# Gaussian Process-based Feature Selection for Wavelet Parameters: Predicting Acute Hypotensive Episodes from Physiological Signals

Franck Dernoncourt, Kalyan Veeramachaneni, Una-May O'Reilly
*MIT CSAIL ALFA*
*Cambridge, MA, USA*
*francky@mit.edu, kalyan@csail.mit.edu, unamay@csail.mit.edu*

***Abstract*—Physiological signals such as blood pressure might contain key information to predict a medical condition, but are challenging to mine. Wavelets possess the ability to unveil location-specific features within signals but there exists no principled method to choose the optimal scales and time shifts. We present a scalable, robust system to find the best wavelet parameters using Gaussian processes (GPs). We demonstrate our system by assessing wavelets as predictors for the occurrence of acute hypotensive episodes (AHEs) using over 1 billion blood pressure beats. We obtain an AUROC of 0.79 with wavelet features only, and the false positive rate when the true positive rate is fixed at 0.90 is reduced by 14% when the wavelet feature is used in conjunction with other statistical features. Furthermore, the use of GPs reduces the selection effort by a factor of 3 compared with a naive grid search.**

## I. Introduction and motivation

Over the past few years the amount of collected medical data, especially physiological signals, has soared at an overwhelming rate. Physiological signals are remarkable in that they are typically very noisy, complex and unstructured: using large data sets might help overcome those challenging characteristics and obtain more generalizable results. It is therefore critical to adopt data mining approaches that can scale up. Notwithstanding, prediction studies on physiological signals often use a modest amount of data.

Wavelets possess the ability to unveil location-specific features within the signal [1]. They are therefore interesting predictor candidates to investigate, but there exists no principled method to set their scale and time shift parameters optimally with respect to a prediction problem. While a simple grid search would yield the optimal wavelet parameters, it does not scale. Gaussian processes (GPs) have been shown to be effective and even sometimes exceed expert-level performance in tuning machine learning algorithms [2] [3]. They may offer an efficient method to select the best wavelet parameters, i.e. the best wavelet features. But GPs bring their own set of parameters, which leads us to a reciprocal question: how do GP parameters affect the search for the best wavelet features?

We explore the use of GPs to select the best wavelet features in the context of Acute Hypotensive Episode (AHE) prediction. An AHE occurs when over 90% of the mean arterial pressure values in a 30 minute window dip below 60 mmHg, according to the most common definition of an AHE. AHEs represent one of the main dangers patients can face in intensive care units (ICUs): an AHE can be life-threatening, and might necessitate an immediate intervention from a nurse or a physician. The mortality rate of patients who experienced an AHE is more than twice the average in the ICU and one third of patients in ICU have at least one AHE [4]. Pinpointing high-quality AHE predictors would be a contribution of high impact and value.

In this paper we present a system that selects the best wavelet predictors for AHE. The selection is performed with GP in a distributed fashion using a massive, high-resolution signal data set. We detail the system in Section III.

The main contributions of this paper are to:

- Predict AHEs using wavelet features extracted from arterial blood pressure (ABP) signals.
- Assess the impact of changing the wavelet parameters, lag, lead and data set size on the prediction quality.
- Implement a robust, scalable system to perform feature selection in a distributed fashion using GP.
- Find the best GP parameters and evaluate how much time it saves when GP is used for feature selection.

The remainder of the paper is organized as follows. Section II presents the related work, Section III introduces the methods we use, Section IV reports on our experimental results, and the paper concludes with a discussion of future work.

## II. Related work

The problem of AHE prediction drew much attention because of the 2009 Challenge organized by PhysioNet and Computers in Cardiology: the goal was to predict AHEs one hour ahead. Many entries performed well [4], but the challenge was based on a very small subset of MIMIC: only 600 hours of ABP, in comparison with the 240,000 hours that the entire data set contains. As a result, some approaches took advantage of the small size of the data. For example, the winning entry [5] use generalized regression neural network multi-models that demand a computationally extensive training phase. While the challenge resulted in interesting insights, many proposed ideas fail to scale up.

Wavelets have already been successfully used for ECG [6] and a few other medical applications [7]. Previous studies on AHE prediction such as [8] indicated their interest in investigating digital signal processing features. There exist many wavelet transforms, which increases the chances of spotting abnormalities among the signals.

A large-scale machine learning and analytics framework, beatDB, was built to mine knowledge from large physiological signals data sets [9]. It provides flexible configurations for hypothesis definition and feature aggregation. This work extends beatDB allowing it to use wavelets and perform feature selection.

## III. Method

### A. Overview

Our goal is to predict an AHE for some amount of time ahead (lead time) using a given window of ABP history (lag time) as shown in Figure 1. Since in this work we focus on feature selection, we simply use a logistic regression as a binary, probabilistic classifier, with 5-fold cross-validation. All features we use are extracted from the ABP signal.

The raw ABP signals are segmented into beats using a beat onset detection program [10]. As the ABP signal contains artifacts and noise, we used some heuristics to detect invalid beats following the method described in [11] and [12]. Wavelets are extracted for each valid beat. Wavelet values located in the lag are then aggregated into 10 sub-windows of same size. We use the mean as the aggregation function. As a result, for a given wavelet with a given set of parameters, we obtain 10 features. Note that this is not what feature selection works on: feature selection will focus on the sets of parameters.

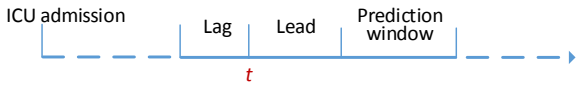


Figure 1. Prediction parameters. $t$ represents the current time. The prediction window corresponds to the duration of the event we try to predict. The lead indicates how much time we try to predict ahead. The lag is the period of time in the past that we use to compute the prediction.

### B. Wavelets

Continuous wavelet transforms (CWTs) are a generalization of Fourier transforms (FTs). They transform signals between time (or spatial) domain and frequency domain. The main difference between CWT and FT is that the former is localized in both time and frequency while the FT is only localized in frequency, thereby enabling CWT to spot abnormalities in a specific part of the ABP signal.

The core idea behind CWT is to use convolutions to quantify how similar the given signal is compared with the given wavelet, like with FT. However, unlike FT, we need to specify both a scale parameter $a$ (a strictly positive real number, a.k.a. dilation parameter), and a time shift, $b$ (a real number, a.k.a. position or translation). The resulting formula is as follows:

$$X(a,b;f,\psi) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t)\overline{\psi}\left(\frac{t-b}{a}\right) dt$$

where:

- $\psi$ is the chosen wavelet,
- $f$ is the signal to be analyzed (namely, the ABP signal).

As a result, we can define a single feature on a beat by specifying one wavelet, one scale and one time shift. For each CWT we investigate, we try different scales, time shifts, lags and leads in order to thoroughly explore the wavelet's ability to predict an AHE. Assessing a specific set of parameters is expensive: grid search is to be avoided. To find the best parameters, we use GP.

### C. Gaussian processes for feature selection

A Gaussian process (GP) is a generalization of the Gaussian probability distribution to functions. It is a particular case of a stochastic process, which is often used to represent the evolution of some random value over time. We use GP to accelerate the search for the best wavelet features.

The data is as follows:

- Training set inputs: $x_{train} = (x_1, x_2, ..., x_n)$
- Training set outputs: $y_{train} = (y_1, y_2, ..., y_n)$
- Test set inputs: $x_{test} = (x_{n+1}, x_{n+2}, ..., x_m)$

In our case, to perform feature selection, each $x_i$ is a 4-dimensional vector (lag, lead, scale and time shift), and each $y_i$ is a 1-dimensional vector (AUROC). $x_{train}$ contains all parameter combinations for which the AUROC has already been computed, and $x_{test}$ are the remaining all parameter combinations. The goal is to find $P(y_{test}|y_{train}, x_{train}, x_{test})$ in order to select among $x_{test}$ the parameter combination $x_i$ that is the most likely to yield a high AUROC.

Since we assume random variables to be Gaussian, we have a closed-form expression for the distribution of the outputs that we were trying to find:

$$P(y_{test}|y_{train}, x_{train}, x_{test}) \sim \mathcal{N}(\mu_{y_{test}}, \sigma^2_{y_{test}})$$

where:

- $A = K_{test-train}(K_{train-train})^{-1}$
- $\mu_{y_{test}} = \mu_{test} + A(y_{train} - \mu_{train})$
- $\sigma^2_{y_{test}} = K_{test-test} - AK_{train-test}$

K is the covariance matrix, and can be seen as the distance matrix between all the inputs (training and test sets put together). The choice of the covariance function, i.e. the function that we use to compute the covariance between two inputs, impacts predictions. The covariance function is typically denoted by $k$, as it is also called kernel.

$$K = \overbrace{\begin{Bmatrix} \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \cdots & k(x_n,x_n) \end{bmatrix} \\ \begin{bmatrix} k(x_{n+1},x_1) & \cdots & k(x_{n+1},x_n) \\ \vdots & \ddots & \vdots \\ k(x_m,x_1) & \cdots & k(x_m,x_n) \end{bmatrix} \end{Bmatrix}}^{x_{train}} \overbrace{\begin{bmatrix} k(x_1,x_{n+1}) & \cdots & k(x_1,x_m) \\ \vdots & \ddots & \vdots \\ k(x_n,x_{n+1}) & \cdots & k(x_n,x_m) \\ k(x_{n+1},x_{n+1}) & \cdots & k(x_{n+1},x_m) \\ \vdots & \ddots & \vdots \\ k(x_m,x_{n+1}) & \cdots & k(x_m,x_m) \end{bmatrix}}^{x_{test}}$$

$$K = \left( \begin{array}{c|c} K_{train-train} & K_{train-test} \\ \hline K_{test-train} & K_{test-test} \end{array} \right)$$

For further details on GP, see [13].

### D. Design and implementation

In order to handle large of data sets, we design a distributed system. One node is used as a data server with a large storage capacity that contains all signal data, a variable number of nodes are used as workers that each computes one AUROC at a time for a set of parameters, and another node is used as a database server that contains all the AUROC results returned by the workers along with the corresponding scale, time shift, lag, and lead parameters that were used.

The data server and the database server are launched first and passively wait for workers to connect whenever the latter need to read or write data. The workers are launched next: the number of workers can be increased or decreased at any time. Each worker performs the same three-step cycle:

1) Feature selection: the worker retrieves the database content and selects the next parameter set to be computed according to the grid search, random search or GP. Once the parameter set is selected, the worker writes a flag in the database so that no other worker computes the same parameter set.
2) Data aggregation: the worker assembles feature data as described in Section III-A.
3) Event prediction: the worker performs the AHE prediction task given the assembled feature data as described in Section III-A, saves the AUROC in the database, and removes its previous flag.

Once the computational budget has been reached or the AUROC has converged, the workers are shut down and the database contains all the results. Figure 2 presents an overview of the worker logic, and Figure 3 shows the interactions between the workers and the database server.

A computer cluster is a hostile environment: router issues, OpenStack DHCP server issues, hard drive failures, connection issues such as with the NFS server and with the Matlab license server, nodes being arbitrarily shut down by the hypervisor, high CPU steal time, and excessive thrashing on some hypervisors. As a result, we create a fourth type of node that acts as a garbage worker collector. It shuts down any worker that fails to return a result in a reasonable amount of time, removes all flags the ailing worker might have left in the database and launches a new node in replacement.

This design results in a robust, scalable system. During the course of our extensive experiments, it ran smoothly for 4 weeks, and scaled up to 200 nodes despite encountering all the above-mentioned cluster-related issues.

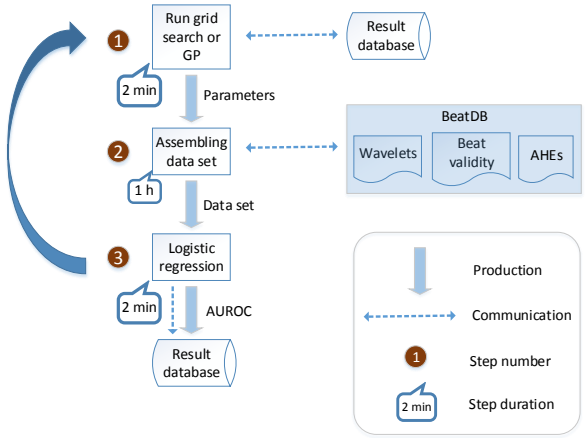


Figure 2. Worker logic. Each worker iteratively computes AUROCs. A worker needs around one hour to compute one AUROC.

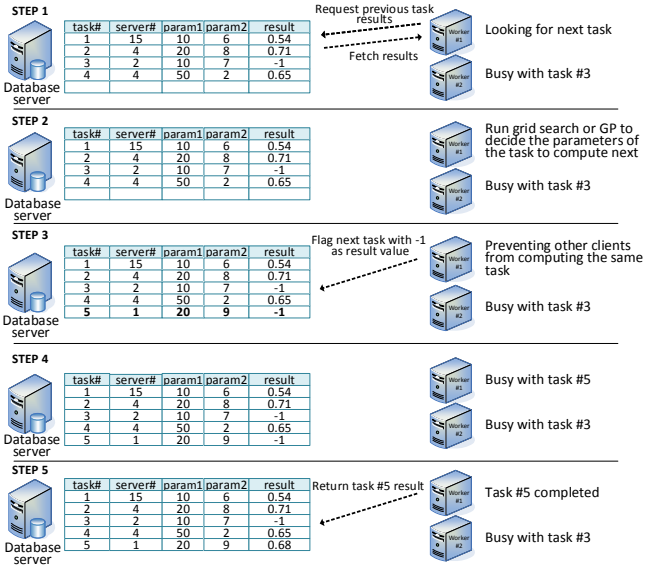


Figure 3. Multi-worker architecture synchronized via a result database. When a worker is launched or has just completed a task, it queries the database, decides which task to compute next based on the result of the query, writes a flag in the database indicating that the task is currently being performed in order to prevent other workers from executing the same task, carries out the actual task, then writes the result in the database.

## IV. Experiments

### A. Data and computing environment

We use the *Multiparameter Intelligent Monitoring in Intensive Care II* (MIMIC II) database version 3, which is available online for free and was introduced by [14], [15]. MIMIC II contains a repository of physiological signals recorded by sensors placed on patients, including the arterial blood pressure (ABP), on which we will focus in the rest of the paper.

There are 6232 patient with ABP recordings. We keep the top 5000 patients with the most ABP data. Since the signal was recorded at 125 Hz and there are a total of 240,000 hours of ABP data, we have 108 billion samples ($240000 \times 60 \times 60 \times 125$) and around 1.2 billion beats (0.9 billion being valid). ABP samples are measured in millimetres of mercury (mmHg). The ratio of AHE events vs. non-AHE events is 10%: this strong class imbalance is one of the main motivations behind the use of AUROC as the metric for the quality of predictions.

We perform the experiments on an OpenStack computer cluster equipped with Intel Xeon L5640 processors. The data is stored on an NFS server. One 2-core OpenStack node with 2 GB of RAM is used as a MySQL database server. The other OpenStack nodes are launched with 4 cores and 4 GB of RAM and are used as workers. The number of workers varies from 1 to 200 depending on resource availability. All nodes run Ubuntu 12.04 x64 LTE.

### B. Wavelet experiments

We first take three different CWTs, viz. Gaussian-2, Haar and Symlet-2. We choose them among ∼100 other CWTs as those 3 CWTs have a low correlation between each other, compared to other sets of CWTs. For each, we explore the first 10 scales, and divide each beat into 19 different time shifts. We pre-compute all wavelet features since we will re-use them when trying different leads, lags, and search parameters: it takes 6 hours using 3 24-core nodes, and storing the results of each CWT requires 300 GB. We also vary the lag and lead times by exploring 6 values for each: [10, 20, 30, 40, 50, 60] minutes. We choose 60 minutes as the maximum lag and lead times by observing the average length of a continuous ABP recording. The questions we investigate in this set of experiments are the following: How does the choice of the CWT impact AHE prediction? What are the best scale and time shift? How much data history (lag time) does one need to make an accurate prediction? How far ahead can one predict AHE? Does a larger data set size lead to a significantly more accurate prediction?

To find the best scale and time shift with different lag and lead times, we perform a grid search for each CWT, i.e. we exhaustively enumerate all possible combinations of parameters. This naive approach will give us a baseline. A worker takes around one hour to compute the AUROC averaged over 5-fold cross-validation for a given CWT, scale, time shift, lag and lead. For each CWT we perform $10 \times 19 \times 6 \times 6 = 6840$ experiments since we try 10 different scales, 19 different time shifts, 6 different lags and 6 different leads. Exploring one CWT therefore costs 6840 hours (150 days) if not done distributedly. Figure 4 presents the AUROCs when the lag and the lead times are fixed to 10 minutes, for the Gaussian-2 CWT. Figure 5 shows the impact of the lag and the lead on the AUROC.

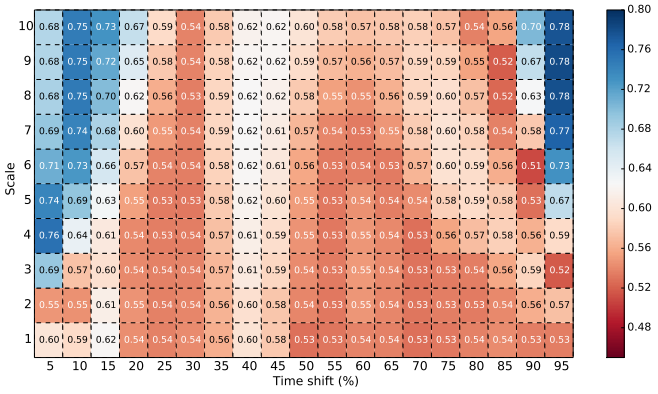


Figure 4. Heat map of the AUROCs for all 10 scales and 19 time shifts using the Gaussian-2 CWT using a lag of 10 minutes, and a lead of 10 minutes. The highest AUROC is 0.78 and is achieved with scale 9 and time shift 95%.

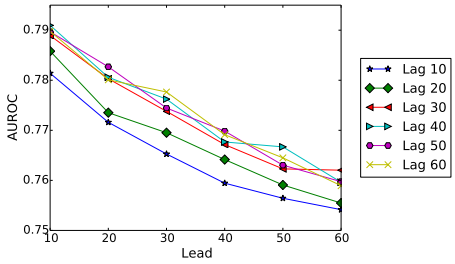


Figure 5. Influence of the lead on the AUROC for the Gaussian-2 CWT. For each pair of lag and lead, we take the maximum AUROC achieved among all 10 scales and 19 time shifts. Increasing the leads reduces the AUROC, which is not surprising as a higher lead means that the prediction is made further ahead in time. Increasing the lag from 10 to 30 minutes improves the AUROC, but once 30 minutes is reached increasing the lag does not further improve the AUROC.

From these experiments we determine that the choice of the CWT has an important impact on the prediction's AUROC. Among the 3 CWTs that we have tried, the Gaussian-2 CWT yields the highest AUROC. Interestingly, the best scales and time shifts stay the same when we change of the lag and lead. Across CWTs we can see some similarities between the best scales and time shifts. Decreasing the lead does make AHE easier to predict, but increasing the lag makes it easier only to some extent: beyond 30 minutes of lag, adding more history is not useful. Increasing the data set size up from 1000 to 5000 patients significantly increases the AUROC (+2.6%) which confirms the importance of

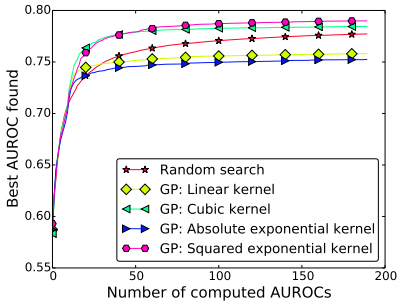


Figure 6. Impact of the kernel choice on the GP with the Gaussian-2 CWT. The squared exponential kernel is the most optimal choice. 10 random points were used for initialization.

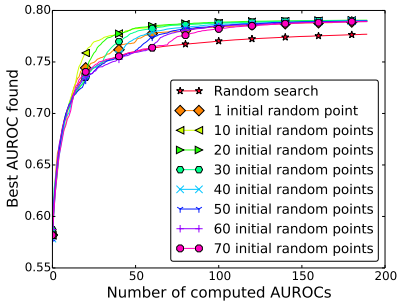


Figure 7. Impact of the choice of the number of random points on the search convergence speed, with the Gaussian-2 CWT. Choosing 10 random points is optimal.

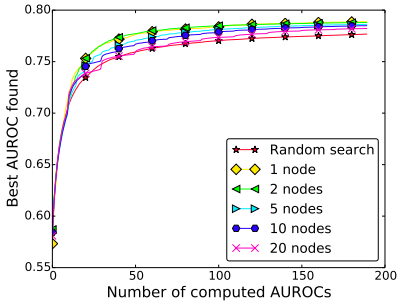


Figure 8. Impact of the number of workers on the convergence speed of the AUROCs obtain with a GP search, using the Gaussian-2 CWT.

performing large-scale studies.

Lastly, we investigate whether these wavelet features can improve the prediction quality when added to 14 statistical features used in [9]: root-mean-square, kurtosis, skewness, systolic blood pressure, diastolic blood pressure, pulse pressure, beat duration, systole duration, diastole duration, pressure area during systole, standard deviation, crest factor, mean, and mean arterial pressure. This comparison is not performed to explore feature selection, but to additively assess CWT as predictors.

Table I summarizes the results. Our results show that adding a wavelet feature leads to a 14% decrease of the false positive rate (FPR) when the true positive rate (TPR) is fixed at 0.90. The additive change to the AUROC was insignificant.

| | Features | Number of features | AUROC | FPR when TPR = 0.90 |
|---|---|---|---|---|
| (1) | Gaussian-2 CWT with scale 9 and time shift 95 | 10 | 0.7897 | 0.61 |
| (2) | Haar CWT with scale 10 and time shift 5 | 10 | 0.7187 | 0.63 |
| (3) | Symlet-2 CWT with scale 6 and time shift 5 | 10 | 0.7286 | 0.61 |
| (4) | 14 initial features using 5 aggregation functions | 70 | 0.9523 | 0.14 |
| | (1) + (4) | 80 | 0.9529 | 0.12 |
| | (2) + (4) | 80 | 0.9528 | 0.12 |
| | (3) + (4) | 80 | 0.9525 | 0.12 |

Table I
CWT features and 14 other features. For each CWT the best scale and time shift are selected.

### C. Gaussian process experiments

In this set experiments we investigate GP to avoid computing exhaustively the AUROCs for all parameter sets. We only compute the AUROCs for the parameter sets selected by the GP based on previously computed AUROCs in order to orient the search toward the most promising parameter sets. In particular, we address the following questions: How much computational effort does using GP save? What are the optimal GP parameters? How does the number of workers impact the convergence speed of the GP? How do GP results compare with the grid search's best AUROCs?

Figure 6 compares the convergence speed of a GP using 4 different kernels and a random search, for the Gaussian-2 CWT. We see that the choice of the kernel is critical: the linear and absolute exponential kernels make the GP worse than the random search, while the squared exponential and cubic kernels perform significantly better than the random search. The squared exponential kernel yields a slightly better result than the cubic kernel.

In order to decide which one to use between the squared exponential kernel and the cubic kernel, we perform 100 searches with each kernel and look at the standard deviation of the AUROC as the number of computed AUROCs increases. The standard deviation is a very important property as in real conditions we will not perform 100 GP searches but many fewer. We therefore need a kernel that supports a reliable GP search, i.e. if we perform two searches the results should be approximately similar (low standard deviation). Figure 9 compares the standard deviation obtained with the squared exponential kernel and the cubic kernel for the Gaussian-2 CWT. The squared exponential kernel is the best choice as it obtains the highest AUROC on average and has the smallest standard deviation.

Figure 6 shows that when performing the search with GP using the squared exponential kernel, one only needs to compute 50 AUROCs to find a near-optimal parameter set (i.e. yielding an AUROC close to the highest achieved with grid search), while it takes random search over 150 AUROCs to find a parameter set of similar predictive quality. The use of GPs reduces the search time by a factor of 3.

Beyond the choice of the kernel, one also needs to decide the number of randomly chosen parameter sets for which the AUROCs need to be computed in order to initialize the GP. Using the squared exponential kernel we vary the number of initial random experiments. Choosing 10 random points is

optimal for this experimental data set, as Figure 7 illustrates.

Lastly, we analyze how the number of workers impacts the convergence speed of the GP. In the previous experiments we have run the GP on one worker only. Intuitively, increasing the number of workers should decrease the convergence speed in terms of number of computed AUROCs, because when one worker fits its GP to find the next parameter set to compute, it has access to fewer results since more tasks are being computed concurrently.

For all experiments we obtained similar results for the Gaussian-2, Haar and Symlet-2 CWTs.

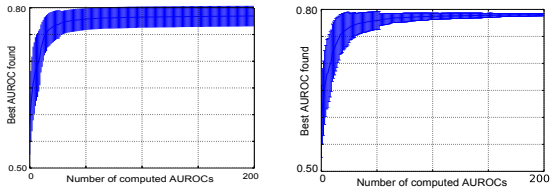


Figure 9. Standard deviation of the cubic kernel (left) and squared exponential kernel (right) with the Gaussian-2 CWT.

## V. Conclusion and Future Work

In this paper we presented a scalable, robust system to perform distributed feature selection on a massive physiological signal data set. With this system we addressed the problem of AHE prediction and assessed wavelets as predictors: the best CWT achieved an AUROC of 0.79, and the FPR when TPR is fixed at 0.90 is reduced by 14% when the wavelet feature is used in conjunction with other statistical features. Decreasing the lead does make AHEs easier to predict, but increasing the lag makes it easier only to some extent: increasing the lag beyond 30 minutes does not improve the AUROC. Furthermore, increasing the data set size up from 1000 to 5000 patients significantly increases the AUROC (+2.6%) which confirms the importance of large-scale studies.

The critical issue with using wavelets is the identification of their optimal parameters. GP allowed us to reduce the search time by a factor of 3. The squared exponential kernel is the best choice as it obtains the highest AUROC on average and has the smallest standard deviation compared to other kernels. GP can be used to optimized any ordinal value and can be a very useful method when performing a grid search is too expensive.

There is still much room for further investigation. We focused on AHE prediction using ABP because of the importance of AHEs and the increasing amount of blood pressure sensors, but our experiments demonstrate that wavelets show promise to predict other medical conditions based on ABP or other physiological signals. Furthermore we restricted ourselves to a very simple classifier, viz. logistic regression, in order to focus on feature selection. Models specialized for time series such as dynamic Bayesian networks [16] might yield higher AUROCs. Lastly, clustering patients may be beneficial for the prediction as well as it would help address the variance among patients.

## Acknowledgment

The authors would like to thank the Li Ka Shing Foundation for its generous support.

## References

[1] P. S. Addison, "Wavelet transforms and the ECG: a review," *Physiological measurement*, vol. 26, no. 5, p. R155, 2005.

[2] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[3] W. Drevo, "Delphi: A distributed multi-algorithm, multi-user, self optimizing machine learning system," *Master's thesis, MIT*, 2014.

[4] G. Moody and L. Lehman, "Predicting acute hypotensive episodes: The 10th annual physionet/computers in cardiology challenge," in *Computers in Cardiology*, vol. 36, 2009, pp. 541–544.

[5] J. Henriques and T. Rocha, "Prediction of acute hypotensive episodes using neural network multi-models," in *Computers in Cardiology, 2009*. IEEE, 2009, pp. 549–552.

[6] C. Li, C. Zheng, and C. Tai, "Detection of ecg characteristic points using wavelet transforms," *Biomedical Engineering, IEEE Transactions on*, vol. 42, no. 1, pp. 21–28, 1995.

[7] M. Unser and A. Aldroubi, "A review of wavelets in biomedical applications," *Proceedings of the IEEE*, vol. 84, no. 4, pp. 626–638, 1996.

[8] Y. Zhou, Q. Zhu, and H. Huang, "Prediction of acute hypotensive episode in ICU using chebyshev neural network," *Journal of Software*, vol. 8, no. 8, pp. 1923–1931, 2013.

[9] F. Dernoncourt, K. Veeramachaneni, and U.-M. O'Reilly, "beatdb: A large scalewaveform feature repository," in *NIPS 2013, Machine Learning for Clinical Data Analysis and Healthcare Workshop*, 2013.

[10] 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.

[11] A. Waldin, "Learning blood pressure behavior from large blood pressure waveform repositories and building predictive models," *Master's thesis, MIT*, 2013.

[12] J. Sun, A. Reisner, and R. Mark, "A signal abnormality index for arterial blood pressure waveforms," in *Computers in Cardiology, 2006*. IEEE, 2006, pp. 13–16.

[13] C. K. Williams and C. E. Rasmussen, "Gaussian processes for machine learning," *The MIT Press*, 2005.

[14] G. B. Moody and R. G. Mark, "A database to support development and evaluation of intelligent intensive care monitoring," in *Computers in Cardiology, 1996*. IEEE, 1996, pp. 657–660.

[15] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[16] Y. B. Kim, J. Seo, and U.-M. O'Reilly, "Large-scale methodological comparison of acute hypotensive episode forecasting using mimic2 physiological waveforms," in *Proceedings of the 2014 IEEE 27th International Symposium on Computer-Based Medical Systems*. IEEE Computer Society, 2014, pp. 319–324.