 features quantifying the overall trend of a signal in the window:

- The simplest and least robust way to compute the change over a window is to calculate the difference between subwindows that cover the last and first 10 % of the signal, respectively.
- For robustness we also fit a linear regression line using least squares and define its slope as the trend over the signal.
- The correlation coefficient between the time positions and the values of the corresponding samples.

In addition to classical summaries we have also implemented some complexity measures. This was motivated by their potential to predict other physiological events: seizures, as recently established in an online challenge posed by the American Epilepsy Society.

4.8 Complexity summaries

Measures of complexity

Shannon entropy

To get a distribution we are first estimating a Gaussian probability density from the histogram of $\mathbf{x}$, using a kernel density estimator (Scott's automatic bandwidth estimation). Subsequently the entropy is found via the formula

$$S = -\sum_{x \in \mathcal{X}} \mathbb{P}[x] \log \mathbb{P}[x]$$

where $\mathcal{X}$ is the support of the estimated density $\mathbb{P}[\cdot]$.

Line length

The line length quantifies the short-term instability of the signal by taking the sum of the absolute differences between successive data-points. It has been profitably used in seizure detection [20]. Given a signal vector $\mathbf{x}$ it is defined by:

$$\mathrm{ll}(\mathbf{x}) = \sum_{i=2}^{n} |x_i - x_{i-1}|$$

Sample entropy

The sample entropy [21] measures the complexity of a physiological time series. It can be interpreted as the conditional probability that two subsequences of length $m+1$ are similar (according to a distance function) given that two subsequences of length-m are similar.

It is parameterized by the embedding dimension m and a tolerance r. To define it we need to develop some concepts first.

We start by defining a template vector of length m as

$$x_m(i) = (x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+m-1})$$

and a distance function $d(\cdot,\cdot)$ defined on two template vectors indexed by $i \neq j$, i.e. the Chebychev or Euclidean distance.

Let then B denote the number of vector pairs of length m over the entire $\mathbf{x}$ having $d(X_m(i), X_m(j)) < r$ and let A be the analogue quantity for sequences of length $m-1$. Then the sample entropy is

$$\mathrm{SampEntropy}(\mathbf{x}) = -\log \frac{A}{B}.$$

Approximate entropy

The approximate entropy [22] is a measure of the regularity of a signal and can be obtained from the following algorithm. Let m be the run length of the data and r a filtering level.

1. Form a sequence of embedding vectors $[\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(N-m+1)$ with each $\mathbf{x}(i) \in \mathbb{R}^m$ defined by $\mathbf{x}(i) = (x_i, x_{i+1}, \ldots, x_{i+m-1})$.
2. Use these vectors to construct for each i, $1 \leqslant i \leqslant N-m+1$:

   $$C_i^m(r) = (\text{number of } x(j) \text{ such that } d[x(i), x(j)] < r)/(N-m+1)$$

   where $d[x, x']$ is a distance function defined as

   $$d[x, x'] = \max_i |x_a - x'_a|$$

   and which represents the maximum difference over all scalar components of the vectors $\mathbf{x}$ and $\mathbf{x}'$.

3. Define
$$\Phi^m(r) = (N - m + 1)^{-1} \sum_{i=1}^{N-m+1} \log C_i^m(r)$$
4. The approximate entropy is finally
$$\mathrm{ApEn} = \Phi^m(r) - \Phi^{m+1}(r)$$
where m and r are fixed parameters.

Besides general measures of signal complexity we can study how close the signal is to truly chaotic behaviour and self-similarity. The classic technique for this is fractal analysis:

Measures of fractal dimension

Fractal dimension is an often used concept from bio-medical signal analysis; prior work has used it to characterize EEG signals in the time domain. Generally, estimating the fractal dimension consists of constructing an embedding sequence of the original time series using the delay method. Among its most common variants we have used Higuchi's and Petrosian's approximations:

Petrosian fractal dimension

The Petrosian fractal dimension [23] is defined in terms of the first order differenced time series $d\mathbf{x}$ of a signal $\mathbf{x}$. Denote by N_Δ the number of sign changes in $d\mathbf{x}$. Then the Petrosian fractal dimension is defined as

$$D = \frac{\log_{10} n}{\log_{10} n + \log_{10}\left(\frac{n}{n+0.4N_\Delta}\right)}$$

Higuchi fractal dimension

The Higuchi fractal dimension [24] is another measure of chaotic behaviour in a time series. Given the original signal $\mathbf{x}$ we construct k new time series $\mathbf{x}_k^m$ via

$$x_k^m = \left\{x_m, x_{m+k}, x_{m+2k}, \ldots, x_{m+\lfloor\frac{N-m}{k}\rfloor k}\right\}, \quad \text{for } m = 1, 2, \ldots, k$$

where m denotes the initial time values, and k is the interval between successive points. For each so constructed curve we define the *average length* by

$$L_m(k) = \frac{\sum_{i=1}^{\lfloor (n-m)/k\rfloor} |x_{m+ik} - x_{m+(i-1)k}|(n-1)}{\lfloor\frac{n-m}{k}\rfloor k}.$$

For each k we compute the average length over $m = 1, \ldots, k$, where k ranges from 1 to k_{max} as:

$$L(k) = \sum_{m=1}^{k} L_m(k)$$

and we then define the Higuchi fractal dimension D implicitly by

$$\log L(k) \propto D \log(1/k)$$

and hence D is the slope of a linear regression fit of $\log L(k)$ against $\log(1/k)$.

The next set of features quantifies the properties of the signal that are exposed when it is looked at as a random process:

Measures of random process

There have been several methods proposed to quantify the complexity of the spectrum of a time series. We have implemented 2 methods that were previously used in the analysis of the EEG signal.

SVD entropy

To find the SVD entropy [25] we construct a set of embedded shifted sequences of the original signal $\boldsymbol{x}$. These embedding sequences are then stacked into a matrix. Next the k singular values of the SVD of this matrix are found and stored in descending order in a vector $\boldsymbol{w}$. Normalize this vector by $\tilde{\boldsymbol{w}} = \frac{1}{\sum_{i=1}^{k} w_i}\boldsymbol{w}$. The SVD entropy is then given by

$$\sum_{i=1}^{k} -\tilde{w}_i \log \tilde{w}_i.$$

Fisher information in the spectrum

Similarly to above, calculate the stacked embedding matrix M and calculate the normalized singular spectrum $\tilde{\boldsymbol{w}}$. Then the spectral Fisher information [26] is calculated by

$$\sum_{i=1}^{k-1} (\tilde{w}_{i+1} - \tilde{w}_i)^2 / \tilde{w}_i$$

Detrended fluctuation analysis

The detrended fluctuation analysis [27] is a classical method to determine statistical self-affinity of a signal. It is computed as follows:

1. Compute the centered series $z_i = x_i - m$ where $m = \sum_{i=1}^{n} x_i$.
2. Divide the centered series (z_i) into windows of length L samples.
3. Fit a local linear least squares regression line inside every window by minimizing the squared deviation
$$E^2 = \sum_{j=1}^{L} (Y_j - ja - b)^2$$
yielding the best solution (a^*, b^*).
4. Calculate the fluctation for each window by
$$\left[\frac{1}{L} \sum_{j=1}^{L} (y_j - aj - b)^2 \right]^{1/2}.$$
5. Repeat the entire calculation for a range of values of L and plot $\log F(L)$ against L. The retrieved slope α, which corresponds on the original scale $F(L) \propto L^{\alpha}$ is the exponent of the detrended fluctuation analysis.

Note that the final exponent α is in $[0, 1]$ and the following cases are distinguished

- $\alpha < 1/2$: Anticorrelation
- $\alpha > 1/2$: Correlation
- $\alpha > 1$: Non-stationarity

allowing us to capture many important properties of $\boldsymbol{x}$ in one scalar.

Hurst exponent

The Hurst exponent [28] is a measure of the long term memory of a time series. In terms of the auto-correlation function it can be characterized as quantifying the decrease of auto-correlation as the time lag between values is increased. Its theoretical form is given implicitly by

$$\mathbb{E}\left[\frac{R(n)}{S(n)}\right] = Cn^H \text{ as } n \to \infty$$

where $R(n)$ is the range of the first n values of the time series and $S(n)$ is the standard deviation of the first n values of the time series.

We estimate it using the following algorithm:

1. Compute a "pyramid" of re-scaled time series of length $n/2, n/4, \ldots$.
2. Calculate the mean $m = n^{-1} \sum_{i=1}^{n} x_i$
3. Create the mean-subtracted version of the time series defined by $z_i = x_i - m$ with $i = 1, \ldots, n$.
4. Calculate the derived cumulative series $y_t = \sum_{i=1}^{t} z_i$.
5. Compute the range $= \max(y_1, \ldots, y_n) - \min(y_1, \ldots, y_n)$
6. Compute the standard deviation $\sqrt{(S(n) = \frac{1}{n} \sum_{i=1}^{n} (x_i - m))}$.
7. Calculate the quotient $R(n)/S(n)$
8. Average this quotient over the pyramid of time series created in the first step.

Spectral analysis is a classic technique for feature extraction from signals. It is often used for prediction of physiological events, because they are preceded by certain energy shifts in the spectrum. Our purpose is to quantify the changes with the following methods:

4.9 Spectral analysis

4.9.1 Fourier analysis

Denote the real input signal by $\mathbf{x}$ (sampled at 125 Hz). As preprocessing we apply a Hanning window to $\mathbf{x}$, yielding $\tilde{\mathbf{x}}$. Next we compute the discrete Fourier transform using the Cooley-Tukey algorithm [29] such that

$$X_k := [\mathcal{F}(\tilde{\mathbf{x}})]_k := \sum_{i=1}^{N} x_n \times \exp(-i2\pi jk/n)$$

for $k \in \{1, \ldots, n\}$ and where $j := \sqrt{-1}$ denotes the imaginary unit.

Because for real input the spectrum $\mathcal{F}(\tilde{\mathbf{x}})$ is symmetric about its midpoint, it suffices to use the first half for the computation of the magnitude spectrum. From this complex-valued output we obtain the amplitudes of the frequencies via:

$$|X_k| := \sqrt{\mathrm{Re}(X_k)^2 + \mathrm{Im}(X_k)^2}$$

This yields a sequence of spectral amplitudes for frequencies in the band $[0, \mathrm{Nyq} = 62.5]$ Hz. For the frequency binning we define the frequency bands $\alpha = [0, 1]$ Hz, $\beta = [1, 2]$ Hz, $\gamma = [2, 3]$ Hz, $\delta = [3, 6]$ Hz, $\epsilon = [6, 9]$ Hz, $\zeta = [9, 12]$ Hz and $\eta = [12, 15]$ Hz. Note that frequencies above 15 Hz have been attenuated by a low-pass filter during preprocessing. The spectral energy in each of the bands is defined as the L2-norm of the vector formed by the corresponding amplitudes $[X_k]$.

We thus obtain 7 spectral frequency features for each signal window $\mathbf{x}$.

As an example a characteristic frequency spectrum of the ICP signal over 30 seconds displayed in Figure 4.5.

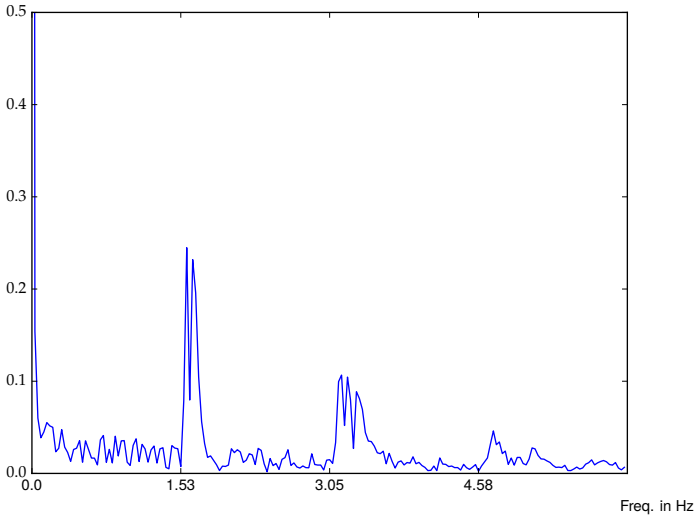


Figure 4.5: Frequency spectrum of the ICP signal over a window of 30 seconds, with the cardiac component around 1.6 Hz clearly visible

Besides the energy in the 7 bands we obtain as Fourier features:

- The frequencies and energies of the 5 largest FFT coefficients in absolute value.

4.9.2 Cepstral analysis

The cepstrum gives a complementary analysis to the frequency binning using the discrete Fourier transform. It can be interpreted as the first derivative of the power spectrum because it quantifies the change in power in the different frequency bands.

It is computed by taking the inverse Fourier transform of the logarithm of the estimated FFT magnitude spectrum. Formally the cepstral vector is defined by

$$\text{Cepstr}(\mathbf{x}) = \text{Re}(\mathcal{F}^{-1}(\log|\mathcal{F}(\mathbf{x}))|$$

where $\mathcal{F}(\cdot)$ denotes the discrete Fourier transform operator.

As features we then extract the 5 first coefficients of this Cepstrum.

4.9.3 Other spectral features

Hjorth mobility

The Hjorth mobility can be interpreted as the mean frequency of a signal and is commonly used in the analysis of electroencephalography signals [30]. It is defined by

$$\text{Mob}_{\mathbf{x}} = \sqrt{\frac{\sigma^2(\mathbf{x}\frac{d\mathbf{x}}{dt})}{\sigma^2(\mathbf{x})}}$$

where $d\mathbf{x}/dt$ is an approximation to the derivative of the signal $\mathbf{x}$.

Hjorth complexity

The Hjorth complexity [30] measures the similarity of a signal to a pure sine wave and the change in frequency over time. It is defined in terms of the Hjorth mobility as

$$\text{Compl}_x = \frac{\text{Mob}(x\frac{dx}{dt})}{\text{Mob}(x)}$$

4.10 Morphology of pulses

Morphological pulse descriptors are collections of features that describe the shape of pulses. The ICP morphological metrics that we have extracted are based on the pulse features of the recently proposed MOCAIP algorithm [31] which is a complete framework consisting of pulse segmentation, sub-peak designation and calculation of pulse-based metrics.

A subset of the morphological features is shown for a typical ICP pulse shape below:

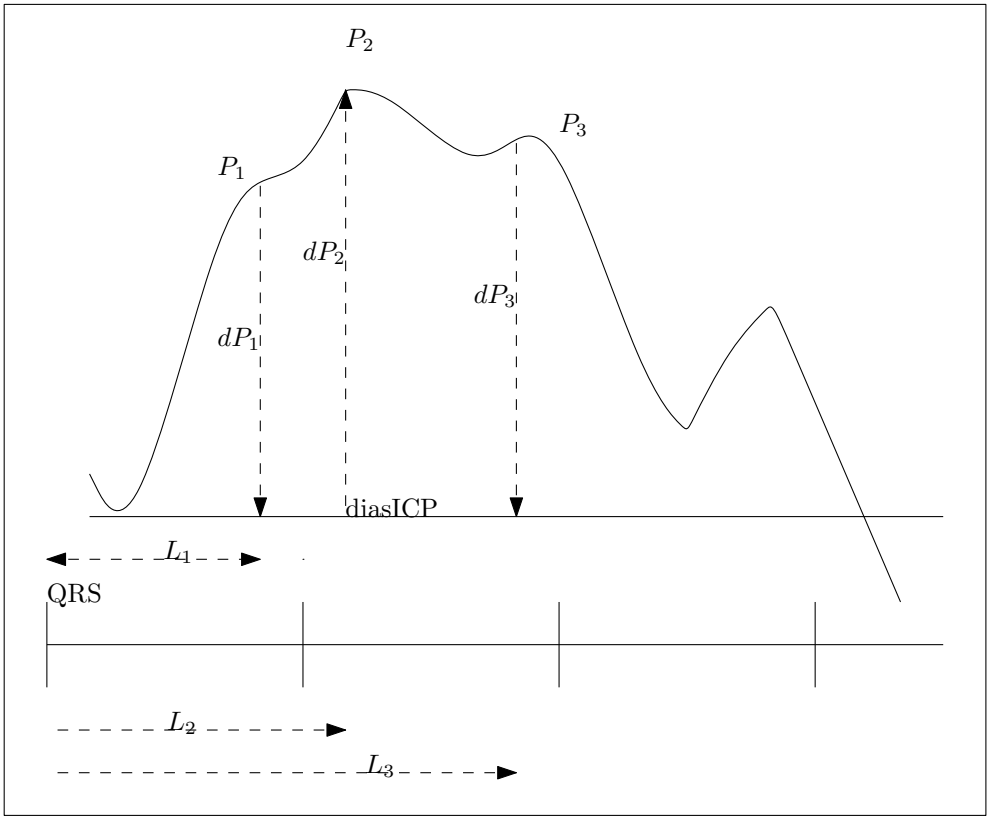


Figure 4.6: The sub-peak amplitude features dP_1, dP_2, dP_3 and the sub-peak latency features L_1, L_2, L_3 are shown for an typical averaged ICP pulse

4.10.1 QRS latency

A necessary preliminary for the detection of the ICP onset is the QRS detection based on a single-lead ECG signal. We have chosen to adapt the classic and robust method [32] for our purposes. An overview of the algorithm is given here:

INPUT: Continuous ECG measurement (lead-II) sampled at 125 Hz

OUTPUT: Locations of detected QRS complexes on the current window

- Apply a FIR (Kaiser window) filter with a pass-band of 5-15 Hz to x, yielding new band-pass filtered signal $\tilde{y}$.
- Apply a derivative (central difference filter of 2nd order) to $\tilde{y}$ to get y.
- Square the signal y point-wise to get z.

- Apply a moving window integrator such that the final output is

$$\tilde{z}_i := \sum_{j=i-w}^{i} z_j$$

for $i = w+1, w+2, \ldots, n$.

The window width w was chosen as 19 samples (150 ms at sampling rate 125 Hz).

Lastly each peak of the integrated waveform is either classified as a *QRS peak* or a noise peak; based on the levels of the peaks, two detection thresholds are continuously updated.

4.10.2 ICP morphological features

ICP pulse segmentation

To identify the onset of each ICP pulse, giving a segmentation into a sequence of ICP pulses, we have used the following algorithm modeled on the technique by [33]:

INPUT: The ICP signal of a window (30s) sampled at 125 Hz, and pre-filtered with a 15 Hz cutoff low-pass filter, the location of the QRS complexes on the current window, an estimate of the current heart rate HR

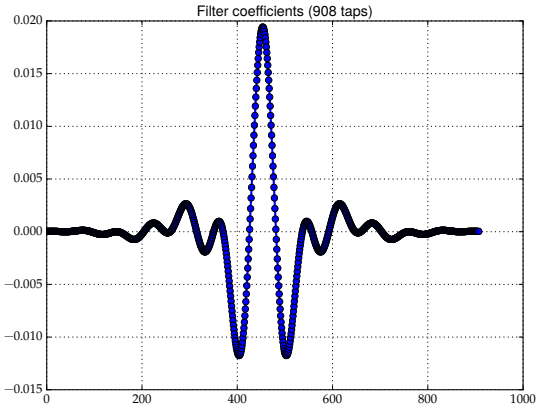
OUTPUT: A sequence of ICP pulse onset locations on the current window

1. Apply a band-pass filter (Kaiser-window with width 0.5/nyq) with pass-band $[0.45 \times \text{HR}, 1.5 \times \text{HR}]$ to the ICP signal, yielding new signal $\mathbf{x}_{\text{BP}}$.
2. Apply a low-pass filter (Kaiser window with width 0.5/nyq) with cutoff frequency of 5 Hz, yielding new signal $\mathbf{x}_{\text{LP}}$.
3. Set parameters of adaptive interval search as $\lambda_{\text{pre}} = 0.1, \lambda_{\text{pos}} = 0.1, \Delta_0 = 13, \Delta_1 = 19, \alpha_0 = 75, \beta_0 = 18$.
4. Initialize $\alpha_{\text{cur}} = \alpha_0, \beta_{\text{cur}} = \beta_0, p_{\text{prev}} = (\alpha_0 + \beta_0)/2, p_{\text{cur}} = p_{\text{prev}}$.
5. `FOR` each detected QRS location `qloc` in the window `DO`:
   - Define the pulse complex search window as $[\text{qloc} + \beta_{\text{cur}}, \text{qloc} + \alpha_{\text{cur}}]$
   - `IF` the pulse complex search window reaches outside $\mathbf{x}_{\text{BP}}$: TERMINATE
   - Search for relative maxima in the pulse complex search window, if none have been found, update adaptive interval bounds and continue next iteration
   - $p_{\text{prev}} := p_{\text{cur}}$
   - Define the location of current ICP `icploc` as the *leftmost* relative maxima found in the window
   - Define the onset search window as $[\text{qloc}, \text{icploc}]$.
   - Search for first relative minima in the onset search window, if none have been found continue the next iteration.
   - Define a horizontal line through the relative minima found.
   - Compute the derivative signal of the onset search window
   - Find a point p on the rising edge of the ICP pulse by taking the *first* relative maximum of the derivative signal, if none have been found continue with the next iteration.
   - Initialize left bound of rising edge as $p - 1$ and right bound of rising edge as $p + 1$.
   - `WHILE` the left bound and right bound are within the onset search window: Extend bounds to left/right by one sample and check whether a regression line fitted to [leftbound, rightbound] is still a line (has correlation coefficient $\geqslant 0.999$. `IF` still a line: Recall the current slope and intercept of regression line, `ELSE`: Break out of loop

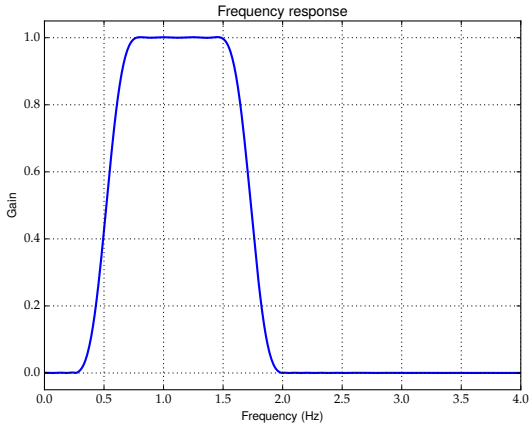
- `IF` no rising edge could be found: Continue with the next loop iteration
- Define the *onset index* as the intersection of the horizontal line from above and the rising edge regression line.
- Store the ICP onset location to the output vector
- Update the adaptive interval bounds:

$$\alpha_{cur} := \alpha_{cur} + \lambda_{pos} * (\Delta_1 - (\alpha_{cur} - p_{cur})) + \lambda_{pre}(p_{cur} - p_{prev})$$

The filter specifications are shown below, first the heart-rate dependent adaptive bandpass filter:



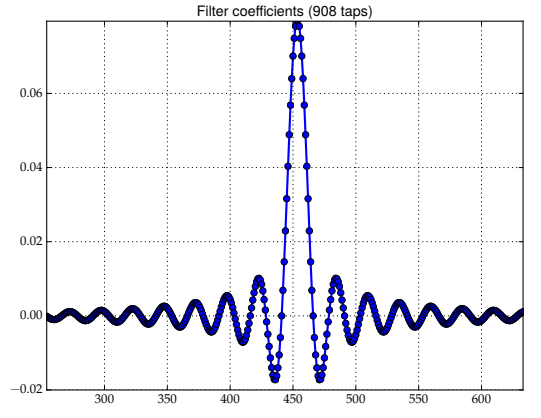
(a) Filter coefficients of Kaiser-window BP filter used for detection of ICP pulse onset



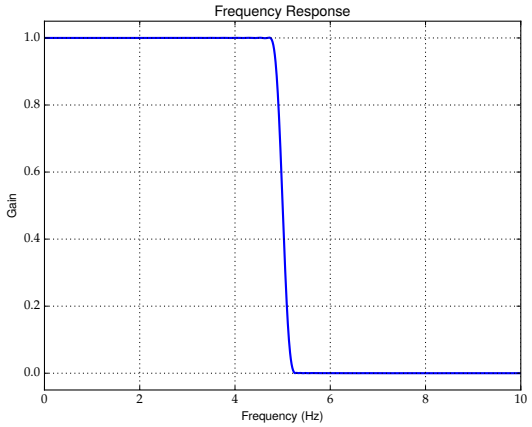
(b) Frequency response of heart-rate dependent band-pass filter

Figure 4.7

The low-pass filter specifications are shown below:



(a) Filter coefficients of Kaiser LP filter for detection of ICP pulse rising edge



(b) Frequency response of the 5 Hz low-pass filter

Figure 4.8

ICP pulse averaging

Often individual ICP pulses have a low signal-to-noise ratio and are polluted with high-/low- frequency noise. The extraction of the morphological metrics can be made more robust by averaging a sequence of detected ICP pulses before the characteristic sub-peaks are detected. The pulse averaging is described below:

INPUT: A sequence of detected ICP onsets, the corresponding ICP signal window.

OUTPUT: An average pulse describing the shape of the detected ICP pulses.

- Set pulsewidth-lowerbound := 62 (samples)
- Set pulsewidth-upperbound := 125 (samples)
- $n_{\text{validpulse}} := 0$
- Initialize the average pulse to the empty signal
- `FOR` each detected ICP onset `DO`:
  1. Extract associated pulse from ICP onset to next ICP onset
  2. `IF` pulse width is in [pulsewidth-lowerbound,pulsewidth-upperbound]:
     - Increment $n_{\text{validpulse}}$ by 1.
     - Resample pulse to canonical length of 100 samples
     - Incorporate pulse into the running average pulse
- `IF`: $n_{\text{validpulse}} < 5$: No average pulse can be computed, `RETURN` the invalid feature.
- Normalize the running average pulse

Note that this method is simpler than the more representative pulse clustering approach presented in [9], but is potentially less precise.

ICP sub-peak identification

Before metrics are extracted, the following sub-peak detection heuristic was applied to the averaged pulse computed in the previous step:

INPUT: The averaged pulse computed in the previous step

OUTPUT: The indices of the detected sub-peaks P_1, P_2, P_3 on the averaged pulse.

1. Detect local maximas as candidate sub-peaks
2. `IF` no relative maxima were found, `RETURN` the invalid feature
3. `ELSE IF` only one relative maxima was found: Assign the P_1, P_2, P_3 to this relative maxima
4. `ELSE IF` only two relative maxima were found: Assign P_1 to the first maximum and assign P_2, P_3 to the second maximum.
5. `ELSE IF` at least three relative maxima were found: Assign P_1 to the first, P_2 to the second, and P_3 to the third relative maximum.

Last the individual ICP pulse metrics, which were first proposed by [10], are extracted. For completeness they are listed below:

Levels

- dP_1, dP_2, dP_3: Amplitude of the sub-peaks
- $dP_{12}, dP_{13}, dP_{23}$: Ratios of above amplitudes

Time delays

- L_1, L_2, L_3: Latencies between the beginning to the first sub-peak and latencies to the second and third subpeaks.
- $dL_{12}, dL_{13}, dL_{23}$: Differences between the sub-peak latencies

Curvature

- $Curv_1, Curv_2, Curv_3$: Local curvature at the sub-peaks
- $Curv_{12}, Curv_{13}, Curv_{23}$: Ratios of the above curvatures

We have also extracted a set of morphological metrics from the ABP pulse. A different method was used, because the ICP pulse segmentation algorithm has certain hard-coded assumptions that make its application to the ABP signal impossible (in particular, the definition of the heart-rate dependent bandpass filters).

4.10.3 ABP morphological features

To delineate individual ABP pulses we have adapted the technique proposed in [34], using different threshold values based on manual evaluation on the MIMIC-II database. The algorithm works as follows:

INPUT: An ABP signal $\mathbf{x}$ sampled at 125 Hz.

OUTPUT: The sequence of identified ABP pulse onset locations on the current window

- Apply a low-pass filter (Kaiser window with width 0.5/nyqfreq) with cutoff frequency 16 Hz to $\mathbf{x}$ yielding the filtered signal $\mathbf{y}$
- Compute the first-order differenced vector on $\mathbf{y}$ yielding $d\mathbf{y}$ such that $dy_i = y_i - y_{i-1}$ for every $2 \leqslant i \leqslant n$.
- Truncate negative values in $d\mathbf{y}$ to 0.
- Compute a windowed slope sum function to yield a new vector $\mathbf{u}$ defined by

$$u_i = \sum_{i-w}^{i} dy_i$$

  for every $i = w+1, \ldots, N$. Here w is the window width and we chose it empirically as $w = 16$ samples.
- Define $\tau_{base} = 3\mu_{\mathbf{u}}$ as the base threshold and define a threshold $\tau = 0.6\tau_{base}$.
- `FOR` every sample in $\mathbf{u}$ `DO`:
  1. `IF` u_i is below τ, continue with next iteration
  2. Find the min/max values (u_{MIN}, u_{MAX}) in a search window $u[i-19, i+19]$.
  3. `IF` the difference of min-max value is too low, discard the pulse and continue the next iteration
  4. Define the search onset level as $l := 0.01u_{MAX}$
  5. Scan backward in time from u_i to beginning of $\mathbf{u}$ until we find a boundary index j such that $z_j \geqslant l$ but $z_{j-1} < l$. Define this boundary index as the detected ABP onset and store.
  6. Update the thresholds by $\tau_{base} = u_{MAX}$ and $\tau = 0.6 * \tau_{base}$.
- `RETURN` the collected ABP onset locations.

We are then extracting a subset of the features that were proposed in [35].

Last but not least, let us look at features that quantify the collective behaviour and inter-dependence of several signals:

4.11 Multi-signal correlation

We quantify the relation of the physiological signals `ICP`, `ABP`, `CPP` by finding their covariance and correlation matrices and further derived values.

Time covariance matrix

For a collection of signals $\mathbf{x}_1, \ldots, \mathbf{x}_k$ of same length $n = |\mathbf{x}_1| = \ldots = |\mathbf{x}_k|$ their covariance is matrix is defined by

$$[C_{ij}] = \frac{1}{n-1} \sum_{k=1}^{n} (x_{i,k} - \mu_{\mathbf{x}_i})(x_{j,k} - \mu_{\mathbf{x}_j})$$

Time correlation matrix

The correlation matrix rescales the covariance matrix to be in the same units of the original data as

$$[P_{ij}] = \frac{C_{ij}}{\sqrt{C_{ii} \times C_{jj}}}$$

Note that it is a symmetric matrix and hence all of its eigenvalues are real. Thus we can summarize this matrix by concatenating its upper right triangle (above and excluding the diagonal) and its real eigenvalues.

Spectral correlation matrix

Denote by $\mathcal{F}_{\text{mag}}(\mathbf{x}), \mathcal{F}_{\text{mag}}(\mathbf{y})$ the magnitude spectra of the two signals $\mathbf{x}$ and $\mathbf{y}$. We define the spectral correlation matrix by computing the covariance matrix $[C_{ij}]$ and the normalized correlation matrix using the magnitude spectrum as the input. Again the real eigenvalues of the spectral correlation matrix were extracted.

Besides the described features we also extract the 9 coefficients of a partial directed coherence analysis [36] (a summary of a multivariate regression model fit to a collection of signals).

4.12 Statistical learning methods

Having constructed all features that were presented in the previous section (each on many different physiological channels or combinations of channels), we stack them into a feature vector $\mathbf{x}$. Combining this process for all emitted samples (every 30 seconds), we stack these feature vectors into a design matrix X.

Let us recall the two prediction problems that we want to solve:

- `onset_icp`: Is there an onset of intracranial hypertension in t minutes from now?
- `mean_icp`: What is the current ICP mean?

The prediction target for the first problem is a label in $\{0, 1\}$; for the second it is a continuous value. Together with the matrix X, these labels, denoted by $\mathbf{y}$, encode instances of the classical machine learning problems: Binary classification and regression. These can be readily solved using generalized linear models and the stochastic gradient descent algorithm:

4.13 Stochastic gradient descent

For fitting of the constructed problem $(X, \mathbf{y})$ we have used the Stochastic Gradient Descent [37] algorithm for large-scale machine learning. In contrast to non-linear SVMs and non-parametric methods fitting is extremely fast with $\geqslant 100000$ samples and hundreds of features taking under 0.1s. In addition, it allows

interleaving the fitting and prediction steps, which would be necessary when our system is used in an online streaming context.

Stochastic gradient descent minimizes the following loss function over the entire training set $\{x_i, y_i\}_{i=1}^n$:

$$J(\boldsymbol{w}, b) = \left[n^{-1} \sum_{i=1}^{n} L(y_i, f(x_i)) \right] + \alpha R(\boldsymbol{w})$$

The factor α controls the amount of regularization applied to $\boldsymbol{w}$.

We are using the logistic loss which fits probability estimates of the positive class:

$$L(y, f(\boldsymbol{x})) = \log[1 + \exp(-y f(\boldsymbol{x}))]$$

where y is the true label, $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$ is a linear classifier and $\boldsymbol{x}$ is the input point in feature space.

On the other hand for regression we use the "ϵ-insensitive loss":

$$L_\epsilon(y, f(\boldsymbol{x})) = \max(0, |y - f(\boldsymbol{x})| - \epsilon)$$

where y is now the continuous response (such as the ICP mean) and $f, \boldsymbol{x}$ are as above. Note that this loss "ignores" errors of smaller than ϵ and is equal to the absolute loss beyond this threshold.

As regularization we take the Elastic-Net which is a convex combination of the well known L2- and L1-regularizers.

$$R(\boldsymbol{w}) = \lambda \|\boldsymbol{w}\|_2 + (1 - \lambda) \|\boldsymbol{w}\|_1 = \lambda \sum_{j=1}^{d} w_i^2 + (1 - \lambda) \sum_{j=1}^{d} |w_i|$$

where $\lambda \in [0, 1]$ is a mixing parameter that we have determined empirically to be 0.2/0.0 for optimal generalization performance in the regression problem and binary classification problems respectively. Note that for $\lambda = 0.0$ the elastic net reduces to the L2-regularizer.

During training the stochastic gradient routine approximates the true gradient of $J(\cdot)$ by considering only one training sample. Each example updates the model parameters $\boldsymbol{w}$ according to the rule

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \left(\alpha \frac{\partial R(\boldsymbol{w})}{\partial \boldsymbol{w}} + \frac{\partial L(\boldsymbol{w}^T \boldsymbol{x}_i + b, y_i)}{\partial \boldsymbol{w}} \right).$$

where η is the *learning rate* which is scheduled for time step t as

$$\eta^{(t)} = \frac{1}{\alpha(t_0 + t)}$$

with t_0 a heuristically defined value. (Optimal learning rate schedule)

For fitting of the model $(\boldsymbol{w}, b)$ we iterate over the entire training set only once - at the beginning of this sole iteration the data-points are permuted uniformly at random.

4.14 Feature selection

To optimize the generalization ability of the linear models feature selection is desirable. We have experimented with 3 different techniques for this:

- Sparsity by L1-regularization [38]: We set the regularization penalty λ such that only approximately a fixed number of variables are selected from the full set of variables. (e.g. only 100 variables for easy interpretability)
- Univariate statistical testing: We use the F-test to test the null hypothesis that a particular feature is a relevant predictor of the target variable. All variables for which the null hypothesis can be rejected at the α level were then selected. We set α in the range $[0.01, 0.05]$.
- Feature scores in L2-regularized optimal weight vector: For different channels and feature types we compute the sums of their coefficients in the optimal weight vector. If this sum is below a certain threshold τ all columns associated with this feature type or channel are deleted from the design matrix.

4.15 Encoding of missing values

As we have discussed in Section 4.3, in some cases, entire windows of signals have to be set to `INVALID`. As a result parts of the `MultiScaleHistory` will contain invalid samples. Before feature construction a linear interpolation step tries to impute these missing values from the neighbouring history. However if large stretches of signal are missing, these cannot be interpolated and remain `INVALID`. It will then not be possible to construct its dependent features. These invalid features are marked with the sentinel value `NAN` in our framework and are written to the design matrix X. Before passing the design matrix to the statistical learning model we transform it in the following way to ensure that it only contains numeric values:

1. Any sentinel value `NAN` will be replaced by the mean of its feature column. As columns are normalized this corresponds to replacing `NAN` by 0.0.
2. For each feature column in the original matrix we introduce a so-called "indicator column" whose entries are valued in $\{0, 1\}$. In the final design matrix feature- and indicator columns are interleaved such that for each entry X_{ij} in a numeric feature column, we have $X_{i(j+1)} := \mathbb{I}[X_{ij} = \mathrm{NAN}]$.

Clearly, this leads to a doubling of the size of the feature matrix. Our intuition is that this technique improves the predictive performance because the classifier can fit a column specific constant value for the case when a feature value is missing. In Section 5.17 we will test if that is the case indeed.

Chapter 5

Experiments and results

In Chapter 1 we defined the following forecasting/estimation problems of interest:

- `onset_icp`: Forecasting of the onset of intracranial hypertension in the future.
- `mean_icp`: Non-invasive estimation of the current or future ICP mean

In case of `mean_icp`, for concreteness, we only consider either the estimation problem with $t = 0$ or a forecasting horizon of $t = 10$ minutes. On the other hand, for `onset_icp`, we are evaluating our model's predictive utility for t either 5,10 or 20 minutes. These three choices represent different tradeoffs between *clinical utility*: how much additional time can be used to prepare an intervention, and *predictive performance*, as a-priori, we expect that the "problem difficulty" increases with growing t.

To quantify the predictive performance of our models we require a set of evaluation metrics that correspond to the utility that a prediction/forecasting model has in clinical practice.

From the perspective of machine learning, `mean_icp` and `onset_icp` are instances of two classical problems, namely

- `mean_icp`: Regression
- `onset_icp`: Binary classification

and we can use well-known evaluation metrics for each:

mean_icp: Regression

We consider the **mean absolute error** to quantify how close predictions are to the actual value of the ICP mean. For a set of target variables y_i and predictions $\hat{y}_i$ it is defined by

$$\text{MAE} := n^{-1} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Different from the mean squared error it can be intuitively interpreted on the original data scale. One of its disadvantages is that it does not distinguish between under- and over-predictions as both types of errors contribute via their absolute value. We are using it because it is an established measure to quantify the forecasting/prediction error of time series and related work uses it to quantify the ICP mean prediction error. To de-emphasive the influence of outliers in the score, we also report the **median absolute error** defined by:

$$\text{MDAE} := [\{|y_i - \hat{y}_i| \,|\, i = 1, \ldots, n\}]_{n/2}$$

where $[\cdot]_k$ denotes the k-th largest value of a set. The median absolute error quantifies the size of the "typical" prediction error, in that half of the errors are smaller and half of the errors are larger. However it does not penalize gross mispredictions as well as the mean absolute error.

`onset_icp`: Binary classification:

Our primary metrics are based on the receiver operating characteristic (ROC) curve. The ROC curve is a plot that summarizes the predictive behaviour of a model as its decision threshold (the least probability of the positive class for which the model assigns a positive label) is varied. Each of its so-called *operating points* represents a different tradeoff between the true positive and false positive rates. The *true positive rate* (or sensitivity) is defined as

$$\mathrm{TPR} := \frac{\mathrm{TP}}{\mathrm{P}}$$

where P is the number of actually positive instances (in our case true onsets of intracranial hypertension) and TP is the number of positive instances which have been correctly identified as being positive. For our problem it measures the capability of the system to retrieve actual hypertensive onsets. The *false positive rate* is defined as

$$\mathrm{FPR} := \frac{\mathrm{FP}}{\mathrm{N}}$$

where N is the number of actually negative instances (in our case episodes that occur before or during intracranial hypertension, but not at the beginning) and FP is the number of negative instances which have been incorrectly identified as being positive. In our context it measures the capability of the system to avoid false alarms; that is signaling an impeding hypertension, when actually none will occur in the near feature. The specificity is $1 - \mathrm{FPR}$.

It is possible to summarize the ROC curve by calculating the integral under the curve. Clearly, as each of TPR and FPR is valued in $[0, 1]$ also the integral will be a scalar between 0.0 and 1.0. A score of 1.0 implies that for each FPR (in particular for 0.0) the TPR is 1.0, which is a "perfect" prediction capability. An area of 0.0, on the other hand, means that the TPR is 0.0 - regardless of the FPR. Every value between 0.0 and 1.0 interpolates between these two extreme situations. An alternative interpretation of the AUROC is that it measures the probability that a randomly chosen positive instance will be ranked higher by the classifier than a randomly chosen negative one, where higher rank means a higher assigned positive class probability. Lastly, we note that a high AUROC is not a necessary condition for a suitable operating point that "works well in medical practice". However it summarizes the discrimination ability of a model in a general sense, without specifying the desired tradeoff between TPR and FPR in advance. Because of this it is used universally to evaluate intelligent medical decision systems. We will always be reporting the macro-averaged AUROC: this is calculated by finding the AUROC for each of the 2 classes separatedly (i.e. the roles of positive and negative class are considered both ways) and then finding their mean. In the envisioned clinical setting this is justified because both "missing" a future hypertension onset and producing too many false alarms can have grave effects: While for the first this is immediately clear – for the second ICU staff could be desensitized to the alarms and interventions would then not be carried out in time.

We will mostly be using the AUROC for conciseness, but will occasionally display the entire ROC curve. It is then possible to visualize the entire set of operating points and select one according to the desirable tradeoff between the true and false positive rates.

Approximation of generalization error

The generalization error of a forecasting/prediction model is a theoretical quantity, defined as the expected error on the unknown background population of samples. In our case this population can be imagined as all the possible summaries of a multi-scale history defined on collection of physiological channels. We would like to find out how large the forecasting error (with respect to the metrics introduced in the previous section) is in this generic situation - on expectation. Unfortunately this expectation cannot be calculated exactly because the distribution of such histories is unknown. However, an often used method to *approximate* the generalization error is **cross validation** (CV). For our purpose we are using its randomized variant: Randomized fold validation.

In randomized fold validation the data-set is repeatedly (k times) partitioned into a training and test set of pre-defined sizes. For each such partition the machine learning model is fitted on the training set and

then used for prediction on the test set. Depending on the problem type (regression or classification) we are then calculating test set scores and quantify the error that was incurred on the test set. In the last step, for each calculated metric, its values in the k-different folds are averaged to get a more precise approximation of the expectation of this error metric on the background population.

We are now explaining the two CV schemes that were used for the `onset_icp` forecasting problem. Note that because of the scarcity of positive labels it is very important that a suitable number are in both training and test sets, and relatedly: The proportion of positive instances should be roughly the same in the training- and in the test set. The next method guarantees that this is always the case:

Class-frequency stratified randomized fold validation

- **INPUT**: `nfolds`: The number of folds that should be used in the cross-validation, `test_ratio`: The size of the test set as a ratio of the entire data size n. We set it to 0.5 so that the evaluation of the error metrics is robust and not determined by too few samples.
- **OUTPUT**: The AUROC score averaged over the randomized folds

1. Pass over the data once and "remember" the data point indices that correspond to positive and negative instances. Also store the number of positive and negative instances as `n_pos` and `n_neg`.
2. Set `cv_score` to 0.
3. **FOREACH** fold in $\{0, \ldots, n\_folds\}$ **DO**:
   - Sample `test_ratio` $\times$ `n_pos` test instances from the data set without replacement
   - Sample `test_ratio` $\times$ `n_neg` test instances
   - Let the test set be the union of these sampled instances
   - Define the training set as the set difference of the entire data set and the test set.
   - **FOREACH** data point in the training set **DO**:
     - **IF** the buffer (1000 data-points) is full:
       - Permute the data points in the buffer uniformly at random
       - Partially train the SGD classifier on the permuted data points
   - **FOREACH** data point i in the test set **DO**:
     - Predict the probability that data point i belongs to the positive class using the learned SGD model.
     - Add both the true label and the predicted score to two vectors.
   - Calculate the AUROC score on the full set of true labels and predicted scores and add it to the vector of CV scores.
4. **RETURN**: The mean of the computed CV scores.

One shortcoming of the previous CV scheme is that it does not guarantee that the same patients or segments which are closely spaced in time are *not* present in *both the training and the test set*. Thus the evaluated scores do not quantify how well a model generalizes beyond patients (as it could just "remember" patterns in the data of an individual patient and then use them for prediction on the same patient). Hence, we have devised the following alternative CV scheme, which ensures that training and test-sets are *patient-disjoint*:

Patient-stratified randomized fold validation

- **INPUT**: `nfolds`: The number of folds that should be used in the randomized fold validation, `test_ratio`: The size of the test set as a ratio of the entire data size n, set to 0.5.
- **OUTPUT**: The AUROC score averaged over the randomized folds.

1. Pass over the data once and "remember" the number of positive and negative instances that are contained in each sample (corresponding to a unique patient) Also store the total number of positive and negative instances as `n_pos` and `n_neg`.

2. Define the minimum number of positive instances in the test set as

$$\texttt{npos\_min\_test} := \texttt{test\_ratio} \times \texttt{n\_pos}$$

and the minimum number of negative instances in the test set as:

$$\texttt{nneg\_min\_test} := \texttt{test\_ratio} \times \texttt{n\_neg}$$

3. In case of a regression problem define the minimum number of instances in the test set as

$$\texttt{n\_min\_test} := \texttt{test\_ratio} \times (\texttt{n\_pos} + \texttt{n\_neg})$$

4. Set the vector of `cv_scores` identically equal to 0.
5. **FOREACH** fold in $\{0, \ldots, \texttt{n\_folds}\}$ **DO**:
   - **WHILE**: Number of sampled negative instances is less than `n_min_test` or number of sampled positive instances is less than `n_pos_test`
     - Sample one segment from the remaining segments not yet sampled
     - Add the number of positive/negative instances in this segment to the current running total of positive/negative instances already sampled
     - Add all instances in the segment to the set of test instances
   - Define the training set as the set difference of the entire data set and the test set.
   - **FOREACH** data point in the training set **DO**:
     - **IF** the buffer (1000 data-points) is full:
       - Permute the data points in the buffer uniformly at random
       - Partially train the SGD classifier on the permuted data points
   - **FOREACH** data point `i` in the test set **DO**:
     - Predict the probability that data point `i` belongs to the positive class using the learned SGD model.
     - Add both the true label and the predicted score to two vectors.
   - Calculate the AUROC score on the full set of true labels and predicted scores, and add it to the vector of CV scores.
6. **RETURN**: The mean of the computed CV scores.

For all experiments we have used the machine learning package `ScikitLearn` 0.16.0-1 [18] on Python 2.7.9-1. The Mersenne twister pseudorandom number generator was seeded with the integer 99.

We are now ready to quantify the forecasting performance of our model on the `onset_icp` problem.

5.1 Classification scores of intracranial hypertension onset forecasting model

We will treat each forecasting horizon t in turn. First we consider $t = 5$, which represents the very minimum for a useful time window to prepare an intervention.

Hypertension onset forecasting $(t = 5)$

The free parameters of the SGD classifier were tuned to maximize the generalization performance of our model (see Section 5.16 for the details). This has yielded the following model:

```
loss: logistic
penalty: L2
alpha: 10.0
learning_rate: 1/(t+t0) (optimal schedule)
```

Here, and in all further experiments using SGD we pass over entire training data set once in randomly permuted order. For this and all other problems in this section we will separatedly report the capacity of our models to abstract from its training data and generalize to a) *other parts of the history* (where possibly the same patient appears in the training and in the test set) and b) *to unseen patients*. First, let us look at the supposedly easier problem, for which class-frequency stratified randomized fold evaluation was used:

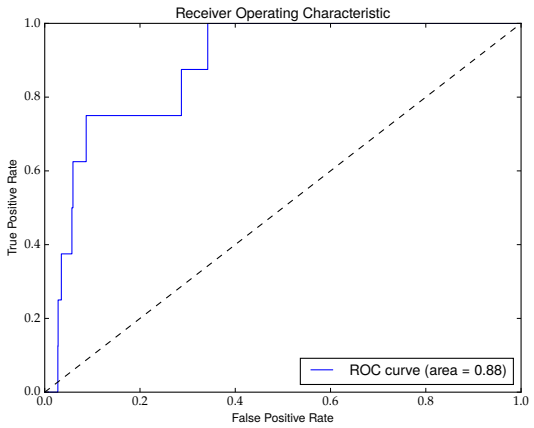
Generalization to other parts of the history

Model type	AUROC (macro-average)
Full model	0.85
Dummy model (no features)	0.50

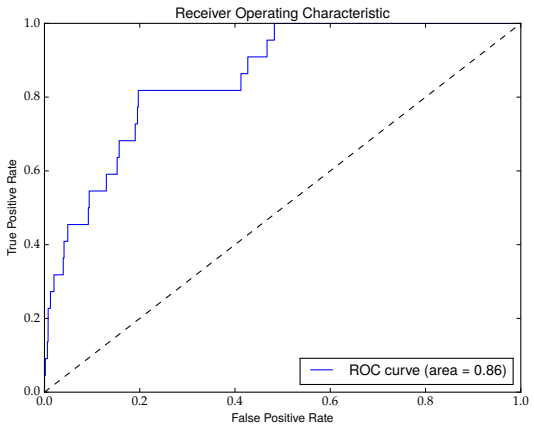
Table 5.1: Hypertension onset forecasting problem with $t = 5$ minutes

We can see that the full model ($d = 3812$) does substantially better than a dummy model which predicts at random according to the frequencies of the two classes in the training set. Its score 0.50 can be interpreted as no discrimination capability beyond random choice in a balanced data set with the same number of positive and negative examples. In contrast, an AUROC of 0.85 for our model means that it can discriminate much better than random choice and the entirety of used features is informative for the hypertensive onset forecasting problem.

As mentioned initially, the ROC curve gives a more complete view of our system's tradeoff between TPR and FPR at chosen operating points. In this and all following ROC curves we have only plotted the results of the first randomized fold and no averaging between folds was attempted, to avoid discontinuous curves (the different folds might lead to different sets of probability thresholds). Because of this, the curves give a weaker approximation to the true theoretical quantity than do the more robust AUROC scores, which are based on 10-fold cross-validation. The area mentioned in the legend of all following ROC curves refers to the AUROC obtained in the first fold.



(a) Hypertension onset forecasting problem with $t = 5$ minutes



(b) Hypertension onset forecasting problem with $t = 5$ minutes (Patient-stratified)

Figure 5.1

In this curve it is possible to identify both an operating point that favours correct retrieval of positive instances (TPR=1.0, FPR $\approx$ 0.35) and a more balanced one at (TPR=0.75,FPR=0.08). Depending on the clinical situation, one of these two would be typically chosen. Let us now look how the model fares in generalizing to unseen patients; here the patient-stratified randomized fold validation was used:

Generalization to other patients

For the SGD classifier the same values for the free parameters as in the previous experiment were used, in particular the parameter α was not fully retuned for this setting. Preliminary tests with a set of α's in close proximity to the original α^* have shown that changing it in either direction leads to insignificant changes to the AUROC (on the order of 0.005).

Model type	AUROC (macro-average)
Full model	0.83
Dummy model (no features)	0.50

Table 5.2: Hypertension onset forecasting problem with $t = 5$ minutes (Patient-stratified)

We can observe that the AUROC is slightly reduced from the 0.85 score that applied for the case when the same patient can appear both in training and test sets. It could be suggested that the SGD classifier is able to "remember" peculiarities from a patient's history which can then be used for more precise identification of onsets on the test data set.

Looking at the ROC curve, we can identify for which operating points the tradeoff between TPR/FPR is worse overall compared to the previous setting: For instance the very top right edge of the blue curve reaches further into the region of a high false positive rate. Whereas previously the FPR for perfect retrieval of positive instances was 0.35 it is now around 0.50, which is significantly worse. Secondly the characteristic horizontal line at TPR=0.80 is shifted to the right, such that this set of relatively balanced operating points has a higher FPR.

Hypertension onset forecasting $(t = 10)$

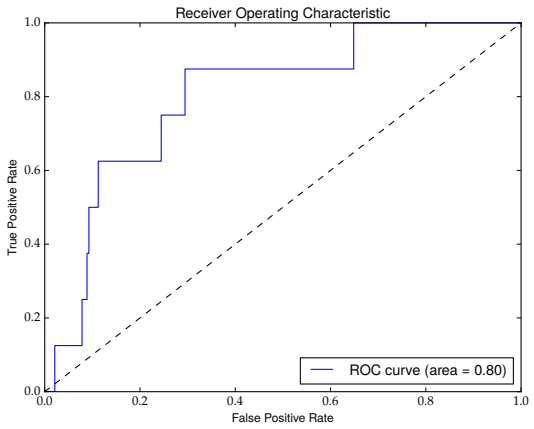
Next we have repeated all experiments for a forecasting horizon of $t = 10$ minutes. For both types of generalization the same SGD classifier parameters as for $t = 5$ were chosen. Again, experiments suggest that the optimal $\alpha^* = 10.0$ value works equally well in this case.

Generalization to other parts of the history

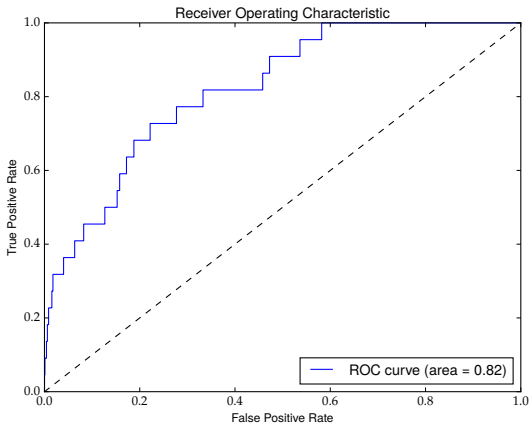
Model type	AUROC (macro-average)
Full model	0.83
Dummy model (no features)	0.50

Table 5.3: Hypertension onset forecasting problem with $t = 10$ minutes

We can see that, as expected, the AUROC is reduced from the 0.85 that was obtained for the same problem / CV scheme with $t = 5$. However, the decrease is minimal, and a look at the left ROC curve suggests that there are still usable operating points, like for instance: A sensitivity of 0.88 and specificity at 0.75.



(a) Hypertension onset forecasting problem with $t = 10$ minutes



(b) Hypertension onset forecasting problem with $t = 10$ minutes (Patient-stratified)

Figure 5.2

As for $t = 5$ minutes generalization to other patients yields a lower AUROC score (reduced from 0.83 to 0.80). The previous operating point with a sensitivity of 0.88 now has a much lower specificity.

Generalization to other patients

Model type	AUROC (macro-average)
Full model	0.80
Dummy model (no features)	0.50

Table 5.4: Hypertension onset forecasting problem with $t = 10$ minutes (Patient-stratified)

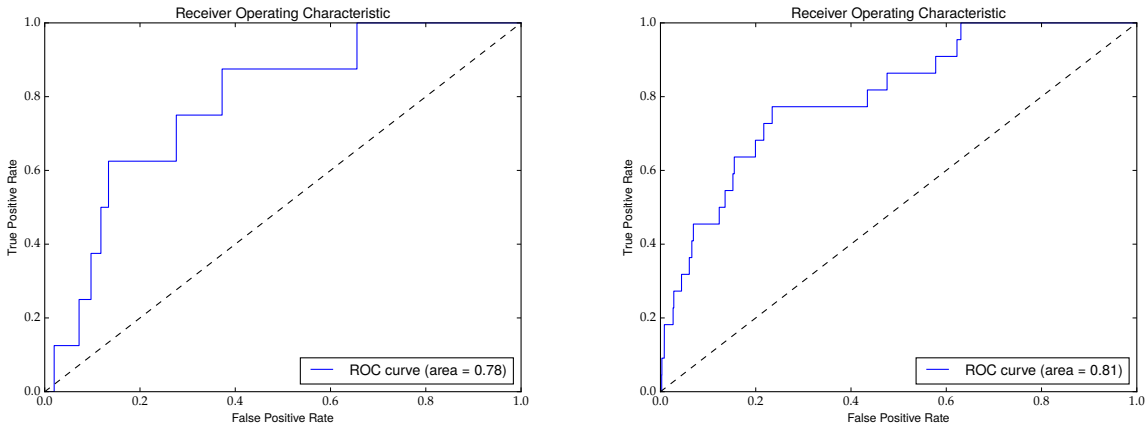
Hypertension onset forecasting $(t = 20)$

Last we look at "long-term" forecasting with a horizon of $t = 20$ minutes. The same parameters for the SGD classifier as for $t = 5, 10$ were used.

Generalization to other parts of the history

Model type	AUROC (macro-average)
Full model	0.84
Dummy model (no features)	0.50

Table 5.5: Hypertension onset forecasting problem with $t = 20$ minutes



(a) Hypertension onset forecasting problem with t = 20 minutes

(b) Hypertension onset forecasting problem with t = 20 minutes (Patient-stratified)

Figure 5.3

The AUROC score of 0.84 is competitive and even slightly higher than the one for t = 10 minutes. We suspect that this is due to random variations in the fold splits.

Generalization to other patients

Model type	AUROC (macro-average)
Full model	0.81
Dummy model (no features)	0.50

Table 5.6: Hypertension onset forecasting problem with t = 20 minutes (Patient stratified)

Generalization to other patients leads to slightly lower AUROC scores, as has been observed also for the experiments with t = 5 and t = 10 minutes.

Hypertension forecasting without invasive ICP information

In this experiment we quantify the predictive value of invasive ICP information for the forecasting of ICH (horizon t = 10). We use the following parameters of the SGD classifier:

```
loss: logistic
penalty: L2
alpha: 10.0
learning_rate: 1/(t+t0) (optimal)
```

We expect this to be a very hard problem as most information about future trends in the ICP value is contained in the ICP waveform/time series itself. In the non-invasive case ICP cannot be measured and a limited number of informative features remain.

AUROC (macro-average)
0.59

Table 5.7: Non-invasive hypertension onset forecasting problem with t = 10 minutes (Patient-stratified)

Indeed, the patient-stratified AUROC score shows that, currently, it is not feasible to forecast intracranial hypertension if the ICP signal is not available at the bed-side monitors.

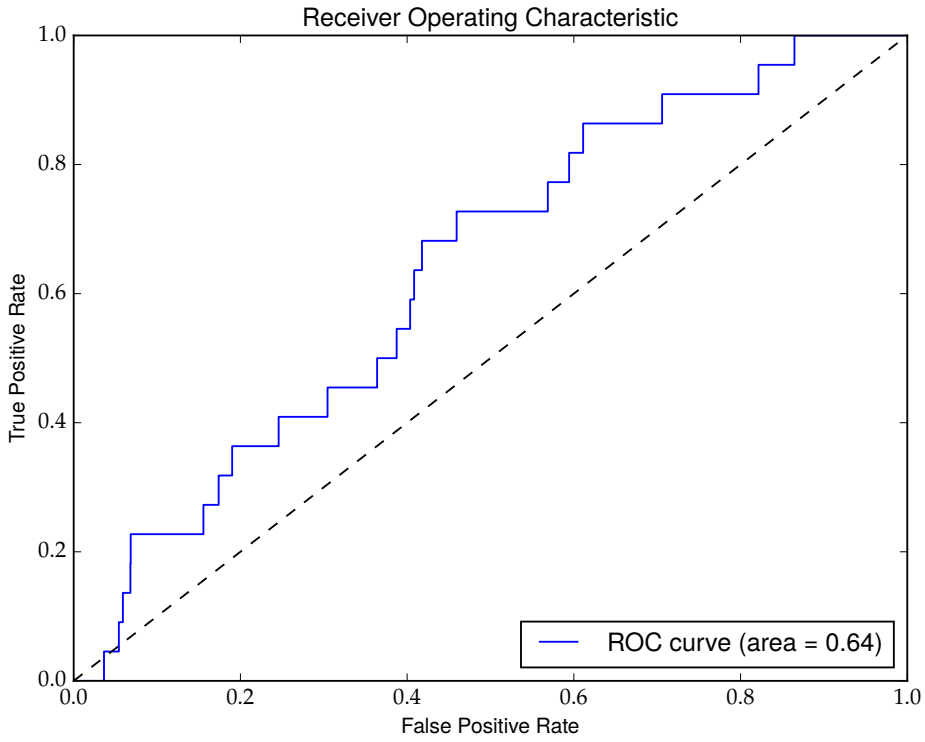


Figure 5.4: Non-invasive hypertension onset forecasting problem with $t = 10$ minutes (Patient-stratified)

The ROC curve illustrates well that the AUROC of 0.59 goes along with operating points that are not much better than random prediction. For instance an operating point of 0.90 sensitivity and 0.30 specificity has no clinical utility, because too many false alarms are triggered.

5.2 Comparison to minute-by-minute features by Guiza et al.

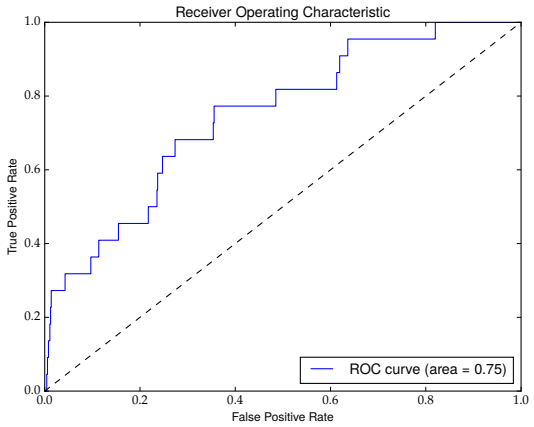
In this experiment we compare our full model to the recently proposed model by Guiza et al [13], which uses features calculated on minute-by-minute MAP/CPP/MAP time series. To enable a fair comparison we have resampled our time series (MIMIC-II) to one value per minute using a running mean smoother and then sub-sampled the data points. The features described in the electronic supplementary material of [13] were reproduced such that the reduced model has $d = 2 * 1041 = 2082$ columns, compared to our full model with $d = 3812$ features. We are using indicator columns to allow both models to handle missing values.

We are evaluating the AUROC scores on the hypertensive onset forecasting problem with $t = 10$ and use the optimized SGD classifier with parameters:

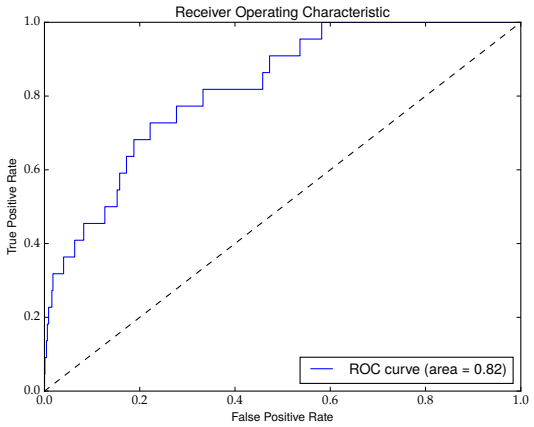
```
loss: logistic
penalty: L2
alpha: 10.0
learning_rate: 1/(t+t0) (optimal)
```

Model	AUROC (macro-average)
Full model	0.81
Guiza model	0.78

Table 5.8: Hypertension onset forecasting problem with t = 10 minutes (Comparison of full and Guiza model, patient-stratified)



(a) Hypertension onset forecasting problem with t = 10 minutes (Guiza model, patient-stratified)



(b) Hypertension onset forecasting problem with t = 10 minutes (Full model, patient-stratified)

Figure 5.5

We can see that, on the MIMIC-II data-set, our model shows slightly superior performance to the minute-by-minute model proposed by Guiza et al. On the ROC curves produced from the first fold's results we can observe that the balanced operating point with a sensitivity of 0.80 is slightly shifted to the left compared to the full model.

5.3 Comparison to morphological features by Hu et al.

In this experiment we compare our full model to the recently proposed model by Hu et al. [11], which is based on the 24 MOCAIP (Morphological clustering) metrics characterizing the shape of the ICP pulse. Since our model is a strict superset (it contains ICP morphological features and others) we can emulate it by discarding all other feature columns such that d = 48. (including 24 indicator columns for missing values).

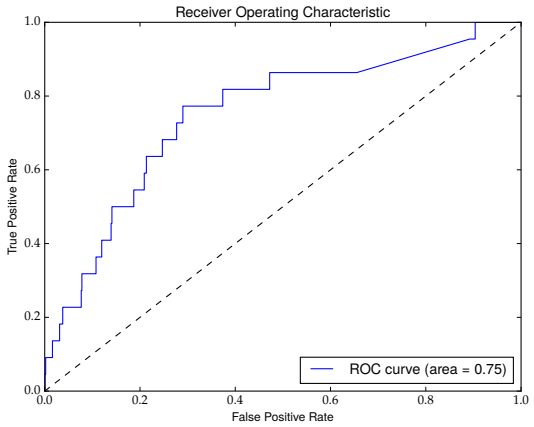
We are solving the t = 10 minutes hypertensive onset forecasting problem using the optimized SGD classifier with parameters:

```
loss: logistic
penalty: L2
alpha: 10.0
learning_rate: 1/(t+t0) (optimal)
```

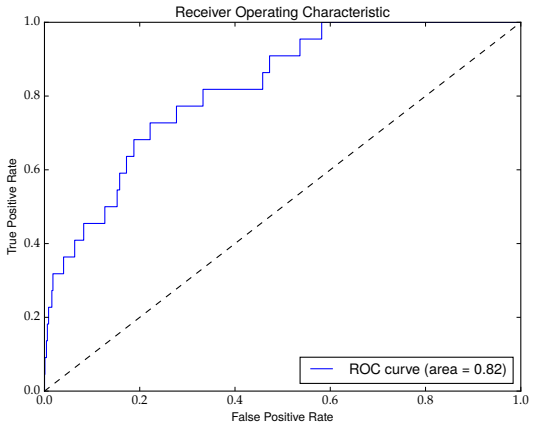
While this setting is optimized for the full model, experiments have shown that also the AUROC of the MOCAIP model is optimal for a regularization parameter of $\alpha = 10.0$.

Model	AUROC (macro-average)
Full model	0.81
MOCAIP model	0.74

Table 5.9: Hypertension onset forecasting problem with $t = 10$ minutes (Comparison of full and MOCAIP model, patient-stratified)



(a) Hypertension onset forecasting problem with $t = 10$ minutes (MOCAIP model, patient-stratified)



(b) Hypertension onset forecasting problem with $t = 10$ minutes (Full model, patient-stratified)

Figure 5.6

Interestingly, the MOCAIP model performs slightly worse than the minute-by-minute based proposed by Guiza et al. We suspect this is because of the low dimensionality of the MOCAIP and its sole focus on morphological features. We will check in later experiments whether morphological features are actually sufficiently predictive for the future ICH.

5.4 Generalization to BrainIT data-set

In this experiment we evaluate whether our model generalizes from the MIMIC-II to the Brain-IT data set. We are using the intracranial hypertension onset forecasting problem with a horizon of $t = 10$ minutes. The learning method is a SGD classifier with the following parameters:

```
loss: logistic
penalty: L2
learning_rate: 1/(t+t0) (optimal)
```

The classifier is trained on our entire subset of the MIMIC-II data-set ($n \approx 169000$). It is then validated on a subset of the Brain-IT data set containing $n = 36000$ samples; this corresponds to 3 patients with 25 days of minute-by-minute ICP data. We report the AUROC scores as the regularization parameter α is varied. All ICP-time series based features from the original model have been chosen so that only features which can be computed on both data-sets are included. This leaves $d = 848$ feature columns.

α	AUROC (macro-average)	Sensitivity	Specificity
0.001	0.78	1.00 (0.86)	0.18 (0.70)
0.01	0.77	1.00 (0.75)	0.18 (0.71)
0.1	0.76	1.00 (0.76)	0.16 (0.71)

Table 5.10: Hypertension onset forecasting problem with $t = 10$ minutes (Full model, Out-of-dataset generalization)

We can see that for the smallest α the generalization performance is maximized. This is expected because of the lower number of degrees of freedom with $d = 848$.

Lastly, we test the generalization performance of the model proposed by Guiza et al from the MIMIC-II to the Brain-IT data set:

Model type	AUROC (macro-average)	Sensitivity	Specificity
Full model	0.78	1.00 (0.86)	0.18 (0.70)
Guiza model	0.76	1.00 (0.78)	0.14 (0.70)

Table 5.11: Hypertension onset forecasting problem with $t = 10$ minutes (Comparison of Full model and Guiza model, Out-of-dataset generalization)

We can see that the out-of-dataset generalization performance of the Guiza model is slightly worse in terms of AUROC score. However still a decent balanced operating point (Sensitivity: 0.78, Specificity: 0.70) can be used. This is illustrated on the following ROC curve:

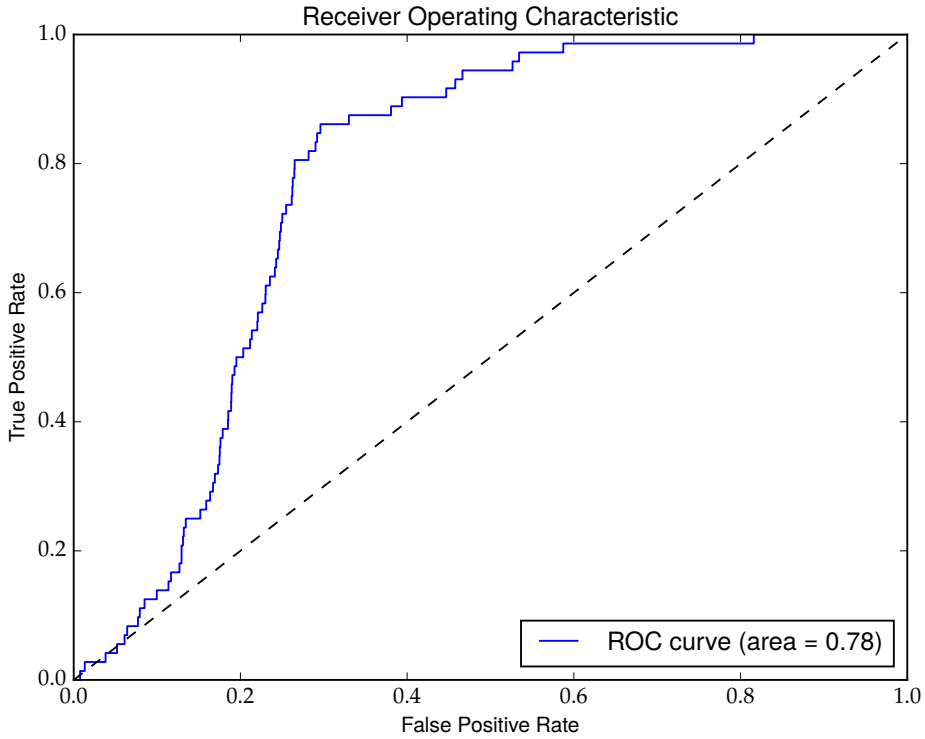


Figure 5.7: Hypertension onset forecasting problem with $t = 10$ minutes (Full model, Out-of-dataset generalization)

5.5 Regression scores of non-invasive ICP mean estimation model

Non-invasive ICP mean estimation ($t = 0$)

In this experiment, we test the generalization performance of the regression model for the non-invasive ICP mean estimation problem. We test both the generalization to different histories (with the same patient possibly appearing in both the training and test sets) as well as generalization to different patients (with one patient either appearing in the training *or* in the test set). We report scores for the regression metrics: Mean absolute error and Median absolute error. We have found empirically that iterating over the data-set 10 times with reshuffling after each iteration gives the best performance scores.

Generalization to different parts of the history

We use the following parameters of the SGD regressor:

```
loss: epsilon_insensitive
alpha: 0.001
l1_ratio: 0.2
learning_rate: eta0/pow(t,power_t)
eta0: 0.01
power_t: 0.25
```

Model type	Mean absolute error	Median absolute error
Full model	3.84	3.04
Dummy (no features)	4.29	3.89

Table 5.12: Non-invasive ICP mean estimation problem with $t = 0$ minutes

We can see that our model performs somewhat better than the dummy model (always predicting the global mean) but it's performance is still somewhat lacking. It seems that with only ABP, blood pressures and heart rates available it is "hard" to predict the current ICP mean.

Generalization to other patients

In the next experiment we will evaluate the generalization performance to new patients. The same parameters for the SGD regressor as in the previous experiments are used. SGD only iterates once over the data set and the data is not shuffled but read in sequential order. We measure the same set of regression metrics:

Mean absolute error	Median absolute error
6.53	5.43

Table 5.13: Non-invasive ICP mean estimation problem with $t = 0$ minutes (Patient stratified)

We can see that the model's performance significantly decreases compared to generalization to other parts of the history (no patient stratification).

Non-invasive ICP mean forecasting ($t = 10$)

In the next set of experiments we evaluate the generalization ability of the SGD regression model for non-invasive ICP mean forecasting. Similarly to above we evaluate generalization to other parts of the history and other patients separatedly:

Generalization to other parts of the history

Model type	Mean absolute error	Median absolute error
Full model	3.89	3.12
Dummy (no features)	4.32	3.88

Table 5.14: Non-invasive ICP mean forecasting problem with $t = 10$ minutes

We can see that forecasting of the ICP mean is not much harder than predicting the ICP mean at the current point in time, as the median absolute error is only slightly increased compared to the prediction problem with $t = 0$.

Generalization to other patients

Mean absolute error	Median absolute error
6.95	6.16

Table 5.15: Non-invasive ICP mean forecasting problem with $t = 10$ minutes (Patient-stratified)

Lastly, generalization ability between patients yields higher mean absolute errors than both generalization ability for the non-invasive prediction problem as well as non-invasive forecasting without patient stratification.

5.6 Feature importances I: Waveform or Time series

In this experiment we quantify for each series type (signal or time series), how much its derived features affect the model's performance, when they are used as the sole features or when they are removed from the full model, respectively. As a reference model we take the full model with all features ($d = 3812$) trained on a tuned SGD classifier with a logistic loss. We use the hypertensive onset forecasting problem with $t = 10$ minutes and report changes to the AUROC score.

Model	AUROC (Sole)	AUROC (Removed)
Full model	0.80	N/A
SIG	0.81 [+0.01]	0.79 [-0.01]
TS	0.79 [-0.01]	0.82 [+0.02]

Table 5.16: Hypertension onset forecasting problem with $t = 10$ minutes (Feature importance by signal type)

We can see that the features derived from waveforms (morphological analysis, spectral analysis,...) are more important than the time series features. Taking *only* waveform features gives a higher AUROC score than the reference model, which suggests that the signal features are more informative per feature than the entire model.

5.7 Feature importances II: Sampling rate

In this experiment we evaluate how "valuable" features are depending on the "scale" from which they are computed. We distinguish between the categories:

- MIN: Minute-by-minute frequency: 0.1 Hz or 1/60 Hz
- MED: Medium frequency: 1 Hz
- HIGH: High frequency: 10 or 100 Hz

We get the following results:

Model	AUROC (Sole)	AUROC (Removed)
Full model	0.80	N/A
MIN	0.78 [-0.02]	0.81 [+0.01]
MED	0.78 [-0.02]	0.80 [0.00]
HIGH	0.81 [+0.01]	0.79 [-0.01]

Table 5.17: Hypertension onset forecasting problem with t = 10 minutes (Feature importance by sampling rate)

Among the sole features only the high frequency ones surpass the AUROC score of the reference model. This result confirms the findings of the previous section, as largely, the categories HIGH and SIG overlap. For the removal of features a similar trend is visible: The high-frequency features, when removed from the model decrease the AUROC whereas removing low-frequency does not change or even increase it.

5.8 Feature importances III: Channels

In this experiment we evaluate how valuable individual channels are for the forecasting of ICH.

We categorize the physiological channels into

- ABP: Arterial blood pressures (ABP, ABP Dias, ABP Sys, ABP Mean)
- CER: Cerebral pressures (CPP/ICP)
- HEART: ECG / Heart rate / Pulse rate (II/HR/PULSE/PVC Rate per Minute)
- OXY: Oxygen saturation (PLETH/SpO2)
- RESP: Respiration rate (RESP)

As usual, AUROC scores for sole and removal effects are shown below:

Model	AUROC (Sole)	AUROC (Removed)
Full model	0.80	N/A
ABP	0.58 [-0.22]	0.81 [+0.01]
CER	0.81 [+0.01]	0.62 [-0.18]
OXY	0.53 [-0.27]	0.80 [0.00]
RESP	0.46 [-0.34]	0.80 [0.00]
HEART	0.61 [-0.19]	0.80 [0.00]

Table 5.18: Hypertension onset forecasting problem with t = 10 minutes (Feature importance by physiological channel)

The first important observation is that the removal of features based on channels besides CER (CPP/ICP) does not decrease the AUROC score. Conversely, taking other channels but CER as sole features leads to a AUROC score that is insufficient for precise forecasting (in the range 0.46-0.61). In the future, experiments should be carried out with combinations of features from the most predictive channels (CER/HEART/ABP) to find out if the AUROC score of 0.81 (sole CER) can be improved.

5.9 Feature importances IV: History scale

In this experiment we evalute how valuable it is to consider a longer history of the channels vs. looking at the immediate or mid-term past.

We categorize the history length into

- SHORT: Immediate history (30 seconds - 10 mins)
- MID: Mid-term history (16 mins - 32 mins)
- LONG: Long-term history (64 mins - 256 mins)

The motivation for this experiment is to check how much history should be stored in an ICU forecasting system to enable accurate predictions.

Model	AUROC (Sole)	AUROC (Removed)
Full model	0.80	N/A
SHORT	0.82 [+0.02]	0.79 [-0.01]
MID	0.78 [-0.02]	0.80 [0.00]
LONG	0.78 [-0.02]	0.82 [+0.02]

Table 5.19: Hypertension onset forecasting problem with $t = 10$ minutes (Feature importance by history scale)

The results suggest that the immediate past (30 seconds-10 minutes) is most important (sole AUROC score of 0.82) and MID/LONG are less so. The results for feature removal confirm this trend and even show that removal of the LONG term features leads to an *increased* score.

5.10 Feature importances V: Feature type

Finally, in this experiment, we evaluate how valuable different feature groups are for the forecasting model of ICH onset ($t = 10$). We categorize the features into the following groups:

- PULSE: Morphological pulse features (ABP/ICP signal), QRS latency
- LEVEL: Average energy, Geometric mean, Max, Mean, Median, Min, Norm
- DISPERSION: Coefficient of variation, Variance, Standard deviation, Energy standard deviation
- TREND: Correlation coeff, Regression line slope, Trend between first and last sample
- SPECT: Spectral energy bands, 5 Cepstrum coefficients, Largest 5 FFT frequencies/coefficients, Hjorth mobility / Complexity
- FRACT: Higuchi/Petrosian fractal dimensions
- DIST: Kurtosis, Skewness
- COMPLEX: Shannon entropy, Line length, Approximate entropy, Sample entropy
- PROCESS: Summary measures of long-term memory of time series process: Fisher information / SVD entropy of embedding sequences, Hurst exponent, Detrended fluctuation analysis
- RAW: Minute-by-minute raw values
- CORR: Correlation of channels / Partial directed coherence

Model	AUROC (Sole)	AUROC (Removed)
Full model	0.80	N/A
PULSE	0.72 [-0.08]	0.80 [+0.00]
LEVEL	0.82 [+0.02]	0.78 [-0.02]
DISPERSION	0.69 [-0.11]	0.80 [0.00]
TREND	0.54 [-0.26]	0.80 [0.00]
SPECT	0.72 [-0.08]	0.80 [0.00]
FRACT	0.58 [-0.22]	0.80 [0.00]
DIST	0.46 [-0.34]	0.80 [0.00]
COMPLEX	0.60 [-0.20]	0.80 [0.00]
PROCESS	0.55 [-0.25]	0.80 [0.00]
RAW	0.77 [-0.03]	0.82 [+0.02]
CORR	0.64 [-0.16]	0.80 [0.00]

Table 5.20: Hypertension onset forecasting problem with $t = 10$ minutes (Feature importance by feature type)

It can be seen that sole models using only location information or raw values (of all channels) yield the highest AUROC scores of 0.82 and 0.77 respectively. However, also the morphological features (PULSE) and spectral energies (SPECT) seem to work relatively well on their own (AUROC score of 0.72 in each case). Trends, fractal-, and random process features should not be used alone but augment a model with location features or raw values. Removal of any of the feature types except locations does not have a negative effect on the AUROC score. In the future this result should be validated on other data sets besides MIMIC-II. The increase in AUROC when RAW features are removed could be explained by the high dimensionality of this feature type compared to all others and its high correlation with the LEVEL features.

5.11 Feature selection I: Univariate correlation

In this section we study the predictive utility of statistical summary features using univariate statistical tests. We use the F-test to test the effect of an individual regressor vs. the target variable (regression) in the `mean_icp` problem (horizon $t = 10, 20, 30$ minutes). Predictors and target variables were centered before F-values and p-values were computed.

Below we give an overview of the p-values for each of the ICP mean features computed at different scales of the history of the last 30 seconds up to the last 256 minutes. A low p-value implies that the null hypothesis, that a predictor is not associated with the target, is rejected at high significance.

Feature	10 min	20 min	30 min
1_sig_ICP_0.5_125_1_mean_1	5.78e-52	1.61e-11	8.85e-05
1_sig_ICP_1_10_1_mean_1	4.98e-52	4.22e-11	1.09e-04
1_sig_ICP_2_10_1_mean_1	9.83e-50	6.45e-11	4.78e-05
1_sig_ICP_4_10_1_mean_1	1.81e-40	2.41e-13	2.50e-05
1_ts_ICP_8_1_1_mean_1	1.29e-29	7.43e-24	7.71e-07
1_ts_ICP_16_1_1_mean_1	8.59e-31	9.16e-19	2.95e-06
1_ts_ICP_32_1_1_mean_1	1.95e-27	1.04e-13	3.35e-04
1_ts_ICP_64_0.1_1_mean_1	2.64e-16	1.31e-07	1.51e-03
1_ts_ICP_128_0.1_1_mean_1	7.95e-16	6.48e-16	5.97e-15
1_ts_ICP_256_0.1_1_mean_1	2.14e-02	5.07e-02	4.67e-03

Table 5.21: p-value comparison of mean features computed at different scales for `mean_icp`

It is evident that for univariate forecasting of the ICP mean in t minutes all statistical summaries are significant at the $\alpha = 0.05$ level except the mean of the last 256 minutes for prediction of the ICP mean

in $t = 20$ minutes. This suggests that both short-term, medium-term and long-term histories of the ICP mean are relevant for non-invasive prediction of the current or future ICP mean.

A similar analysis was carried out for different feature types computed on the ICP, ABP and CPP signals:

Feature	10 min	20 min	30 min
1_sig_ICP_0.5_125_1_mean_1	1.26e-03	4.96e-01	3.90e-01
1_sig_ICP_0.5_125_1_entropy_1	2.88e-01	1.43e-03	7.29e-01
1_sig_ICP_0.5_125_1_kurtosis_1	1.07e-01	4.70e-02	1.16e-01
1_sig_ICP_0.5_125_1_linelength_1	5.05e-03	1.18e-01	1.04e-01
1_sig_ICP_0.5_125_1_lrslope_1	3.31e-01	9.62e-01	8.23e-01
1_sig_ICP_0.5_125_1_median_1	6.78e-04	3.57e-01	3.62e-01
1_sig_ICP_0.5_125_1_skewness_1	5.64e-01	7.34e-01	1.33e-01
1_sig_ICP_0.5_125_1_std_1	1.55e-01	2.30e-01	7.17e-01
1_sig_ABP_0.5_125_1_mean_1	1.74e-01	9.04e-02	3.12e-02
1_sig_ABP_0.5_125_1_entropy_1	6.12e-01	1.21e-01	5.42e-01
1_sig_ABP_0.5_125_1_kurtosis_1	3.95e-01	6.12e-02	2.14e-01
1_sig_ABP_0.5_125_1_linelength_1	2.71e-01	2.71e-01	8.39e-01
1_sig_ABP_0.5_125_1_lrslope_1	6.07e-02	7.59e-01	9.70e-01
1_sig_ABP_0.5_125_1_skewness_1	8.92e-01	8.37e-01	2.94e-02
1_sig_ABP_0.5_125_1_std_1	7.35e-02	7.06e-01	3.19e-01
1_ts_CPP_8_1_1_mean_1	3.90e-02	5.37e-01	1.88e-01
1_ts_CPP_8_1_1_entropy_1	4.43e-02	5.55e-02	8.25e-01
1_ts_CPP_8_1_1_skewness_1	2.28e-03	8.94e-04	1.77e-03
1_ts_CPP_8_1_1_std_1	3.44e-02	1.92e-03	5.81e-01

Table 5.22: p-value comparison for different feature types for `mean_icp`

We see that only the ICP mean, the ICP line length, the ICP median, the ABP linear regression trend, the ABP standard deviation and the CPP skewness are significant for an horizon of 10 minutes. For the 20 minute forecasting horizon only the ICP entropy, the ABP mean, the ABP kurtosis and CPP entropy, skewness and standard deviation are significant at the $\alpha = 0.05$ level. For the 30 minute horizon only the CPP skewness is significant.

5.12 Feature selection II: L1-sparsity (Regression)

In this section we try to answer the question: which of the feature columns (among all columns not derived from ICP/CPP channels for the regression problem) is most relevant for the two prediction tasks. We limit ourselves to the $t = 0$ ICP mean regression problem, because the L1-regularizer causes numerical imprecision for all classification losses.

When interpreting the results one should keep in mind one of the unfortunate properties of L1-regularization: When a group of predictors is highly correlated, then L1 feature selection might include an arbitrary subset in the model, while the other equally informative predictors are discarded. This problem could be alleviated with a stability selection with many rounds of L1 feature selection on bootstrap resamples. We will not use this method here because we will shed light on the feature importances using other methods which might then complement the "picture" we get from this experiment.

We set $\alpha = 0.05$ and 106 among 715 features were selected. We categorize the features by the channel they are derived from. In brackets the length of the history, on which the feature was calculated, is given. A $(+)$ means that the feature received a positive coefficient in $\boldsymbol{w}$, whereas $(-)$ means that it received a negative coefficient. As the features are normalized $(+)$ could be interpreted in the following way: An increase over the feature mean is correlated with an increase over the global ICP mean in the data, even when the effect of all other feature columns are accounted for.

ABP signal (125 Hz) (4 selected features)

```
R3: Time ratio between global downstroke time and upstroke time (last 30 seconds) (+)
R4: Time ratio between the dicrotic and systolic heights (last 30 seconds) (+)
Energy in the 12-15 Hz frequency band (last 30 seconds) (+)
4th coefficient of cepstrum (last 30 seconds) (-)
```

For the definition of the R3/R4 features, see [35].

PLETH signal (10/125 Hz) (4 selected features)

```
Standard deviation (last 30 secs) (-)
Standard deviation (last 1 mins) (-)
Standard deviation (last 2 mins) (-)
Standard deviation (last 4 mins) (-)
```

Diastolic ABP time series (1 Hz) (1 selected feature)

```
Shannon entropy (last 32 mins) (-)
```

Systolic ABP time series (1 Hz) (2 selected features)

```
Standard deviation (last 32 mins) (-)
Standard deviation (last 64 mins) (-)
```

MAP time series (0.1/1 Hz) (57 selected features))

```
LAST 256 MINUTES:
Detrended fluctuation analysis coefficient (+)
4th coefficient of cepstrum (+)
5th coefficient of cepstrum (+)
Fisher information of embedding sequences (-)
Hurst exponent (-)
SVD entropy of embedding sequences (+)
-----------------------------------------------------------------------
LAST 128 MINUTES:
Coefficient of variation (-)
Correlation coefficient (+)
Standard deviation over energy in sub-segments (-)
4th coefficient of cepstrum (+)
Linear regression line trend (+)
Trend between first and last sample (+)
Standard deviation (-)
Variance (-)
-----------------------------------------------------------------------
LAST 64 MINUTES
Coefficient of variation (-)
Standard deviation over energy in sub-segments (-)
Linear regression line trend (+)
Trend between first and last sample (+)
Min (+)
Signal/noise ratio (+)
```

```
Standard deviation (-)
Variance (-)
----------------------------------------------------------------------
LAST 32 MINUTES:
Average energy per sample (+)
Coefficient of variation (-)
Standard deviation over energy in sub-segments (-)
2nd coefficient of cepstrum (-)
3rd coefficient of cepstrum (-)
5th coefficient of cepstrum (+)
Higuchi fractal dimension (+)
Hjorth complexity (+)
Linear regression line slope (+)
Trend between first and last sample (+)
Median (+)
Min (+)
Signal/noise ratio (+)
Standard deviation (-)
Variance (-)
----------------------------------------------------------------------
LAST 16 MINUTES:
Average energy per sample (+)
2nd coefficient of cepstrum (-)
3rd coefficient of cepstrum (-)
5th coefficient of cepstrum (+)
Geometric mean (+)
Higuchi fractal dimension (+)
Max (+)
Mean (+)
Median (+)
Euclidean norm (+)
----------------------------------------------------------------------
LAST 8 MINUTES:
Average energy per sample (+)
3th coefficient of cepstrum (-)
5th coefficient of cepstrum (+)
Geometric mean (+)
Higuchi fractal dimension (+)
Linear regression line slope (-)
Max (+)
Mean (+)
Median (+)
Euclidean norm (+)
```

Heart rate time series (0.1/1 Hz) (5 selected features)

```
Standard deviation (last 256 mins) (-)
Standard deviation (last 128 mins) (-)
Shannon entropy (last 32 mins) (-)
Shannon entropy (last 16 mins) (-)
Shannon entropy (last 8 mins) (-)
```

Pulse rate time series (0.1/1 Hz) (3 selected features)

```
Shannon entropy (last 64 mins) (-)
Shannon entropy (last 32 mins) (-)
Shannon entropy (last 16 mins) (-)
```

Respiration rate time series (0.1/1 Hz) (4 selected features)

```
Standard deviation (last 256 mins) (-)
Standard deviation (last 128 mins) (-)
Standard deviation (last 64 mins) (-)
Shannon entropy (last 32 mins) (-)
```

Oxygen saturation time series (0.1 Hz) (3 selected features)

```
Shannon entropy (last 256 mins) (-)
Shannon entropy (last 128 mins) (-)
Shannon entropy (last 64 mins) (-)
```

Minute-by-minute features (from Guiza model)

```
MAP median (last 5 mins) (+)
MAP median (last 10 mins) (+)
MAP median (last 20 mins) (+)
MAP 2nd coefficient of cepstrum (last 240 mins) (+)
MAP 4th coefficient of cepstrum (last 240 mins) (-)
3rd largest FFT coeff frequency (last 240 mins) (+)
MAP Min-min resample (last 17 among 240 mins) (each +)
```

Overall, the majority of features that were selected are based on the MAP channel, with all scales of the history contributing to the set of selected features. Also some of the RAW MAP features were selected. The range of feature types based on MAP is quite broad: Instances of levels, dispersions, fractal dimensions, process and morphological features are all included in the set of selected features. Lastly, it should be noted that for auxiliary channels besides MAP only the standard deviation and entropy were selected (rather than the mean).

5.13 Feature selection III: L2 coefficient weight (Classification)

This section complements the previous section, in that it covers the $t = 10$ hypertension onset forecasting problem and in that it utilizes a weight vector w that is not regularized by the L1- but the L2-penalty. In this experiment all 3812 feature columns could be selected, i.e. ICP and CPP derived features are not excluded any more. We report the 100 features that received the highest absolute value in w^*_{train}. As features were standardized, these could be interpreted as the "importances" of individual features for the logistic regression.

We use the following parameters for the SGD classifier:

```
loss: logistic
penalty: elasticnet
l1_ratio: 0.0 (pure L2 regularizer)
alpha: 0.01
learning_rate: 1/(t+t0) (optimal)
```

The AUROC score (10-fold CV) of the fitted classifier was 0.82.

ICP signal (10/125 Hz) (selected 14 features)

```
LAST 30 SECONDS
Morph: Minimum of averaged pulse: 0.005
Morph: Relative elevation of first sub-peak (DP1): -0.006
Energy contained in the 0-1 Hz band: 0.005
Shannon entropy: -0.006
2nd/3rd/4th coefficient of cepstrum: 0.011/0.008/0.006
Geometric mean: 0.005
Kurtosis: 0.008
Mean: 0.005
Median: 0.005
Skewness: -0.012
-------------------------------------------------------------
LAST 1 MINUTES
Skewness: -0.005
Signal/noise ratio: 0.007
-------------------------------------------------------------
LAST 2 MINUTES
5th coefficient of cepstrum: 0.006
Signal/noise ratio: 0.011
```

ECG (II lead) signal (125 Hz) (selected 1 feature)

```
Latency between QRS complexes (last 30 seconds): -0.006
```

MAP time series (0.1/1 Hz) (selected 2 features)

```
Petrosian fractal dimension (last 256 minutes): 0.006
Correlation coefficient (last 8 minutes): 0.005
```

CPP time series (1 Hz) (selected 8 features)

```
LAST 128 MINUTES
Signal/noise ratio: 0.005
---------------------------------------------------------------
LAST 32 MINUTES
3rd coefficient of cepstrum: -0.006
Kurtosis: 0.005
---------------------------------------------------------------
LAST 16 MINUTES
3rd coefficient of cepstrum: -0.007
Higuchi fractal dimension: 0.005
Kurtosis: 0.005
Skewness: 0.005
---------------------------------------------------------------
LAST 8 MINUTES
4th coefficient of cepstrum: 0.006
```

Heart rate time series (0.1/1 Hz) (selected 6 features)

```
Mean (last 256 mins): 0.006
Mean (last 128 mins): 0.006
Mean (last 64 mins): 0.006
```

```
Mean (last 32 mins): 0.006
Mean (last 16 mins): 0.005
Mean (last 8 mins): 0.005
```

ICP time series (0.1/1 Hz) (selected 23 features)

```
LAST 128 MINUTES
Average energy per sample: 0.006
Mean: 0.005
Trend between first and last sample: 0.005
Median: 0.005
Signal/noise ratio: 0.006
------------------------------------------------------------
LAST 64 MINUTES
Average energy per sample: 0.006
Mean: 0.005
Median: 0.006
Euclidean norm: 0.005
------------------------------------------------------------
LAST 32 MINUTES
Average energy per sample: 0.005
4th coefficient of cepstrum: 0.005
Mean: 0.005
Median: 0.007
Euclidean norm: 0.005
------------------------------------------------------------
LAST 16 MINUTES
Shannon entropy: 0.006
Max: 0.005
Median: 0.006
Euclidean norm: 0.005
Standard deviation: 0.005
------------------------------------------------------------
LAST 8 MINUTES
Correlation coefficient: 0.007
Shannon entropy: 0.006
Linear regression line slope: 0.006
Trend between first and last sample: 0.007
```

Pulse rate time series (0.1/1 Hz) (selected 6 features)

```
Mean (last 256 mins): 0.008
Mean (last 128 mins): 0.006
Mean (last 64 mins): 0.006
Mean (last 32 mins): 0.006
Mean (last 16 mins): 0.006
Mean (last 8 mins): 0.006
```

Oxygen saturation time series (0.1 Hz) (selected 1 feature)

```
Shannon entropy (last 128 mins): -0.005
```

Multi-time series (selected 4 features)

```
Correlation coeff. of CPP/ICP (last 16 mins): -0.008
Correlation coeff. of CPP/MAP (last 16 mins): -0.007
2nd coefficient of Partial Directed Coherence of ICP/MAP/CPP: 0.005
3rd coefficient of Partial Directed Coherence of ICP/MAP/CPP: 0.005
```

Minute-by-minute features from Guiza model (selected 34 features)

```
2nd largest MAP FFT coefficient frequency (last 240 mins): 0.007
2nd largest CPP FFT coefficient frequency (last 240 mins): 0.007
4th largest ICP FFT coefficient frequency (last 240 mins): 0.005
ICP standard deviation (last 10 mins): 0.005
ICP median (last 20 mins): 0.006
ICP standard deviation (last 20 mins): 0.006
Subset of ICP min-min resample (last 240 mins): 28 in range 0.005-0.008 each
```

5.14 Non-linear methods I: Kernel SVMs

Hypertensive onset forecasting problem ($t = 10$)

In this experiment we evaluate whether non-linear methods can achieve a better generalization performance than functions linear in the feature space. We use the patient-stratified 10-fold randomized validation.

We have used a kernelized SVM [39] with the following parameters (the model has not been fully tuned, but the default parameters have been found to work well).

```
kernel: rbf (Gaussian kernel)
gamma: 0.00025 (~ 1/3800)
```

Instances are weighted in the objective function inversely proportional to their classes' frequency in the training set. Note that we are facing a high-dimensional problem with $d > n$ so we expect that the decision surface needs to be smoothed considerably by setting a small C. We get the following results for the $t = 10$ minute hypertensive onset forecasting problem:

C	**AUROC (macro-average)**
0.01	0.20
0.1	0.62
1.0	0.81
10.0	0.78

Table 5.23: Hypertension onset forecasting problem with $t = 10$ minutes (Patient stratified, Kernel SVM)

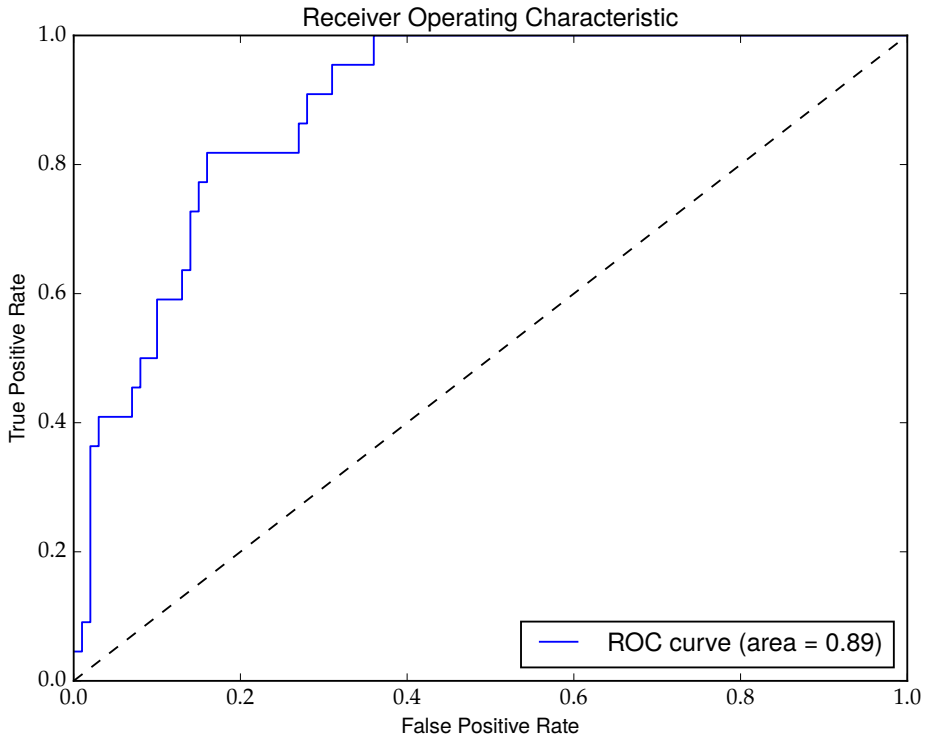


Figure 5.8: Hypertension onset forecasting problem with t = 10 minutes (Patient stratified, Kernel SVM)

The non-linear Kernel SVM achieves slightly better AUROC scores than the linear method (as evaluated in Section 5.1), even though the problem is high-dimensional, is over-parameterized, and a Gaussian kernel that is prone to overfitting was used. The ROC curve shows that an operating point with sensitivity of 0.82 and specificity of 0.81 is available.

Non-invasive ICP mean estimation problem

We have also checked whether the performance of the regression can be improved by using kernel SVMs. We use an epsilon-insensitive SVM regressor with the following parameters:

```
epsilon: 0.1
kernel: rbf (Gaussian kernel)
gamma: 0.0007 (~ 1/1430)
```

The results are shown below and should be compared to the ones obtained in Section 5.5:

C	**Mean absolute error**	**Median absolute error**
0.001	3.89	3.47
0.01	3.89	3.46
0.1	3.90	3.45
1.0	4.19	3.73

Table 5.24: Non-invasive ICP mean estimation problem with t = 0 minutes (Patient stratified, Kernel SVM)

The mean absolute error is significantly reduced by the introduction of a kernel method. Almost a 50 % reduction from the SGD score (6.95) can be observed.

5.15 Non-linear methods II: Ensemble of extremely randomized trees

Hypertensive onset forecasting problem ($t = 10$)

To compare our model against the recently proposed [12] model that includes ICP morphological features, we have fitted our features with the same ensemble method. We are using the following parameters for an ensemble of extremely randomized decision trees [40]. Not all of its parameters have been fully tuned, only the ensemble size has been increased until generalization performance no longer improves.

```
Number of trees: 500
Maximum number of features considered per split: 60
Unlimited tree depth
Splitting criterion: Gini
Min. samples for split: 2
```

We use the patient-stratified cross-validation scheme.

AUROC (macro-average)
0.81

Table 5.25: Hypertension onset forecasting problem with $t = 10$ minutes (Patient stratified, Ensemble of extremely randomized trees)

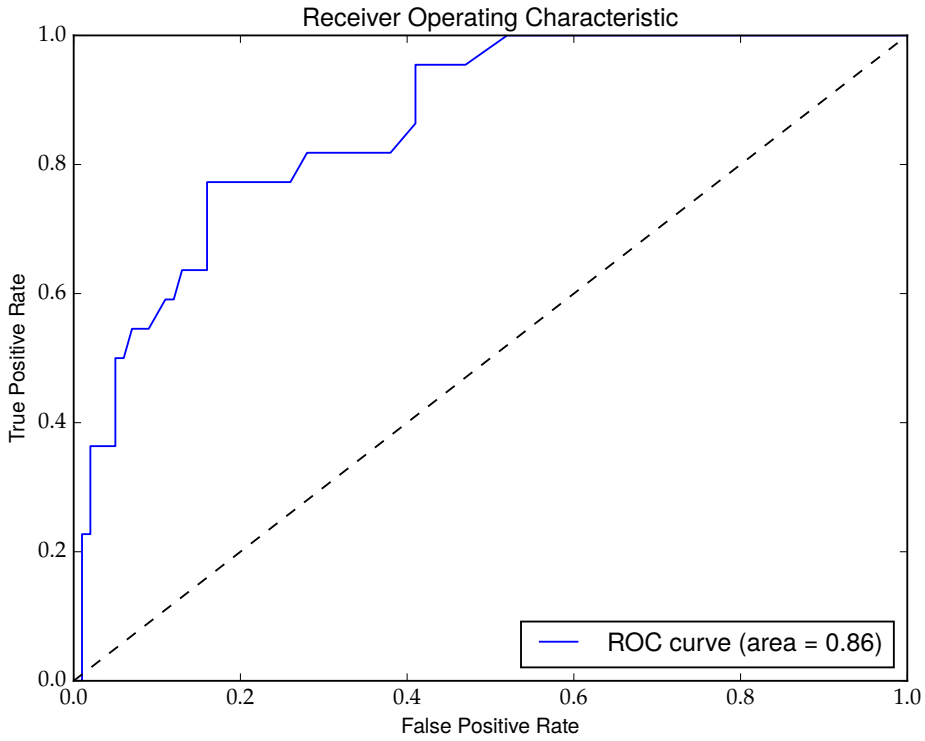


Figure 5.9: Hypertension onset forecasting problem with $t = 10$ minutes (Patient stratified, Ensemble of extremely randomized trees)

We can recover the same AUROC score as for the non-linear kernel SVM. The performance of the two methods is comparable on the MIMIC-II data-set, with sensitivity and specificity at around 80 %.

Non-invasive ICP mean estimation problem

In the next experiment, we evaluate the generalization ability of the extremely randomized tree ensemble for the ICP mean estimation problem. We have now used the following parameters:

```
Number of trees: 500
Maximum number of features considered per split: 1430 (all features)
Unlimited tree depth
Splitting criterion: MSE (Mean-squared error)
Min. samples for split: 2
```

Mean absolute error	Median absolute error
4·44	3·74

Table 5.26: Non-invasive ICP mean estimation problem with $t = 0$ minutes (Patient stratified, Ensemble of extremely randomized trees)

The mean absolute error is higher than the one obtained for the linear SGD method (3.84) and the one obtained for the non-linear kernel method (3.89).

5.16 SGD tuning parameters

In this section we evaluate the dependence of generalization performance on the free parameters of the Stochastic Gradient Descent method implemented in `SciKitLearn` [18]. We are using sequential tuning parameter selection: We fix an ordering of the parameters, and then optimize the parameters in turn with respect to the fixed optimal parameters of the preceding steps. This is not an exhaustive exploration of product space of possible parameters, but should be good enough for our purposes. We will optimize the parameters for the `onset_icp` and the `mean_icp` problem in turn:

Non-invasive ICP mean estimation

First we look at the non-invasive ICP mean estimation problem.

Loss function

For this experiment we fix a minimal set of parameters of the SGD regression model:

```
penalty: L2 (ridge regression)
alpha: 0.0001 (regularization parameter)
learning_rate: eta0/pow(t,power_t)
```

where `power_t`= 0.25 and `eta0`= 0.01. We vary the loss function in the range (squared_loss, huber, epsilon_insensitive, squared_epsilon_insensitive). For the last two losses the sensitivity threshold is set to $\epsilon = 0.1$.

Loss function	Mean absolute error
squared_loss	641549.22 (divergence)
huber	7.26
epsilon_insensitive	3.87
squared_epsilon_insensitive	566016059723.48 (divergence)

Table 5.27: Non-invasive ICP mean estimation problem with $t = 0$ minutes (by loss function)

Note that the optimization problems associated with the squared losses have diverged, probably because of the wrong setting of the learning rate. After this experiment, we have decided to use the ϵ-insensitive loss function based on its superior performance compared to the Huber loss.

Regularization penalty

At this stage, the following parameters of the SGD regression model were fixed:

```
loss: epsilon_insensitive
penalty: L2 (ridge regression)
learning_rate: eta0/pow(t,power_t)
eta0: 0.01
power_t=0.25
```

We then vary the regularization parameter α in the range $(0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0)$ and find the one that achieves the lowest mean absolute error:

α	**Mean absolute error**
0.00001	3.87
0.0001	3.87
0.001	3.85
0.01	3.86
0.1	3.96
1.0	4.77

Table 5.28: Non-invasive ICP mean estimation problem with $t = 0$ minutes (by α)

Based on this result we have chosen the regularization parameter as $\alpha = 0.001$.

Elastic net mixing parameter

We now move from the L2-loss to its generalization, the elastic net [41] which is a convex combination of the L1 and the L2 regularizer. There is a new free parameter `l1_ratio` which determines the mixing between L1 and L2 regularizer. For this experiment we fix the parameters of the SGD regression model to:

```
loss: epsilon_insensitive
alpha: 0.001
learning_rate: eta0/pow(t,power_t)
eta0: 0.01
power_t=0.25
```

We vary the `l1_ratio` in the range $(0, 0.2, 0.4, 0.6, 0.8, 1.0)$.

l1_ratio	**Mean absolute error**
0.0	3.85
0.2	3.84
0.4	3.88
0.6	3.86
0.8	3.86
1.0	3.85

Table 5.29: Non-invasive ICP mean estimation problem with $t = 0$ minutes (by elastic net mixing parameter)

The results show that the impact of `l1_ratio` is minimal. We have fixed it to $= 0.2$ at this stage.

Learning rate schedule

Now we will look at the free parameters of the SGD algorithm. The most important one is the learning schedule, which determines how the coefficient vector $\boldsymbol{w}$ is updated in each fitting step. We fix the parameters of the SGD regression model to:

```
loss: epsilon_insensitive
alpha: 0.001
l1_ratio: 0.2
eta0: 0.01
power_t=0.25
```

We then vary learning rate between the three schedules

- `constant`: η_0
- `optimal`: $1.0/(\alpha t)$
- `invscaling`: $\eta_0/t^{\text{power}_t}$

Learning rate schedule	Mean absolute error
`constant`	5.46
`optimal`	490.84
`invscaling`	3.84

Table 5.30: Non-invasive ICP mean estimation problem with $t = 0$ minutes (by learning rate schedule)

We will only consider the inverse scaling learning schedule as it performs best and is the recommended schedule for regression problems solved with SGD.

Initial learning rate

In the next experiment, we determine the optimal parameters of the inverse scaling learning rate schedule. We first optimize η_0, the initial learning rate. The parameters of the SGD regression model were fixed to:

```
loss: epsilon_insensitive
alpha: 0.001
l1_ratio: 0.2
learning_rate: eta0/pow(t,power_t)
power_t=0.25
```

and η_0 was varied in the range $(1.0, 0.1, 0.01, 0.001, 0.0001)$.

η_0	Mean absolute error
1.0	79.52
0.1	8.99
0.01	3.84
0.001	7.26
0.0001	9.53

Table 5.31: Non-invasive ICP mean estimation problem with $t = 0$ minutes (by initial learning rate)

Based on this result we are fixing the initial learning rate to the optimal value $\eta_0^* = 0.01$.

Learning rate decay

The last parameter to be set is the learning rate decay. We fix all other parameters of the SGD regression model to:

```
loss: epsilon_insensitive
alpha: 0.001
l1_ratio: 0.2
learning_rate: eta0/pow(t,power_t)
eta0: 0.01
```

and vary the decay power_t in the range $(0.1, 0.25, 0.5, 0.75, 1.0)$.

power_t	Mean absolute error
0.1	4.27
0.25	3.84
0.5	5.78
0.75	7.23
1.0	8.28

Table 5.32: Non-invasive ICP mean estimation problem with $t = 0$ minutes (by learning rate decay)

As the last step of our sequential tuning parameter selection we fix the learning rate decay to $\text{power}_t^* = 0.25$. The fully tuned model that we have obtained was then used for all experiments in Section 5.5.

Forecasting of hypertension onset

The same sequential tuning protocol was followed for the forecasting problem with $t = 10$ minutes. First we determine the loss function:

Loss function

The initial parameters of the untuned SGD classification model were set to:

```
penalty: L2 (SVM)
alpha: 0.0001 (regularization parameter)
learning_rate: 1/(t+t0)
```

The loss function was then varied in the range (log, modified_huber). Other loss functions were available, but they were not considered, because they do not support the output of positive class probabilities.

Loss function	AUROC score
log	0.68
modified_huber	0.68

Table 5.33: Hypertension onset forecasting problem with $t = 10$ minutes (by loss function)

As the results are inconclusive we have picked the logistic loss. Our motivation was that it is the most prominent and well-understood loss function for binary classification.

Regularization penalty

Next we have optimized the regularization parameter α with respect to the fixed logistic loss function. The parameters of the SGD classification model were fixed to:

```
loss: log
penalty: L2 (linear SVM)
learning_rate: 1/(t+t0)
```

and the regularization penalty α was varied in the range $(0.0001, 0.001, 0.01, 0.1, 1.0, 10.0)$.

α	**AUROC score**
0.00001	0.68
0.0001	0.68
0.001	0.66
0.01	0.68
0.1	0.73
1.0	0.80
10.0	0.83

Table 5.34: Hypertension onset forecasting problem with t = 10 minutes (by α)

We can observe that strong regularization is needed and $\alpha^* = 10.0$ is the optimal value for our data-set.

Elastic net mixing parameter

As in the previous section, we move to the elastic net regularizer and determine the optimal mixing parameter `l1_ratio`. The parameters of the SGD classification model were fixed to:

```
loss: log
penalty: elasticnet
learning_rate: 1/(t+t0)
alpha: 10.0
```

As before, we vary the elastic net mixing parameter `l1_ratio` in the range $(0, 0.2, 0.4, 0.6, 0.8, 1.0)$.

l1_ratio	**AUROC score**
0.0	0.83
0.2	0.50
0.4	0.50
0.6	0.50
0.8	0.50
1.0	0.50

Table 5.35: Hypertension onset forecasting problem with t = 10 minutes (by elastic net mixing parameter)

The results show that introducing L1-regularization via `l1_ratio` $\geqslant 0.2$ leads to optimization divergence and renders the model useless. The only suitable choice is a fully L2-regularized SGD classifier with `l1_ratio`$= 0.0$.

Learning rate schedule

Having fixed loss function and regularizer, we determine the optimal learning rate schedule for the problem. The parameters of the SGD classication model are fixed to:

```
loss: log
penalty: l2
alpha: 10.0
eta_0: 0.01
```

and the learning rate schedule was varied between the 3 choices:

- **constant**: $\eta_0 = 0.01$
- **invscaling**: $\eta_0 / t^{\text{power}_t}$ with $\eta_0 = 0.01$ and $\text{power}_t = 0.5$
- **optimal**: $1/(t + t_0)$ with t_0 set by a heuristic proposed by Bottou.

Learning rate schedule	AUROC score
constant	0.57
optimal	0.83
invscaling	0.80

Table 5.36: Hypertension onset forecasting problem with $t = 10$ minutes (by learning rate schedule)

As the last step in the sequential tuning parameter search we fix the learning rate schedule to `optimal`. Notably also the `invscaling` schedule performs well, although it has been designed for regression problems.

5.17 Encoding of missing values

In this experiment we check whether the model performance is influenced by the "missing-value column indicator" technique explained in Section 4.15. For this we are training the SGD classifier (with the best parameters obtained before) on the hypertensive onset forecasting problem with $t = 10$ (CV validation stratified according to patient identity). The SGD classifier parameters are:

```
loss: logistic
penalty: L2
alpha: 10.0
learning_rate: 1/(t+t0) (optimal)
```

Without the indicator columns $d = 1906$ and missing values are simply imputed by the mean of the respective feature column. As data are standardized, this corresponds to substituting 0.0 for missing values. The full model (including indicator columns) has $d = 3812$ features. The comparison gave the following result:

With indicator columns?	AUROC (macro-average)
No	0.80
Yes	0.80

Table 5.37: Hypertension onset forecasting problem with $t = 10$ minutes (indicator column inclusion)

We can see that the indicator column technique does not improve the classification performance. In other words: we may simply replace missing values by their mean. A related question should be explored in further experiments: What is the overall impact of missing values and how much do we gain or lose by excluding samples that contain any missing features.

Chapter 6

Conclusion

6.1 Summary of findings

In this work we have delivered two prediction models for:

- Forecasting of intracranial hypertension (ICH) onset in t minutes ($t \in \{5, 10, 20\}$)
- Non-invasive estimation of current intracranial pressure (ICP) mean

We have validated the generalization performance of our models for the two tasks and conducted a series of experiments to illuminate the contribution of features and statistical learning models to the ultimate performance of our methods. These findings could contribute to building more robust and economic forecasting systems for the intensive care unit (ICU).

Our main findings and insights gained during construction of the feature generation framework and experiments are summarized below:

Experimental findings

Our models for the ICH forecasting problem achieve Area under the ROC curve (AUROC) scores in the range 0.80-0.85 (horizon of up to 20 minutes) and balanced operating points of sensitivity/specificity of around 80 %. Thus, their discrimination performance is consistent with previously reported models and sufficient to be used in the ICU to extend the time period to prepare interventions. We think, initially, they should be used to supplement and not replace human judgement.

Our classification models generalize to other patients as well as to other parts of the history. Usually the AUROC is only 0.01-0.03 points lower when patients in the test set are unseen during training.

The AUROC scores of our classification models degrade gracefully as the forecasting horizon is increased from 5 to 20 minutes. For patient-stratified validation they decrease from 0.83 (5 minutes) to 0.81 (20 minutes). Further experiments are needed to evaluate whether increases of the horizon beyond 20 minutes are feasible.

Comparisons to implementations of previously reported models by Guiza et al. and Hu et al. *on the MIMIC-II data-set* suggest that our method (which can be seen as a super-set of both of these models) performs better with an AUROC score of 0.81 compared to 0.78 and 0.74, respectively.

We have tested out-of-data-set generalization (transfer learning) of our ICH forecasting model and found that AUROC scores are reduced to 0.78. This suggests that our model does not "break down" as it encounters data with different quality, retrieval conditions and applied pre-processing.

Our models for the non-invasive estimation of the ICP mean perform somewhat better than dummy mean regressors (4.29 mmHg), with a mean absolute error of 3.84 mmHg. This improvement should not be discounted and might be relevant in clinical practice. We believe that there is room for improvement by exploring features beyond the ones that we have considered.

Evaluation of feature importances has established that none of the feature types except levels (mean,median,max,min) are essential to building a highly discriminative forecasting model for ICH. In the experiments we have also found that

1. Features based on 125 Hz sampled signals are more essential than time series or minute-by-minute features. Hence, high-frequency monitors should be integrated into an ICU forecasting system, whenever possible.
2. When faced with a choice of which channels to record/store for more accurate ICH forecasting we recommend that only cerebral signals (ICP and Cerebral perfusion pressure) should be stored. Inclusion of Arterial blood pressure (ABP), oxygen saturation, respiration- and heart rates did not improve the predictive performance of our models.
3. Features calculated on the very recent history (30 seconds-10 minutes) are most informative for ICH forecasting. There is no need to spend storage resources on storing up to 1-4 hours of history, as the resulting features might be uninformative and even "confuse" the classifier.
4. Features that are particularly informative for ICH forecasting are: Levels, morphological features on ICP/ABP pulses, Spectral energies and raw values. However it should be kept in mind that less can be more because features might be highly correlated. We recommend picking a combination of these features by an exhaustive subset selection using randomized fold validation.
5. Feature selection experiments using L1/L2 regularization indicate that individual features of certain feature types are relevant even though the corresponding "block" might not be informative on the whole. We recommend that these features are identified using L1 stability selection and then added to the features mentioned in the previous item.

The indicator column technique for encoding missing values is not necessary to achieve good performance when mean imputing missing values. We suspect its positive effect is counter-balanced by the fact that the columns increase the size of feature matrix by 2 and spawn many uninformative predictor variables.

We recommend that free parameters of stochastic gradient descent are carefully tuned to the properties of a data-set or the context in which the forecasting system should be deployed. The most relevant parameters are α and the loss function.

General insights

In forecasting of ICH, we are faced with a highly imbalanced binary classification problem. It is key to success that both the machine learning methods (via introducing a custom loss function that reflects our preferred tradeoff between sensitivity and false alarm rate) and the evaluation methods reflect this.

It is desirable to sub-sample the negative instances to improve the fitting speed and rectify the highly imbalanced label distribution somewhat.

In our data set of $\approx$ 170000 instances, there were 86 cases of intracranial hypertension onset. Our experience suggests that time should be spent to experiment with the definition of the prediction target such that a certain minimum number of positive instances are available. Of course it would be even more desirable to collect data in a targeted way, such that we can apply stricter definitions for ICH and still collect a suitable number of positive cases to allow for robust fitting of the classifier/evaluation.

Missing values are very common in physiological data and it is important to deal with them in a systematic manner: We have used linear interpolation and filtering methods to smooth the signal and allow for estimation of these missing values.

Feature generation for physiological signals could benefit from sophisticated methods to deal with missing values in generalized linear models / SGD. In our work we have used mean imputation combined with indicator columns for missing values, but we suspect that this method has its limitations. Input from and joint work with the theoretical pattern recognition community could lead to much improved solutions for this problem.

The available data-set (in our case MIMIC-II) represents the main bottleneck of the feature generation as it can "introduce" arbitrary data quality issues and missing values that the downstream feature

generation can often not fix anymore. It is of prime importance to combine feature generation with a sophisticated signal processing architecture that improves the signal-to-noise ratio as signals are collected. In our work we attempted this to a certain extent using linear interpolation and low-pass filtering.

The software architecture that emerged during construction of the Python feature generation application is very general and could be applied to the prediction of arbitrary target values (either existence of events like seizures, or continuous values) based on features defined on histories of physiological signals: The essential ingredients of the architecture are the pre-processing pipelines that clean each channel in a specialized way and the value history that buffers intermediate values to avoid re-computation during feature generation.

Behind the finalized set of features there is a complex dependency graph that connects intermediate values (like Fourier transform coefficients, or location of detected pulses) with the final features (which can be imagined as leaves of this dependency graph). It is important that the construction of features traverses this graph and stores any computed value in a cache, to minimize the time spent on feature construction.

The statistical learning model has secondary importance for the performance of an ICH forecasting system. For example, in our experiments, we have found that non-linear kernel support vector machines, ensembles of randomized trees and "workhorse" generalized linear models fitted with SGD actually lead to similar AUROC scores. With this in mind, we recommend that the most flexible and fastest (in terms of fitting and prediction speed) model is chosen.

This work represents a stepping stone towards the development of intelligent forecasting systems deployed in the ICU. Let us finally discuss some possible extensions towards this ultimate goal.

6.2 Further work

Data cleaning

Currently, the pre-processing of the data is simplistic. If a data point is missing and sufficiently many data points are available in its neighbourhood it is filled in using linear interpolation. If there are too few valid data points in the multi-scale history then its dependent features are not explicitly computed but filled in with the sentinel value `NAN` (Not-a-number). To avoid "losing" too many data points in downstream machine learning model fitting we use an indicator column technique, such that missing features are imputed by the mean of the respective feature column.

One weakness of the linear interpolation approach is that the synthesized signal appears unnatural and contains no high frequency components. Two plausible approaches would be to fill in gaps in the signal with previously seen ICP segments, possibly pertubed by Gaussian white noise. On the other hand, for time series a simple auto-regressive moving average model could be used to fill in missing values.

Potentially, the missing value estimation could even be done by using our own models to predict channels that are missing and fill them in automatically.

Evaluation on Traumatic brain injury (TBI) specific data-sets

Our methods should be evaluated on data-sets which only contain patients with the TBI condition (or at least with more episodes of ICH). One limitation of the MIMIC-II data-set is that it includes patients with a varying set of conditions and some but not all have ICH episodes. It would also be desirable to test the method on data-sets which contain additional channels like Central Venous Pressure with fewer missing values than the MIMIC-II data-set. Ideally, new data should be collected in an ICU with the goal of testing the ICH forecasting system already in mind.

Specialization of feature generation framework to alternative tasks

As we have noted, the software architecture of our framework generalizes to other conditions. We hope that its principles could be exploited for the forecasting of epileptic seizures, intracranial hypotension or other physiological events.

Integration into online signal analysis architecture of an ICU

Our work can be interpreted as a proof-of-concept that a collection of features constructed from physiological channels (ICP,ABP,...) combined with SGD learning models yields discriminative classification. In practice, however, our framework should be deployed as part of a larger joint hardware-software architecture where data samples are not read from text files but come streaming in from sensors. The speed of feature generation ($\approx$ 1s per 30s window) is fast enough to allow real-time feature generation with a new data point emitted roughly every second.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

FORECASTING INTRACRANIAL HYPERTENSION USING TIME SERIES AND WAVEFORM FEATURES

Authored by (in block letters):

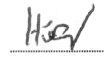
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
HÜSER	MATTHIAS

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 9th April 2015	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Bibliography

[1] S. Neff and R. Subramaniam, "Monro-kellie doctrine.," *Journal of neurosurgery*, vol. 85, no. 6, pp. 1195–1195, 1996.

[2] J.-Y. Fan, C. Kirkness, P. Vicini, R. Burr, and P. Mitchell, "Intracranial pressure waveform morphology and intracranial adaptive capacity," *American Journal of Critical Care*, vol. 17, no. 6, pp. 545–554, 2008.

[3] J. D. Miller, D. P. Becker, J. D. Ward, H. G. Sullivan, W. E. Adams, and M. J. Rosner, "Significance of intracranial hypertension in severe head injury," *Journal of Neurosurgery*, vol. 47, no. 4, pp. 503–516, 1977.

[4] S. Badri, J. Chen, J. Barber, N. R. Temkin, S. S. Dikmen, R. M. Chesnut, S. Deem, N. D. Yanez, and M. M. Treggiari, "Mortality and long-term functional outcome associated with intracranial pressure after traumatic brain injury," *Intensive care medicine*, vol. 38, no. 11, pp. 1800–1809, 2012.

[5] N. Juul, G. F. Morris, S. B. Marshall, and L. F. Marshall, "Intracranial hypertension and cerebral perfusion pressure: influence on neurological deterioration and outcome in severe head injury*," *Journal of neurosurgery*, vol. 92, no. 1, pp. 1–6, 2000.

[6] A. Bhatia and A. K. Gupta, "Neuromonitoring in the intensive care unit. i. intracranial pressure and cerebral blood flow monitoring," *Intensive care medicine*, vol. 33, no. 7, pp. 1263–1271, 2007.

[7] P. K. Eide, "A new method for processing of continuous intracranial pressure signals," *Medical engineering & physics*, vol. 28, no. 6, pp. 579–587, 2006.

[8] R. Sahjpaul and M. Girotti, "Intracranial pressure monitoring in severe traumatic brain injury–results of a canadian survey.," *The Canadian journal of neurological sciences. Le journal canadien des sciences neurologiques*, vol. 27, no. 2, pp. 143–147, 2000.

[9] R. Hamilton, P. Xu, S. Asgari, M. Kasprowicz, P. Vespa, M. Bergsneider, and X. Hu, "Forecasting intracranial pressure elevation using pulse waveform morphology," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 4331–4334, IEEE, 2009.

[10] X. Hu, P. Xu, F. Scalzo, P. Vespa, and M. Bergsneider, "Morphological clustering and analysis of continuous intracranial pressure," *Biomedical Engineering, IEEE Transactions on*, vol. 56, no. 3, pp. 696–705, 2009.

[11] X. Hu, P. Xu, S. Asgari, P. Vespa, and M. Bergsneider, "Forecasting icp elevation based on prescient changes of intracranial pressure waveform morphology," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 5, pp. 1070–1078, 2010.

[12] F. Scalzo, R. Hamilton, S. Asgari, S. Kim, and X. Hu, "Intracranial hypertension prediction using extremely randomized decision trees," *Medical engineering & physics*, vol. 34, no. 8, pp. 1058–1065, 2012.

[13] F. Güiza, B. Depreitere, I. Piper, G. Van den Berghe, and G. Meyfroidt, "Novel methods to predict increased intracranial pressure during intensive care and long-term neurologic outcome after traumatic brain injury: Development and validation in a multicenter dataset*," *Critical care medicine*, vol. 41, no. 2, pp. 554–564, 2013.

[14] F. Zhang, M. Feng, S. J. Pan, L. Y. Loy, W. Guo, Z. Zhang, P. L. Chin, C. Guan, N. K. K. King, and B. T. Ang, "Artificial neural network based intracranial pressure mean forecast algorithm for medical decision support," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pp. 7111–7114, IEEE, 2011.

[15] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark, "Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database," *Critical care medicine*, vol. 39, no. 5, p. 952, 2011.

[16] G. B. Moody, R. G. Mark, and A. L. Goldberger, "Physionet: a web-based resource for the study of physiologic signals," *IEEE Eng Med Biol Mag*, vol. 20, no. 3, pp. 70–75, 2001.

[17] I. Piper, G. Citerio, I. Chambers, C. Contant, P. Enblad, H. Fiddes, T. Howells, K. Kiening, P. Nilsson, and Y. Yau, "The brainit group: concept and core dataset definition," *Acta neurochirurgica*, vol. 145, no. 8, pp. 615–629, 2003.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] F. S. Bao, X. Liu, and C. Zhang, "Pyeeg: an open source python module for eeg/meg feature extraction," *Computational intelligence and neuroscience*, vol. 2011, 2011.

[20] R. Esteller, J. Echauz, T. Tcheng, B. Litt, and B. Pless, "Line length: an efficient feature for seizure onset detection," in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, vol. 2, pp. 1707–1710, IEEE, 2001.

[21] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 278, no. 6, pp. H2039–H2049, 2000.

[22] S. M. Pincus, "Approximate entropy as a measure of system complexity.," *Proceedings of the National Academy of Sciences*, vol. 88, no. 6, pp. 2297–2301, 1991.

[23] A. Petrosian, "Kolmogorov complexity of finite sequences and recognition of different preictal eeg patterns," in *Computer-Based Medical Systems, 1995., Proceedings of the Eighth IEEE Symposium on*, pp. 212–217, IEEE, 1995.

[24] T. Higuchi, "Approach to an irregular time series on the basis of the fractal theory," *Physica D: Nonlinear Phenomena*, vol. 31, no. 2, pp. 277–283, 1988.

[25] S. J. Roberts, W. Penny, and I. Rezek, "Temporal and spatial complexity measures for electroencephalogram based brain-computer interfacing," *Medical & biological engineering & computing*, vol. 37, no. 1, pp. 93–98, 1999.

[26] C. J. James and D. Lowe, "Extracting multisource brain activity from a single electromagnetic channel," *Artificial Intelligence in Medicine*, vol. 28, no. 1, pp. 89–104, 2003.

[27] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of dna nucleotides," *Physical Review E*, vol. 49, no. 2, p. 1685, 1994.

[28] H. E. Hurst, R. P. Black, and Y. Simaika, *Long-term storage: an experimental study*. Constable, 1965.

[29] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[30] B. Hjorth, "Eeg analysis based on time domain properties," *Electroencephalography and clinical neurophysiology*, vol. 29, no. 3, pp. 306–310, 1970.

[31] X. Hu, T. Glenn, F. Scalzo, M. Bergsneider, C. Sarkiss, N. Martin, and P. Vespa, "Intracranial pressure pulse morphological features improved detection of decreased cerebral blood flow," *Physiological measurement*, vol. 31, no. 5, p. 679, 2010.

[32] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *Biomedical Engineering, IEEE Transactions on*, no. 3, pp. 230–236, 1985.

[33] X. Hu, P. Xu, D. J. Lee, P. Vespa, K. Baldwin, and M. Bergsneider, "An algorithm for extracting intracranial pressure latency relative to electrocardiogram r wave," *Physiological measurement*, vol. 29, no. 4, p. 459, 2008.

[34] W. Zong, T. Heldt, G. Moody, and R. Mark, "An open-source algorithm to detect onset of arterial blood pressure pulses," in *Computers in Cardiology, 2003*, pp. 259–262, IEEE, 2003.

[35] V. G. Almeida, J. Vieira, P. Santos, T. Pereira, H. Pereira, C. Correia, M. Pego, and J. Cardoso, "Machine learning techniques for arterial pressure waveform analysis," *Journal of Personalized Medicine*, vol. 3, no. 2, pp. 82–101, 2013.

[36] L. A. Baccalá and K. Sameshima, "Partial directed coherence: a new concept in neural structure determination," *Biological cybernetics*, vol. 84, no. 6, pp. 463–474, 2001.

[37] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[38] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[39] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171–1220, 2008.

[40] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[41] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.