A 1.2-0.55V General-purpose Biomedical Processor with Configurable Machine-learning Accelerators for High-order, Patient-adaptive Monitoring

Kyong Ho Lee, Naveen Verma
Department of Electrical Engineering, Princeton University, Princeton, NJ 08544
email: {kyonglee, nverma}@princeton.edu

***Abstract*— Machine learning offers powerful advantages in sensing systems, enabling the creation and adaptation of high-order signal models by exploiting the sensed data. We present a general-purpose processor that employs configurable machine-learning accelerators to analyze physiological signals at low energy levels for a broad range of biomedical applications. Implemented in 130nm LP CMOS, the processor operates from 1.2V-0.55V (logic). It achieves real-time EEG-based seizure detection at 273μW (at 0.85V) and patient-adaptive ECG-based cardiac-arrhythmia detection at 124μW (at 0.75V), yielding overall energy savings of 62.4$\times$ and 144.7$\times$ thanks to the accelerators.**

I. Introduction

Low-power ambulatory technologies have emerged for acquiring physiologically-indicative patient signals (e.g., EEGs, ECGs, etc.). The key challenge to enabling high-value biomedical devices, however, is deriving clinically-relevant inferences from such signals. Specific disease states must be detected in the presence of numerous physiologic variances, raising the need for high-order signal models. Additionally, the manifestations of the disease states can be highly variable from patient-to-patient and over time [1], [2], requiring dynamic model adaptations yet with minimal effort from human experts. Machine learning offers powerful capabilities for overcoming these challenges [3], but the computations are energy intensive and have limited low-power sensors to use them only on a highly reduced scale [4]–[6]. While low-power processors have recently emerged for traditional DSP algorithms [7], [8], we present a flexible processor *specialized for high-order machine-learning functions*. This design introduces the following contributions:

- A flexible accelerator is integrated with a general-purpose CPU. The CPU enables programmable feature extraction for a range of physiological signals and clinical applications, and the accelerator, through hardware configurability, enables design-point selection within an algorithmic-performance vs. energy and memory-usage space.
- Dynamic, patient-specific model adaptations are enabled in the background through an in-place hardware implementation for embedded active learning. This permits hardware sharing for kernel functions while minimizing the overhead of context switches during real-time detection.
- Algorithms are developed and demonstrated (using clinical datasets), showing the use of the processor for continuous, patient-adaptive monitoring within a network (enabled by an integrated Bluetooth radio interface).

II. Architecture

The processor architecture is derived from the energy limitations in medical applications, specifically caused by the need

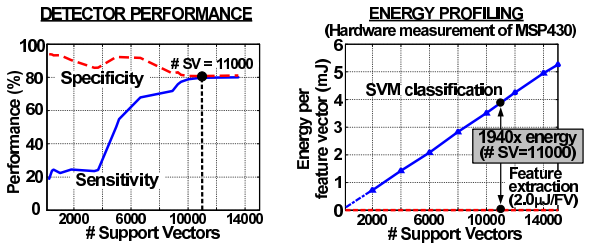


Fig. 1. Performance and energy of representative machine-learning algorithm [ECG-based cardiac arrhythmia detection using patient data (from [9])].

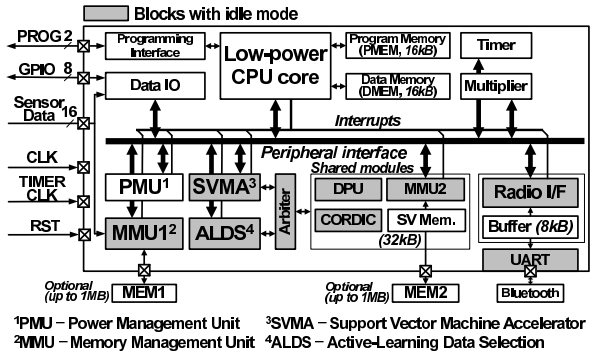


Fig. 2. Processor architecture with machine-learning accelerators.

for high-order models. The support-vector machine (SVM) is a state-of-the-art machine-learning framework for modeling and classification. It analyzes signal segments by representing them as feature vectors. A classification model is formed by constructing a decision boundary using selected feature vectors from a labeled training set. The chosen vectors, called the support vectors (SVs), reflect the model complexity. Having profiled several medical applications, we show the typical challenge in Fig. 1; namely, high-order models are needed for accuracy, but they cause the energy to scale and dominate over feature extraction and sensor instrumentation.

In the proposed processor, configurable accelerators are used to enable flexible modeling and classification at low energy. A block diagram is in Fig. 2: a general-purpose 16b CPU provides programmable feature computation over a wide range of signals; an SVM accelerator (SVMA) enables various classifier algorithms and formulations for application diversity and energy scalability; an active-learning data selection (ALDS) accelerator enables dynamic model adaptations, with minimal human effort, by selecting the optimal instances of sensor data for model refinement; and a radio interface manages exchange of data and models so that the monitoring and patient-adaptation processes can occur over a network. Additionally, a power-management unit (PMU) enables idle-mode clock gating of each block through software, and

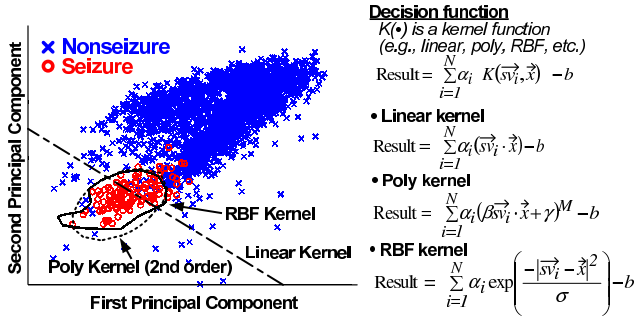


Fig. 3. SVM decision boundaries in an EEG-based seizure-detection application for different kernel functions (functions shown on the right).

memory-management units (MMU1, MMU2) enable background control for continuous data logging and model updates.

III. Accelerator Architectures

As shown in Fig. 2, the SVMA and the ALDS share computational modules, including a data-path unit (DPU) and CORDIC engine, as well as memory and control modules (i.e., a 32kB SV memory and MMU2, which enables extensibility to off-chip memory). The blocks are described below.

A. Support Vector Machine Accelerator (SVMA)

The SVMA is configurable via memory-mapped control registers for algorithmic and performance-vs-energy flexibility. For algorithmic flexibility, the SV memory can be partitioned to create multiple SVM instances, which can then be combined for multi-class and adaptive-boosting algorithms [10]. For energy-performance flexibility, three kernel functions (K) can be selected [i.e., linear, polynomial, and radial-basis function (RBF)] and an alternate formulation for the polynomial kernel can be chosen [11]. Fig. 3 shows the kernel computations (where $\vec{x}$ represents an input feature vector, $s\vec{v}_i$ represents the SVs, and α_i, β, γ, M, σ, and b represent training parameters); while the linear kernel is the least computationally intensive, the polynomial and RBF kernels provide increasing flexibility to enhance SVM performance, (as illustrated in Fig. 3 for EEG feature-vector data, projected to two dimensions via principal component analysis for visualization).

It has been shown that polynomial kernels can provide high performance (comparable to RBF) for many medical applications and that the computational energy for large SV models can be substantially reduced through the following formulation [11]:

$$\begin{aligned} f(x) &= \sum_{i=1}^{N} (\beta s\vec{v}_i \cdot \vec{x} + \gamma)^2 \alpha_i y_i - b \\ &= \sum_{i=1}^{N} [1 \quad \vec{x}]\,[\gamma \quad \beta s\vec{v}_i]^T\,[\gamma \quad \beta s\vec{v}_i]\,[1 \quad \vec{x}]^T\,\alpha_i y_i - b \qquad (1) \\ &= \sum_{i=1}^{N} \vec{X}\vec{S}_i{}^T \vec{S}_i \vec{X}^T \alpha_i y_i - b = \vec{X}\Big(\sum_{i=1}^{N} \vec{S}_i{}^T \vec{S}_i \alpha_i y_i\Big)\vec{X}^T - b \end{aligned}$$

By representing the SVs as a matrix, the kernel is formulated as a linear multiplication, allowing the input feature vector to be factored out of the summation. This enables precomputation over all of the support vectors, overcoming the energy scaling of Fig. 1. The trade-off, however, is that matrix formulation results in more severe energy scaling with the number of features. Nonetheless, simulations profiling the proposed accelerator are shown in Table I (for two representative medical applications [1], [12]), indicating that this formulation results

TABLE I

CYCLE-COUNT/MEMORY REQUIREMENTS AND PERFORMANCE (BASED ON PATIENT DATA [9]), SHOWING THE POTENTIAL FOR ENERGY SCALABILITY

Arrhythmia detection†					
Kernel	Performance			Cycle count (kcycles)	SV memory required (kB)
	True Pos.	True Neg.	False Pos.		
RBF	75.9%	90.3%	25.4%	1341	640.0
Poly (2nd order)	74.6%	89.0%	28.1%	1684	1094.4
Poly Reform.				1.8	1.9
Linear	57.1%	90.6%	30.4%	0.09	0.08
Seizure detection††					
Kernel	Performance			Cycle count (kcycles)	SV memory required (kB)
	Sensitivity	Latency	Specificity		
RBF	100%	4.8 sec	1.2 /day	29.6	16.5
Poly (2nd order)	100%	4.4 sec	2.4 /day	24.9	16.2
Poly Reform.				9.7	9.4
Linear	100%	15.0 sec	18.0 /day	0.20	0.19

Sensitivity: (# seizures detected) / (# total seizures) x 100
Latency: average delay of detector after electrographic onset
Specificity: # false positives per day
† Support vectors required for RBF, poly and linear kernels are 14246, 24363 and 14246, respectively.
†† Support vectors required for RBF, poly and linear kernels are 169, 166 and 72, respectively.

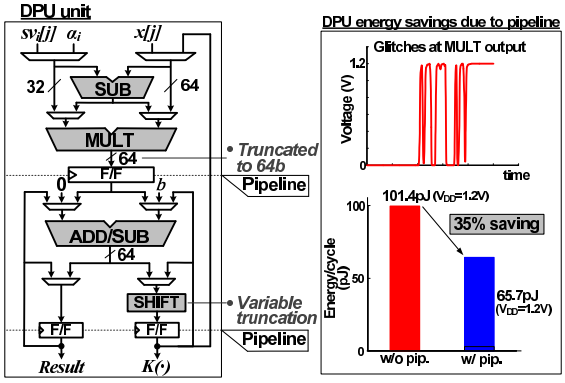


Fig. 4. Flexible data-path unit (DPU) to support feature-space computations; simulations show the energy benefit of the pipeline architecture.

in substantial reduction of the memory requirements and cycle count, thus indicating energy savings; results for the RBF and linear kernels are also shown, illustrating that the SVMA enables scalability in performance versus energy and memory-usage. For an arrhythmia detector, the polynomial reformulation gives 745× and 337× reduction in cycle count and memory usage (compared to an RBF kernel), while for seizure detection, it gives 3× and 1.8× reduction, respectively. The linear kernel offers further savings by trading-off performance.

B. Data-path Unit (DPU)

The classifier kernel functions are implemented by the flexible DPU shown in Fig. 4. The model parameters (sv_i, α_i, and b) are applied in a specialized two-stage pipeline. Flip-flop insertion for the pipeline substantially mitigates active-glitching and leakage energy (i.e., by reducing the critical-path delay [13]); transistor-level simulations (Fig. 4) show that glitch-mitigation results in 35% energy savings at V_{DD}=1.2V.

The barrel shifter in the DPU enables computation of decision functions having large SV sets and large SV dimensionality by truncating intermediate computation results: otherwise, computational overflows result even with small SV dimensionalities and sets, yielding poor performance, as in Fig. 5(a). Bit-true simulations show that the shifter enables performance comparable to a floating-point implementation, reliably managing the variable dynamic range needs. Fig. 5(b) shows analysis for selecting the SV bit precision based on the RMS error for classification. 16b representations are found to yield sufficient accuracy over a wide range of profiled

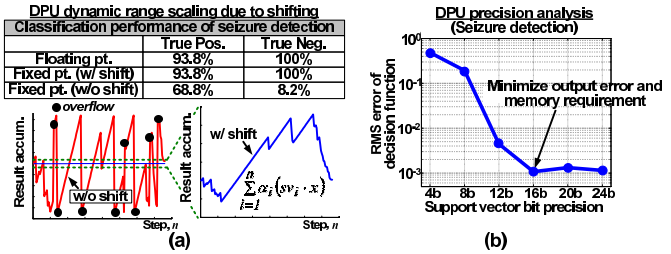


Fig. 5. (a) The in-line truncation of shifting prevents overflow yielding comparable performance to the floating-point implementation. (b) 16b SVs minimize memory requirement while retaining classifier performance.

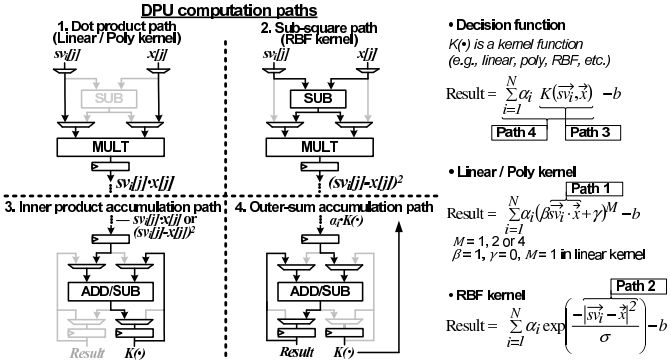


Fig. 6. DPU computational paths (DPU simplified for illustration).

applications; beyond this the accuracy tends to saturate.

The various computational paths through which the DPU implements the selectable kernel functions and their formulations is shown in Fig. 6. Paths 1 and 2 represent configurations for the top half of the DPU (Fig. 4), and paths 3 and 4 represent configurations for the bottom half. The various decision-function formulations are realized via the four paths, though the exponential transformation for the RBF kernel is achieved by the CORDIC engine (described in Sec. III-D).

C. ALDS Accelerator

Patient-customized models lead to substantially improved accuracy in medical detectors [1], [2]. However, significant clinical effort is required to form a customized SV model. The ALDS accelerator enables *active learning*, where a patient-generic model seeds the detector, and the sensed patient data is continuously assessed to choose the most informative instances for refining the model. This permits algorithms (e.g., [2]) where greatly-reduced patient data can be transmitted to clinical experts over the network to minimize the burden of data analysis and model training.

Fig. 7 shows the ALDS computations. Instances from a large pool of sensed data are selected based on a score, which uses two metrics that can be programmably weighted. The first metric (c) is the output magnitude of the SVM decision function (i.e., marginal distance from decision boundary), representing the confidence of the current model. The second metric (d) is provided to enhance the diversity of the selected data; this enables coverage over large regions of the feature space, as in Fig. 8, and is shown to result in substantially improved model convergence [14]. Both metrics involve computations in the kernel-transformed feature space (e.g., d derives the angle between transformed vectors); the SVMA and DPU, which are specifically optimized for feature-vector kernel computations, are thus leveraged. As shown in Fig. 7, however, a large

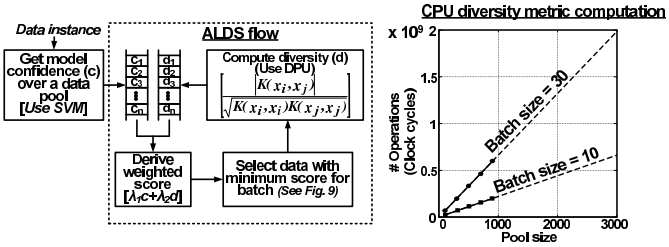


Fig. 7. ALDS computations (left) and simulation of CPU core (right), illustrating the need for background computation via an accelerator.

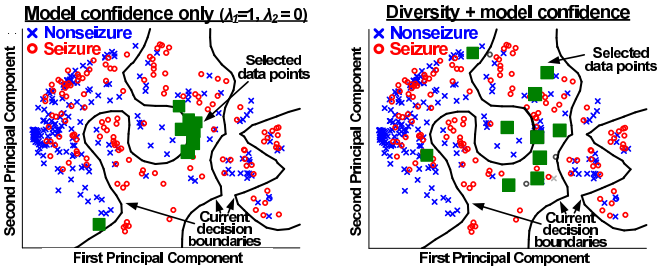


Fig. 8. Without diversity metric (left), selected data may be clustered; with diversity metric (right), large regions of feature space are represented.

number of operations are required as the pool size increases. Since large pools are necessary for the greatest data reduction, enabling background computation via the ALDS accelerator is an important feature.

However, since model adaptations occurs slowly, the arbiter of Fig. 2 enforces priority over the SVMA and DPU for real-time SVM computations. Interruptions of the ALDS occur between the iterations shown in Fig. 7. To minimize the overhead of context switches, we use an 'in-place' implementation of the selection process, as shown in Fig. 9. Since the weighted selection metric for each feature vector is positive, it can be stored as a 15b number; this allows the MSB to be used as a running marker to indicate selected feature vectors, which can thus be skipped during subsequent iterations.

D. CORDIC Module

An embedded CORDIC engine is used to enable hardware-efficient computation of the exponential function required in the RBF kernel. The limitation with conventional CORDIC architectures, however, is a narrow range of convergence, as shown in Fig. 10. To extend the range, we employ a pre-scale and post-scale scheme [15], as shown. The pre-scale represents the argument as a quotient (Q) and a remainder (D), which explicitly lies within the convergence range. The post-scale then derives the correct value through simple shifting operations. As shown, the effective convergence range is thus substantially increased with minimal hardware complexity.

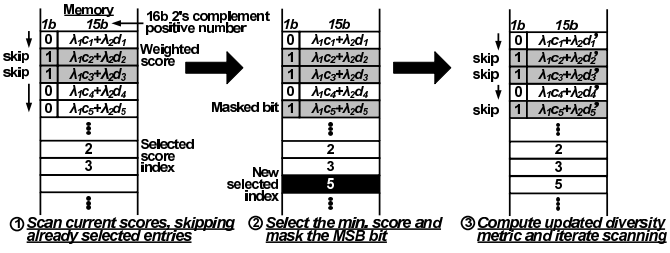


Fig. 9. Weighted metrics are stored as 15b, and MSB is used to indicate previously selected data points to be skipped in subsequent iterations.

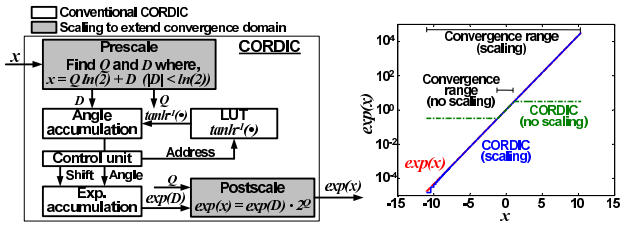
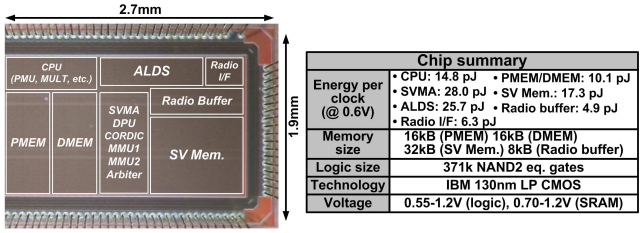


Fig. 10. CORDIC implementation (left) and the resulting region of convergence with and without scaling (right).



Chip summary	
Energy per clock (@ 0.6V)	• CPU: 14.8 pJ • PMEM/DMEM: 10.1 pJ • SVMA: 28.0 pJ • SV Mem.: 17.3 pJ • ALDS: 25.7 pJ • Radio buffer: 4.9 pJ • Radio I/F: 6.3 pJ
Memory size	16kB (PMEM) 16kB (DMEM) 32kB (SV Mem.) 8kB (Radio buffer)
Logic size	371k NAND2 eq. gates
Technology	IBM 130nm LP CMOS
Voltage	0.55-1.2V (logic), 0.70-1.2V (SRAM)

Fig. 11. Die photo and prototype IC summary.

IV. Measurement Results and Demonstrations

The processor is implemented in 130nm LP CMOS from IBM (shown in Fig. 11). It operates from 1.2-0.55V (0.7V for SRAMs), and the measured energy/cycle for the CPU, SVMA, and ALDS modules (including their SRAMs) is shown. The processor enables algorithms employing high-order data-driven signal models and patient-adaptive capabilities, and it can be used for a range of biomedical sensor applications, thanks to programmable signal feature extraction by the CPU.

To demonstrate the application potential, we have implemented a seizure detector (based on the 18-channel EEG algorithm in [1]) and an arrhythmia detector (based on the single-channel ECG algorithm in [12]). We have also implemented a patient-adaptive arrhythmia detector. All tests are performed using clinical patient data [9].

The test setup and a block diagram of the adaptive algorithm are shown in Fig. 12. The ALDS module performs on-going batch selection, iteratively choosing 5 instances out of each pool of 100 feature vectors (this results in 20x data reduction, though much larger pools can be used in practice for greater data reduction). The radio I/F transmits batches and receives updated SV models over a Bluetooth radio that communicates with a base station (Bluetooth also permits interfacing with cell phones for WAN connectivity). The radio I/F controls Rx and Tx, by sharing the 8kB radio buffer. Fig. 13 shows measured results from the chip; active-learning affords substantial improvement, iteratively achieved over a patient-generic model that seeds the detector. The random-learner samples batch data at random, highlighting the performance benefit of the ALDS approach.

Fig. 14 shows the measured energy for the applications. The

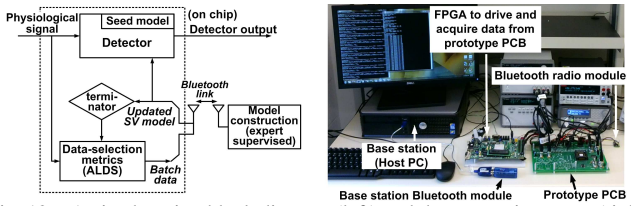


Fig. 12. Active learning block diagram (left) and demonstration setup (right).

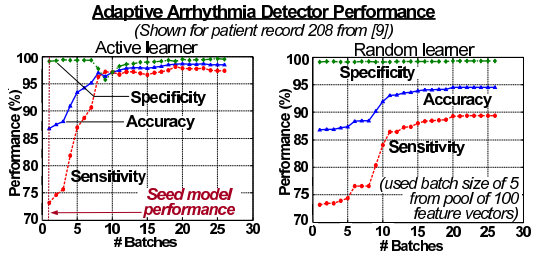


Fig. 13. Active learning performance measured from the chip showing the benefit of the ALDS approach.

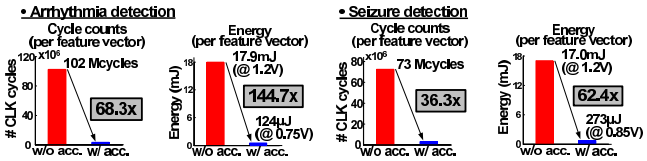


Fig. 14. Cycle count and energy savings for arrhythmia and seizure detection.

accelerators reduce the total cycle counts by 68.3× and 36.3× for the arrhythmia and seizure detectors, respectively; this allows the processor to reduce V_{DD} (without the accelerators, the applications cannot run in real time even at full V_{DD}). The resulting energy savings are 144.7× and 62.4×, with a total power consumption of 124μW (@ 0.75V, 1.5MHz) and 273μW (@ 0.85V, 2MHz), respectively.

Acknowledgment

Partial support provided by the Qualcomm Innovation Fellowship and GSRC, one of six research centers under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation Entity. IC fabrication provided by MOSIS.

References

[1] A. Shoeb, et al., "Application of Machine Learning to Epileptic Seizure Onset Detection," *Int. Conf. Machine Learning*, Jun. 2010.
[2] K. J. Jang, et al., "Scalable customization of atrial fibrillation detection in cardiac monitoring devices: Increasing detection accuracy through personalized monitoring in large patient populations," *EMBC*, Aug. 2011, pp. 2184-2187.
[3] A. Csavoy, et al., "Creating support circuits for the nervous system: Considerations for "brain-machine" interfacing," *Symp. on VLSI Circuits*, Jun. 2009, pp. 4-7.
[4] N. Verma, et al., "A Micro-power EEG acquisition SoC with integrated seizure detection processor for continuous patient monitoring," *Symp. on VLSI Circuits*, Jun. 2009, pp. 62-63.
[5] A. Shoeb, et al., "A micropower support vector machine based seizure detection architecture for embedded medical devices," *EMBC*, Sept. 2009, pp. 4202-4205.
[6] J. Yoo, et al., "An 8-channel scalable EEG acquisition SoC with fully integrated patient-specific seizure classification and recording processor," *ISSCC*, Feb. 2012, pp. 292-294.
[7] J. Kwong, et al., "An energy-efficient biomedical signal processing platform," *ESSCIRC*, Sept. 2010, pp. 526-529.
[8] M. Ashouei, et al., "A voltage-scalable biomedical signal processor running ECG using 13pJ/cycle at 1MHz and 0.4V," *ISSCC*, Feb. 2011, pp. 332-334.
[9] Physionet. *http://www.physionet.org*
[10] R. Schapire, "A brief introduction to boosting," *Proc. 16th Int. Joint Conf. Artificial Intell.*, 1999
[11] K. H. Lee, et al., "Improving kernel-energy trade-offs for machine learning in implantable and wearable biomedical applications," *Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 1597-1600.
[12] P. de Chazal, et al., "Automatic Classification of Heartbeats Using Morphology and Heartbeat Interval Features," *IEEE Tran. Biomed. Eng.*, Jul. 2004, pp. 1196-1206.
[13] M. Seok, et al., "A 0.27V 30MHz 17.7nJ/transform 1024-pt complex FFT core with super-pipelining," *ISSCC*, Feb. 2011, pp. 342-344.
[14] K. Brinker, "Incorporating Diversity in Active Learning with Support Vector Machines," *Int. Conf. Machine Learning*, Aug. 2003.
[15] J. A. Walthur, "A unified algorithm for elementary functions," *AFIP Spring Joint Computer Conference*, 1971, pp. 379-385.