# SEQUENTIAL PEAK DETECTION FOR FLOW CYTOMETRY

Gökhan Gül, Sabine Alebrand, Michael Baßler, Jörn Wittek
*Fraunhofer Institute for Microengineering and Microsystems (IMM)*
Carl-Zeiss-Str. 18-20, 55129 Mainz, Germany
{Goekhan.Guel, Sabine.Alebrand, Joern.Wittek, Michael.Bassler}@imm.fraunhofer.de

***Abstract*—Circulating tumor cells in blood are identified by means of sequential peak detection taking into account the memory and real time applicability constraints. Three different spatial domain algorithms: derivative approach, energy detector and baseline method are compared with three different peak detection algorithms based on machine learning: linear and non-linear support vector machines and artificial neural networks. Performance of the peak detection algorithms are tested on both synthetic and real data. Experimental results indicate superiority of machine learning algorithms over the other three algorithms which are widely used in practice. Due to Gaussianity assumption in the signal model, a linear support vector machine is found to be as good as other machine learning schemes.**

***Index Terms*—Peak detection, flow cytometry, machine learning, classification, filtering, field programmable gate array.**

## I. Introduction

In biology and medicine flow cytometry is a well-established measurement technique in order to determine the number of cells in a fluidic sample [1]. In this method dispersed cells typically flow through a capillary where they are detected e.g. by fluorescence or stray light measurements after optical excitation. As each cell will lead to an attributed peak in the detection signal, a crucial factor determining the accuracy of cell counting is the accuracy and robustness of the peak detection algorithm. Although instant peak detection is not a mandatory requirement for routine lab applications, for applications where the measurements shall be done continuously (e.g. continuous monitoring of drinking water quality) or where objects shall be manipulated directly after their detection (e.g. circulating tumor cell detection and isolation in blood), real-time peak detection is required [2].

There are plenty of methods applied to the detection of peaks in time series. A conventional way of peak detection is via filtering and thresholding [3]. Depending on the application, more sophisticated filtering approaches may be preferable. For example, for the diagnosis of epilepsy, auto-regressive modeling followed by Kalman-filtering for parameter estimation can be used to detect epileptic spikes [4]. For the detection of electrocardiograph (ECG) signal peaks conventional filtering can be replaced by its adaptive counterpart [5]. Simpler approaches in order to account for signal non-stationaries may consider higher order statistics before thresholding [6], whereas more (computationally) complicated approaches may consider neural network based adaptive matched filtering [7]. All above mentioned techniques have the prerequisite that the entire signal is present at the time of detection. There are also algorithms, which are designed to be able to work online at the absence of the entire signal. For example, filtering and thresholding is again an applicable technique for sequential detection of peaks. Since this approach is able to satisfy real time constraints such as speed and memory, it can be implemented on a field programmable gate array (FPGA) [8]. Another computationally inexpensive approach that is implementable on an FPGA was proposed in [9], where the authors considered local maxima scalogram. Adaptive adjustments of a sliding window depending on the signal to noise ratio (SNR) levels with the following thresholding was proposed for real time peak detection in [10]. Recently, sequential learning with neural networks were proposed for online detection of peaks both in electroencephalography as well as photoplethysmogram signals in [11] and [12].

In this paper sequential real-time peak detection is studied as the initial step of the identification of circulating tumor cells in blood. The real-time application is to be realized on a moderate FPGA chip, therefore, only the algorithms, which are computationally inexpensive and which fulfill storage requirements, i.e.:

(a) Filtering and thresholding
(b) Energy detector
(c) Baseline method
(d) Machine learning (ML) approaches

are considered for comparison. According to our knowledge, this is the first work, which compares different types of sequential peak detection algorithms as the initial step of identification of circulating tumor cells in blood taking into account speed and storage constraints for practically realizable implementations.

The rest of the paper is organized as follows. In Section II, signal model is given. In Section III, sequential peak detection algorithms are introduced. In Section IV, experimental results are presented and finally in Section V, the paper is concluded.

## II. Signal Model

The theoretical signal model is deduced from the real time experimental setup shown in Figure 1, where fluorescent objects, e.g. cells marked with Carboxyfluorescein succinimidyl ester (CFSE) are dispersed in a fluid flowing in a microfluidic channel of $500$ $\mu$m width and $60$ $\mu$m height. Regarding the channel width they are focused around the center of the

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

channel by hydrodynamic focusing (two sheath flows, one sample flow). The flow rate of the sample is $0.05$ milliliter per minute and the sheath flow is $0.5$ milliliter per minute. The size of the objects is typically between $10$- and $15$ $\mu$m. At the point of detection the microfluidic channel is illuminated by a $488$ nm laser. When the fluorescent objects cross the laser illumination zone fluorescence light will be emitted, which is optically filtered from the excitation light and finally collected by a silicon photomultiplier detector. The analog signals from the detector are sampled by an analog to digital converter, which has a sampling frequency of $125$ kHz. Due to the hydrodynamic focusing and a low object concentration chosen in this experiment the objects will ideally pass the illumination zone one after another. Consequently, each object crossing the laser spot will lead to a peak signal in the measured data.

There are two sources of disturbances underpinning the peak signal. The first source is inevitable dark current noise and quantum noise. The second source is caused by the cell flow rate imperfections and may be modeled by an additional lower variance Gaussian signal. Hence, the discretized signal model can be defined as

$$r[n] = \underbrace{s[n] + v[n]}_{x[n]} + \tilde{w}[n] \tag{1}$$

where $s$ is a deterministic Gaussian shaped signal, $v$ is a secondary Gaussian shaped signal with a much lower amplitude than that of $s$, and $\tilde{w}$ is the random noise process. The noise characteristics can be extracted from the real time signal at the absence and presence of primary and secondary signals. At the absence of signal, the noise is white and Gaussian distributed, i.e. $\tilde{w} = w$, with mean $\mu$ and variance $\sigma^2$ as justified by Gaussian curve fitting with respect to minimum mean squared error (MMSE) criterion. At the presence of signal, the noise is correlated to the input signal; the higher the signal amplitude, the lower the effect of noise. This correlation can be extracted again from the real data using the model

$$\tilde{w}[n] := \begin{cases} \frac{w[n]}{\theta_1 x^2[n] + \theta_2 x[n] + \theta_3}, & \text{if } x[n] \geq w[n] \\ w[n], & \text{otherwise,} \end{cases} \tag{2}$$

where $\theta_1, \theta_2$ and $\theta_3$ are parameters to be found based on the best fit of $r$ to the real data according to the MMSE criterion. Once the fitting parameters are found, the theoretical model given by (1) and (2) can be simulated for various signal and noise parameters in order to train and test the designed algorithms.

## III. Sequential Peak Detection Algorithms

In this section, the peak detection algorithms that satisfy the speed and storage constraints are presented. The complexity of training is irrelevant as it can be done off-line but the testing complexity is expected to be linear and the calculations are expected to be parallelizable such that full benefit of the FPGA can be obtained.

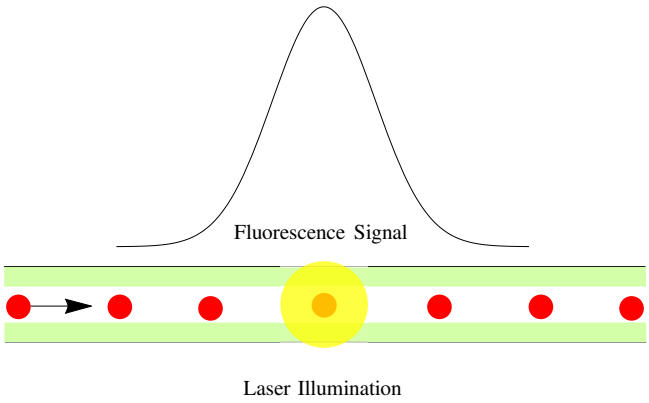


Fig. 1. Microfludic channel with flowing fluorescent objects, e.g. fluorescence marked cells. The objects are illuminated by a laser at the point of detection, leading to a Gaussian fluorescence signal per object.

### A. Filtering and Thresholding

Real time sequential peak detection can be realized through filtering for noise removal and the following thresholding for detection. Since the noise is white and hence it has uniform effect on all frequency components, for the sake of simplicity, the following moving average filter can be used to surpass the effects of noise

$$y_l[n] = \frac{1}{L} \sum_{k=0}^{L-1} r[n-k], \tag{3}$$

where $L$ is the filter length. The thresholding after low-pass filtering may be not robust against high amplitude spikes which may be present due to voltage fluctuations. Therefore, the high-pass filtering

$$y_h[n] = y_l[n] - y_l[n-1], \tag{4}$$

followed by the low-pass filtering of $y_h$ with (3) leading to $y_d$ and thresholding as

$$p[n] := \begin{cases} r[n-1], & \text{if } y_d[n-1] < 0,\ y_d[n] > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

is preferred. This method is also known as the derivative approach.

### B. Energy Detector

For a deterministic signal in white Gaussian noise, where the complete signal is available, and the maximum likelihood estimation (MLE) of the signal in noise results nothing but the received signal itself, the optimum detector which maximizes SNR is known to be the energy detector [13, p. 250]

$$y_e[n] = \frac{1}{W} \sum_{k=0}^{W-1} r[n-k]^2, \tag{6}$$

where $W$ is the moving window length. The peak signal $p$ is obtained by thresholding $y_e$, i.e. replacing $y_d$ by $y_e$ and the threshold $0$ by some suitable threshold $t$ in (5).

### C. Baseline Tracking

This algorithm takes into account the deviations from the baseline assuming that the peak signal can statistically be modeled as time series of outliers which deviate from the majority of data points with respect to the signal amplitude. In order to identify the outlying signal, an adaptive baseline signal $y_b$ is generated from $r$ as

$$y_b[n] := \begin{cases} y_b[n-1]+1, & \text{if } 0 < r[n]-y_b[n-1] < \tau, \\ y_b[n-1]-1, & \text{if } 0 < y_b[n-1]-r[n] < \tau, \\ y_b[n-1], & \text{otherwise,} \end{cases} \quad (7)$$

with the initial condition $y_b[0] = r[0]$, where $\tau$ is a parameter defining the strength of deviations up to which the baseline signal $y_b$ follows the output signal $r$. Above the given threshold $\tau$, the baseline signal $y_b$ stops following $r$ and waits until the condition $|r[n]-y_b[n-1]| < \tau$ holds again. Let $t_1$ be the time instance when $|r[n]-y_b[n-1]| \geq \tau$ holds for the first time and let $t_2 > t_1$ be the time instance when the condition $|r[n]-y_b[n-1]| < \tau$ holds once again. The peak is detected as

$$p[n] := \begin{cases} \max(r[n], p[n-1]), & \text{if } t_1 \leq n \leq t_2 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

with the initial condition $p[t_1] = r[t_1]$. In order to prevent false alarms due to unrealistic (low variance) peaks, e.g. due to noise or secondary physical effects, $t_2 - t_1$ needs to be bounded. That is, for all $p[n] \neq 0$ if $t_2 - t_1 < t$ then $p[n] := 0$ for some user defined threshold $t$.

### D. Machine Learning Algorithms

In the sequel three different machine learning algorithms will be introduced. The details of classification and peak detection will be explained in Section IV.

*1) Linear SVM:* For a two class classification problem, given $l$ samples of $n$-dimensional training vectors $\boldsymbol{x}_i \in \mathbb{R}^n$ and their corresponding class labels $y_i \in \{-1, 1\}$, $i \in \{1, \ldots, l\}$, the solution of the following primal optimization problem

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned} \quad (9)$$

is an affine $n$-dimensional hyperplane parameterized by $\boldsymbol{w}$ and $b$, where $C > 0$ is a user defined regularization parameter. Solving the primal problem (or its dual see [14]) leads to the training. Consequently, for every unknown sample $\boldsymbol{x}$, the testing is obtained by

$$\text{sign}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right). \quad (10)$$

The Equation (10) indicates that the complexity of testing is linear in the number of features, $n$.

*2) Non-Linear SVM:* A linear SVM can be extended to a non-linear SVM via changing the inner product term ${\boldsymbol{x}_i}^T\boldsymbol{x}_j$ by its kernelized version $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, in the dual optimization problem and solving the underlined equations for training. The testing is done by

$$\text{sign}\left(\sum_{i=1}^{l}\alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \quad (11)$$

where $\alpha_i \in \mathbb{R}$ are the Lagrangian parameters imposed on the dual problem [14]. The testing complexity of non-linear SVM depends on the kernel but it is $\mathcal{O}(nl)$ for some well known kernels such as the Gaussian kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$. In comparison to linear SVM, roughly $l$ times more memory is required. Memory and speed requirements can be linearized by either cropping the support vectors or reducing the number of training samples.

*3) Artificial Neural Networks:* An artificial neural network (ANN) can be defined as a graph, which connects the input data (data samples) to the output data (class labels). ANNs comprise an input layer, several hidden layers and an output layer. Each neuron in a particular layer computes a function of the form $f(\boldsymbol{w}^T\boldsymbol{x}+b)$, where $f$ is called the activation function, e.g. $f(x) = \tanh(x)$. There are various training algorithms to train the ANNs e.g. the back propagation algorithm [15]. Neural networks are highly parallelizable in software and hence for reasonably chosen numbers of layers and neurons, ANNs satisfy both the low complexity testing as well as the storage constraints to be implementable on an FPGA.

## IV. Experimental Results

The following steps are taken in the experiments:

- Find the optimum parameters of the signal model defined by Equations (1) and (2) from the real data according to MMSE criterion.
- Train and test different peak detection algorithms on the designed signal model.
- Test the optimized algorithms on the real data.

### A. Optimum Signal Parameters

Random signal parameters extracted from the real data are the mean $\mu$ and variance $\sigma^2$ values of the noise $w$. Deterministic signal components $s$ and $v$ have known parameters but they vary from one peak signal to the other. From the real data, distribution of mean, variance and peaks (maximum amplitudes) of $r$ were extracted. How $x$ is divided into the signal $s$ and artifact $v$ components for each peak signal is unclear. Therefore, we made the assumption that the variance of $s$ and $v$ are the same at each realization, as the variance is related to the speed of the cells, however, the artifact signal $v$ has a maximum amplitude which is eight times less than that of the peak signal $s$. This assumption makes the distribution of the peaks of $r$ similar to that of $s$. Hence, we can model the distribution of the peaks of $s$ based on the distribution of the peaks of $r$. The location of $v$, -the mean value of the artifact Gaussian signal-, is dependent on the location of the

TABLE I
DETECTION PERFORMANCES OF MACHINE LEARNING ALGORITHMS FOR TWO DIFFERENT CLASSIFIERS.

| | Lin. SVM | Non-lin. SVM | ANN |
|---|---|---|---|
| Noise vs. Signal | 0.8921 | 0.8996 | 0.9006 |
| Peak vs. Other Signal Comp. | 0.9231 | 0.9601 | 0.9716 |

peak signal $s$. This dependency is randomly modeled, i.e., the artifact signal has a mean $\mu_v = \mu_s + U$, where $U$ is a uniformly distributed random variable on $[-\epsilon, \epsilon]$, where $\epsilon$ is a small positive number.

### *B. Simulations on the Signal Model*

Once the signal parameters and their distributions are obtained, the signal model can be used for training and testing of various peak detection algorithms. In order to simulate various signal to noise ratios peak amplitudes of $s$ are modeled by the peak amplitude $AU$, where $U$ is a random variable which is uniform on $[0, 1]$ and $A$ is the mean peak value. Accordingly, SNR is defined as follows:

$$\text{SNR} = 10 \log_{10} A/\sigma. \tag{12}$$

In the training phase, for each specific SNR value, optimum parameters of the first three peak detection algorithms are determined using a grid search approach. For machine learning based peak detection algorithms, optimum classifier parameters are determined only once and for a specific SNR value, i.e. 12.5dB. This uneven process may only result in a slight drawback against the machine learning approaches because under the noise only hypothesis the noise is stationary. The details of training for the specific algorithms are as follows. For the non-linear SVM a Gaussian kernel is adopted. The ANN is a feed-forward network with two hidden layers, each having 20 neurons. For training the ANN, the Levenberg-Marquardt algorithm is used where each neuron is associated with the $\tanh$ activation function [16]. There are two classifiers designed, one of which separates the noise from the signal and the other separates peaks from other signal components such as falling edge or rising edge. In order to obtain the training set, peak signals are randomly sampled from the related signal or noise components by a window of size $W = 20$ generating 15000 samples of length 20 per class. The same process is repeated once again to obtain the testing data set. The training and testing data sets (each $30000 \times 20$ matrices) are created randomly for 50 times. Optimum classifier parameters are found on the training data set using five-fold cross validation. The average detection accuracy is used for performance assessment. The detection performances of three different machine learning algorithms for SNR=12.5dB are listed in Table I. Although the performances of algorithms for noise vs. signal classification are similar, the linear classifier is not as good as the other two classifiers for the classification of peaks from the other signal components.

In the testing phase, all six (trained or optimized) algorithms are tested using Monte-Carlo simulations with 10000 randomly generated peak signals $r$. As mentioned, each machine learning algorithm is identified with two classifiers. The first classifier decides whether the window corresponds to noise or signal, and in case it belongs to a signal, the second classifier decides whether it is a non-peak or peak. If the second classifier fails to detect any peak, mean position of the signal components from the first classifier is used for peak detection. For all algorithms a peak is assumed to be detected correctly if the related detector gives a positive decision which varies at most $T = 10$ discrete points in time from the actual position of the original peak. Since for every $N = 4000$ discrete points in time, only one peak signal is generated (e.g. $s$ with minimum width of 10 points and maximum width of roughly 60 points), the data is unbalanced with respect to the first classifier, i.e. there are possibly much more false alarms (in fact maximum of $k = N - W - 2T$) than missed detections. Therefore, Neyman-Pearson type of classifiers are designed with false alarm ratios close to zero. For the second classifier, the data is much more balanced, therefore false alarm constraints are relaxed in comparison to the first classifier. Let $P_F$ denote the false alarm probability, $P_M$ denote the miss detection probability and $P_0$ denote the a-priori probability of the noise-only hypothesis against the alternative hypothesis that the data sample is a peak within an acceptable margin $T$. Then, the overall performance of algorithms can be evaluated by the Bayesian error probability

$$P_E = P_0 P_F + (1 - P_0) P_M \tag{13}$$

where $P_0 = (k - 1)(1 - P_0)$. Taking into account the speed and storage constraints the sliding window and filter lengths of $W, L \in \{10, 20, 30\}$ have been considered for all algorithms. Additionally, in order to be resistant against artifact peaks, which are more apparent for high SNRs, a peak is declared if no higher amplitude peak appears after 60 points in time, and otherwise the previously detected peak is replaced by the higher amplitude one. The best results obtained by all six algorithms have been depicted in Figure 2. We can see that all machine learning schemes have similar error probabilities and they perform better than the conventional peak detection algorithms (e.g., ca. 1dB difference at low SNRs in comparison to the baseline method). At high SNRs there is an error floor characteristics. This is due to random peak amplitudes which are lower bounded by 0 unlike peak widths which are lower bounded by 10 points in time.

### *C. Simulations on the Real Data*

The real data has been obtained by a physical procedure described in Section II. There are around 100 peaks recorded in the data, which are around $2.9 \cdot 10^7$ discrete points in length. Baseline of the signal is not constant and changes slightly in time. This difference from the developed signal model necessitates retraining of the peak detection algorithms. Furthermore, the data obtained is not labeled and therefore whether the peaks really correspond to the blood cells or some artifacts is unknown. All six peak detection algorithms have been applied to the real data and were able to detect the peaks successfully. For the sake of clarity the results of only two peak detection algorithms have been plotted and shown in Figure 3.

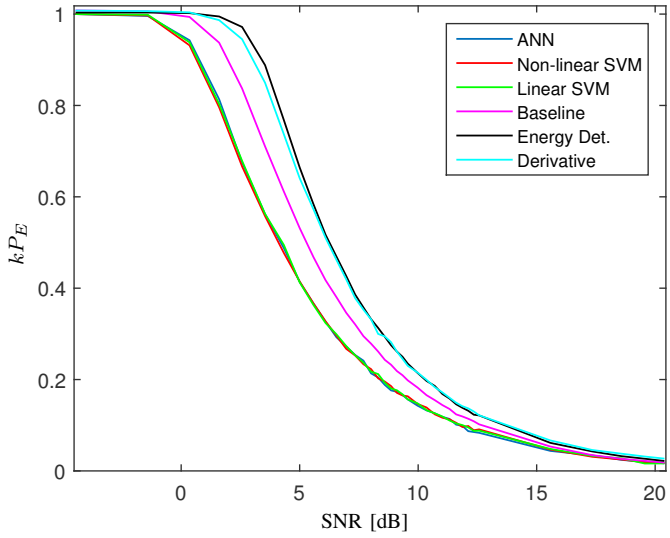


Fig. 2. Scaled error probability over various SNR values for six different peak detection algorithms.

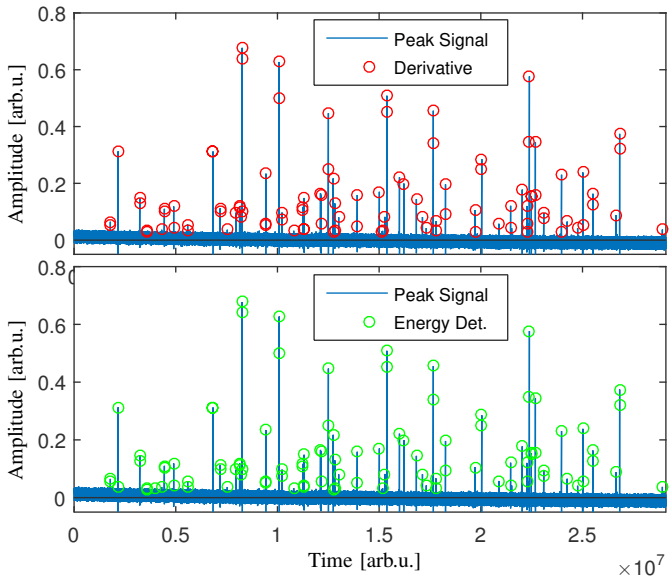


Fig. 3. Peak signal and detected peaks for the derivative approach and the energy detector.

## V. Conclusion

The problem of sequential peak detection as the initial step of the identification of circulating tumor cells in blood has been studied. In order to model the characteristic of the measurement device a theoretical model with some free parameters has been considered. Optimum parameters of the model have been found according to MMSE criterion. Based on this model three different conventional and three different machine learning based sequential peak detection algorithms have been proposed. The machine learning schemes employ a hierarchical classification procedure, where out of two classifiers the first classifier separates signal from the noise and the second classifier extracts the peak from the signal. Simulation results on the synthetic data indicate that the energy detector and the derivative approach have similar performances whereas the baseline method performs better. Additionally, the peak detection algorithms based on machine learning outperform the conventional schemes although the training process for the machine learning algorithms considers only a single SNR point. On the real data all algorithms were able to detect high amplitude peaks, while again the ML algorithms were better for lower amplitudes. The results obtained are promising in order to develop measurement systems that benefit both from a cost optimized hardware and a high accuracy peak detection algorithm.

## References

[1] Marion G. Macey, *Flow Cytometry: Principles and Applications*, Humana Press, Totowa, NJ, 2007.

[2] Wen-Hui Weng, I-Lin Ho, Chi-Chia Pang, Sow-Neng Pang, Tung-Ming Pan, and Wai-Hung Leung, "Real-time circulating tumor cells detection via highly sensitive needle-like cytosensor-demonstrated by a blood flow simulation," *Biosensors and Bioelectronics*, vol. 116, pp. 51–59, 2018.

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

[4] A. T. Tzallas, V. P. Oikonomou, and D. I. Fotiadis, "Epileptic spike detection using a kalman filter based approach," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2006, pp. 501–504.

[5] S. Jain, A. Kumar, and V. Bajaj, "Real-time detection of electrocardiograph peaks: A genetic algorithm based approach," in *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb 2017, pp. 262–266.

[6] M. Schmidt, J. W. Krug, A. Gierstorfer, and G. Rose, "A real-time qrs detector based on higher-order statistics for ecg gated cardiac mri," in *Computing in Cardiology 2014*, Sept 2014, pp. 733–736.

[7] Q. Xue, Y. H. Hu, and W. J. Tompkins, "Neural-network-based adaptive matched filtering for qrs detection," *IEEE Transactions on Biomedical Engineering*, vol. 39, no. 4, pp. 317–329, April 1992.

[8] H. K. Chatterjee, R. Gupta, and M. Mitra, "Real time electrocardiogram wave peak detection algorithm and its implementation on fpga," in *Proceedings of The 2014 International Conference on Control, Instrumentation, Energy and Communication (CIEC)*, Jan 2014, pp. 204–209.

[9] A. M. Colak, Y. Shibata, and F. Kurokawa, "Fpga implementation of the automatic multiscale based peak detection for real-time signal analysis on renewable energy systems," in *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, Nov 2016, pp. 379–384.

[10] Dong H. Zhou Q. Xu M. Li X. Wu G. Ni, K., "Interference peak detection based on fpga for real-time absolute distance ranging with dual-comb lasers," in *2015 International Conference on Optical Instruments and Technology: Optoelectronic Measurement Technology and Systems*, Aug 2015, vol. 9623.

[11] B. N. Sumukha, R. C. Kumar, S. S. Bharadwaj, and K. George, "A novel approach to peak detection using sequential learning algorithm," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, Dec 2016, pp. 862–867.

[12] B. N. Sumukha, R. C. Kumar, S. S. Bharadwaj, and K. George, "Online peak detection in photoplethysmogram signals using sequential learning algorithm," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 1313–1320.

[13] Steven M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*, Prentice-Hall Inc, New Jersey, 1993.

[14] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[15] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–, 1986.

[16] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *Trans. Neur. Netw.*, vol. 5, no. 6, pp. 989–993, 1994.