HARDWARE SPECIALIZATION OF MACHINE-LEARNING KERNELS: POSSIBILITIES FOR APPLICATIONS AND POSSIBILITIES FOR THE PLATFORM DESIGN SPACE

(Invited)

Kyong Ho Lee, Zhuo Wang, and Naveen Verma
Princeton University

ABSTRACT

This paper considers two challenging trends affecting low-power sensing systems: (1) the applications of interest increasingly involve embedded signals that are very complex to analyze; and (2) the platforms themselves face elevating constraints in terms of energy and possibly cost. Motivated by the complexities of analyzing the application signals, we emphasize the benefits of data-driven approaches. Most notably, these approaches are based on machine learning, as opposed to traditional DSP. We consider how the algorithms lend themselves to specialized signal-analysis platforms. Hardware specialization is well-regarded as an approach to address issues of computational efficiency, performance, and capacity, thus playing a key role in leveraging Moore's Law. However, we describe how hardware specialization of machine-learning kernels, this time with an explicit focus on error resilience, can also play a powerful role in enabling *system-wide fault tolerance*, thereby aiding Moore's Law on another dimension.

Index Terms— accelerators, embedded systems hardware resilience, machine learning.

1. INTRODUCTION

Generally speaking, the physical systems that we are interested in sensing can be extremely complex; these might include physiological systems in the context of medical devices [1], high-value infrastructure in the context of industrial monitoring [2], or large-scale civil structures in the context of smart cities [3]. Constructing analytical models of the embedded signals to enable inferences from sensor data is very often unviable. In such scenarios, data-driven signal-analysis techniques can play a valuable role [4]. Data-driven techniques use sensor data, not only to probe a physical system of interest, but also as means of constructing models for signal analysis. Such techniques have gained high relevance in embedded sensing applications for two reasons: (1) powerful frameworks for data-driven modeling and analysis have recently emerged from the domain of machine learning, and (2) sensor networks, composed of low-power nodes, have made data, that is potentially highly informative, available on a large scale. As a result of these factors, sensing nodes have begun to incorporate data-driven approaches to solve complex embedded-signal-analysis problems [5, 6, etc.].

Aside from analysis of application signals, data-driven techniques have also shown great promise for addressing modeling and prediction challenges on the platform-design and -validation levels. This time, the strength of these techniques has been in representing a platform's own states, when such states are too complex to model analytically. Successful applications of machine-learning methods for managing workloads in high-performance computing platforms [7], thermal fluctuations in voltage-performance scalable systems [8], bug exposure in post-silicon validation methodologies [9], etc., are all driving explorations of how such methods can be more extensively leveraged.

While machine-learning methods appear to have strong relevance to the design of low-power embedded systems, in fact they have been investigated from the perspective of ultra-low-power systems on only a very limited level. The purpose of this paper is to motivate such a perspective. This involves three aspects. The first aspect is that machine-learning frameworks have not been formulated with low-power system constraints in mind. As a result, even very simple frameworks can pose limiting challenges in terms of computational energy, memory requirements, etc. The Second aspect is that by precisely considering how machine-learning frameworks might be used within sensing applications, architectures for low-power devices can exploit hardware specialization, yielding a strong lever for overcoming oppositions to the system constraints. Once suitably incorporated within the hardware architecture, the third aspect is that machine-learning methods can aid the operation and design-space tradeoffs for the platform itself. In particular, we look at the urgent challenge of hardware faults, which can arise due to aggressive manufacturing and/or low-energy circuit-operation regimes. In both cases, the challenges associated with modeling the faults and/or their manifestations within the embedded data makes machine learning a potentially powerful approach.

2. MACHINE-LEARNING KERNELS FROM THE PERSPECTIVE OF PLATFORM ENERGY

Machine learning is a rich domain, offering a wide range of frameworks. Broadly, the various frameworks take different approaches to statistical modeling in an effort to address either the behavior of the system being modeled or the manner in which an algorithm might use the resulting model. For the purposes of this paper, we focus on a class of frameworks called discriminative methods. These take a

simplistic approach, in that they attempt to model a system as a particular variable of interest. Despite this simplicity, discriminative methods form the basis for powerful classification/regression algorithms, which can address a wide range of emerging sensor applications [10]. Below, we provide an analysis of the energy limitations posed by such frameworks and how these can be addressed either through mathematical approximation or reformulation.

For analysis and illustration at the kernel-function level, we focus on support-vector-machine (SVM) classifiers, which have gained popularity due to their computational efficiency and training robustness. The SVM is a discriminative supervised-learning framework that performs classification on data, which has been represented as a feature vector. During a training phase it uses training feature vectors to construct a decision boundary in the feature space that optimally separates regions associated with different classes of interest; though binary classification is most common, multi-class algorithms can be achieved via various extensions [11]. The decision boundary is represented by selecting instances of training feature vectors that lie near the edges of the different class-data distributions. These instances are called the support vectors, and are used to compute the classification decision function.

The challenge that typically arises is that for the strongest classification kernels, the energy of computing the decision function scales with the complexity of the support-vector model. Fig. 1 illustrates this energy scaling for a radial-basis-function (RBF) kernel, profiled on an MSP430 microprocessor, with respect to two dimensions that represent model complexity: (1) feature-vector dimensionality, and (2) size of the support-vector set. As a result, in many applications, where high-order models are required, the energy of the kernel dominates. The points corresponding to two representative medical-sensor applications are shown emphasizing the importance of both dimensions.

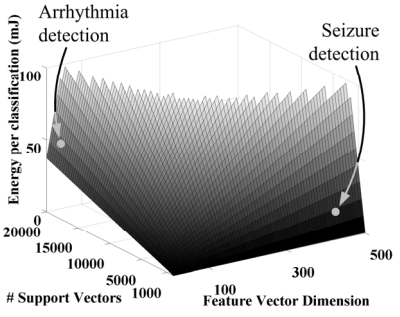

Fig. 1: Energy of SVM classification on MSP430 showing scaling with size of the support-vector set and the dimensionality of feature vectors [12].

To address the kernel energy, a variety of approaches have been considered. In [13] a very low-energy formulation for simple linear kernels is used. Eq. 1 gives the SVM decision function, showing how a linear kernel permits a factorization wherein the computation over all support vectors is reduced via summation to a single vector ($\boldsymbol{sv_i}$ is a support vector, $\boldsymbol{x}$ is the input data's feature vector, and all other parameters are from training). This overcomes the energy scaling shown in Fig. 1 with respect to the size of the support-vector set.

$$\begin{aligned} &Let\ \ sv_i = [sv_{i1}\ sv_{i2} \cdots sv_{iM}]\ \ and\ \ x = [x_1\ x_2 \cdots x_M] \\ &\sum_{i=1}^{N} K(sv_i, x)\alpha_i y_i - b \\ &= \sum_{i=1}^{N} (sv_i \cdot x)\,\alpha_i y_i - b \qquad \text{[Linear kernel]} \\ &= \left(\sum_{i=1}^{N} sv_i\,\alpha_i\,y_i\right)\cdot x - b \ \ = \ \ w \cdot x - b \end{aligned} \tag{1}$$

The challenge is that in many applications, linear kernels show poor performance compared to non-linear kernels; non-linear kernels benefit from the ability to represent much more flexible decision boundaries in the feature space. To enhance the performance of weak, low-energy kernels, a powerful technique is adaptive boosting (adaboost) [14]. It constructs a strong kernel by iteratively combining weak kernels in a linear combination; the weights for the linear combination are determined through data-driven training. The resulting classifier is substantially more flexible, yielding low-energy and greatly enhanced performance in many applications [15].

Rather than approximating a desired decision boundary through boosting, another approach is to approximate the decision function associated with a strong kernel. In [16], non-linear SVM kernels, including RBF and sigmoid, are implemented in a logarithmic number system. This has the benefit of reducing multiplications to additions and subtractions, and, thanks to the reduced precision requirements enabled by logarithmic compression, the logarithmic transformations can be realized via look-up tables and/or piece-wise linear approximations.

Although approximations can substantially mitigate energy consumption, they do not overcome the problematic energy scaling with respect to model complexity shown in Fig. 1. To address this an alternate formulation, that uses increased dimensionality to create a linear kernel from a non-linear kernel, has been proposed [12]; as with the linear-kernel formulation in Eq. 1, this enables factorization to overcome energy scaling with the size of the support-vector set. Eq. 2 shows the formulation applied to polynomial kernels. Though these kernels offer only an intermediate level of flexibility compared to RBF kernels, they have shown to yield comparable performance for many applications, and they substantially outperform linear kernels [12].

The linear formulation is achieved by converting the support vectors into matrices. The matrix formulation has the drawback that energy-scaling with feature-vector dimensionality is exacerbated (e.g., scaling is quadratic for a second-order polynomial kernel); however, as described in [17] several generic and application-specific techniques

$$
\begin{aligned}
&Let\ sv_i = [sv_{i1}\ sv_{i2} \cdots sv_{iM}]\ and\ x = [x_1\ x_2 \cdots x_M] \\
&\sum_{i=1}^{N} K(sv_i, x)\alpha_i y_i - b \\
&= \sum_{i=1}^{N} (\beta\, sv_i \cdot x + \gamma)^2 \alpha_i y_i - b \qquad \text{[2nd order polynomial kernel]} \\
&= \sum_{i=1}^{N} [1\ x_1\ x_2 \cdots x_M] \begin{bmatrix} \gamma \\ \beta sv_{i1} \\ \beta sv_{i2} \\ \vdots \\ \beta sv_{iM} \end{bmatrix} [\gamma\ \beta sv_{i1}\ \beta sv_{i2} \cdots \beta sv_{iM}] \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} \alpha_i y_i - b \\
&= [1\ x_1\ x_2 \cdots x_M] \underbrace{\left(\sum_{i=1}^{N} \begin{bmatrix} \gamma \\ \beta sv_{i1} \\ \beta sv_{i2} \\ \vdots \\ \beta sv_{iM} \end{bmatrix} [\gamma\ \beta sv_{i1}\ \beta sv_{i2} \cdots \beta sv_{iM}] \alpha_i y_i \right)}_{Precomputed\ (M+1\ x\ M+1\ matrix)} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} - b \quad (2)
\end{aligned}
$$

exist for dimensionality reduction. As a result, the formulation enables substantial energy savings, between 30-2000× in medical-sensor applications [12].

3. EXPLOITING ALGORITHMIC STRUCTURE FOR SPECIALIZATION

The previous section noted that a key limitation of machine-learning kernels is how their energy scales with model complexity. This section looks at algorithmic structure at the application level to explore architectures for hardware specialization that can address kernel energy. An important observation is that signal-analysis algorithms can typically be divided into two parts: (1) signal feature extraction, to suitably represent the data in preparation for classification; and (2) classification, through the application of a high-order model. Table 1 shows energy profiling results from two representative applications pertaining to medical-signal analysis. What we see is that datasets requiring high-order models for analysis lead to a scenario wherein classification energy substantially dominates. We note, however, that classification can be achieved through various kernel functions. Feature extraction, on the other hand, requires a high degree of programmability, as it is closely tied to the application and application signals of interest. Its energy, on the other hand, is far lower. These factors suggest that feature-extraction computations should preferably be delegated to highly-programmable hardware (i.e., CPUs) while classification should be delegated to specialized, energy-efficient co-processors. In an integrated microprocessor, this motivates an accelerator-based architecture.

Table 1. Feature extraction and classification energy; Classification energy dominates over feature extraction.

Application	Feature Extraction Energy	Classification Energy	Ratio*
Arrhythmia	1.56 mJ per feature vector	49.52 mJ per classification	31.7
Seizure	1.44 mJ per feature vector	26.98 mJ per classification	18.7

*Ratio = (Classification Energy) / (Feature Extraction Energy)

Fig. 2 shows an accelerator-based microprocessor that was proposed for SVM-based signal-classification applications [18]. The SVM accelerator (SVMA) and active-learning data-selection (ALDS) modules compute classification kernels and training-data selection metrics for active learning [19], respectively; supporting modules, such as a data-path unit for arithmetic and a CORDIC for non-linear transformations, are also included.

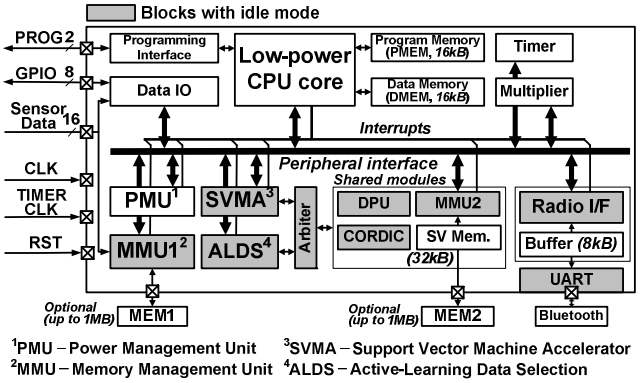

Fig. 2: Architecture block diagram of an accelerator-based microprocessor [18].

An important concern with a hardware-specialized architecture, however, is the need for selective configurability. As discussed in the previous section, there exist a wide range of kernels and kernel formations, leading to a variety of implementation tradeoffs. A noteworthy characteristic of data-driven modeling frameworks is that the optimal design-point parameters can depend strongly on the application data. To enable optimization of the design-point parameters, the classification accelerator in [18] enables configurability of the kernel function and its formulation. Table 2 shows the performance as well as computational complexity (represented by cycle count) and memory requirements for two medical-sensor applications over the configuration space.

Fig. 3 shows the die photo and the chip summary along with the measured energy from two medical-sensor applications. As shown, the use of hardware accelerators dramatically reduces the overall energy (by 144× and 62×, respectively) thanks to the algorithmic structure of the applications.

4. DATA-DRIVEN HARDWARE RESILIENCE THROUGH MACHINE-LEARNING KERNELS

An architecture based on hardware accelerators, as above, is motivated by asymmetry in how specific computations impact the overall energy; namely, hardware specialization is beneficial for certain computations if reducing the energy of those computations has substantial leverage for reducing the overall energy of the application. In this section, we show that with machine-learning kernels, a new driver for hardware specialization may also be resilience; i.e., resilience of machine-learning kernels can be leveraged to substantially improve the overall platform resilience.

Table 2: Illustration of performance, computational-complexity, and memory usage of kernels over configurability space [18].

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.

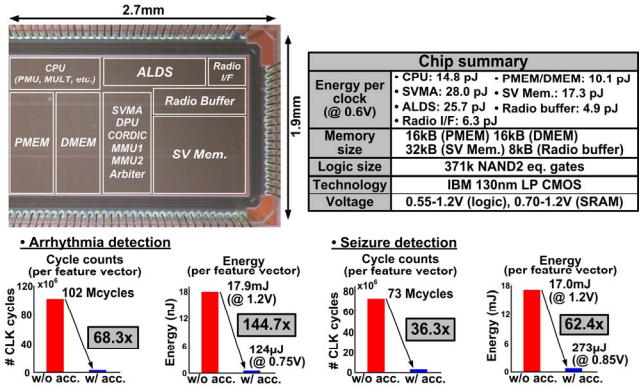

Fig. 3: Die photo, chip summary, and measured application energy of accelerator-based microprocessor [18].

Asymmetry based on resilience characteristics is in fact a principle that has been exploited previously. Primarily, this has been in the context of control-flow errors in a processor, which have a more severe impact on system resilience compared to data-computation errors. Recognizing this has motivated specialized architectures and design methodologies. In the case of architectures, either portions of a microprocessor [20] or entire cores [21] have been explicitly designed to be error resilient for handling control flow, while the remainder of the microprocessor or cores have been designed to have relaxed reliability for data computation. In the case of design methodologies, approaches have been developed wherein the control-critical logic paths of a microprocessor are synthesized to have increased timing margin [22]. Aside from control-flow errors, hardware asymmetry has also been used in signal-processing architectures to efficiently exploit redundancy through the use of statistical error correction. Algorithmic noise tolerance (ANT) [23] employs a primary, full-precision processor that is allowed to make errors alongside a reduced-precision estimator. While the full-precision processor thus benefits from relaxed design constraints, the estimator permits error detection and correction, through the the use of the estimator output. Other approaches for exploiting redundancy have also been proposed [23].

The availability of an error-protected machine-learning accelerator enables an alternate approach. Below we present an overview and simulation results for data-driven hardware resilience (DDHR) [24]. In this approach, learning enables adaptations in a classification decision function. This overcomes the impact of bit-level errors originating from hardware faults in the feature-extraction processor, but also in all previous stages (including data-acquisition and data-conversion blocks). Thanks to the kernel-function formulations and hardware-specialized architectures presented previously, the energy and area costs of a resilient classifier implementation can be substantially reduced; DDHR thus targets the hardware associated with acquiring and deriving signal features, which are becoming increasingly complex and diverse with the expanding scope of applications.

4.1 Overview of the Approach

Fig. 4 illustrates the concept of DDHR. A key premise of machine-learning systems is that the data used during the learning (training) phase must exhibit the same statistics as the data expected during the analysis phase. In the approach shown, the training features are thus obtained from the error-prone feature-extraction subsystem. Using this data, an *error-aware model* can be constructed for classification. The error-aware model is meant to model the feature-data variances caused due to the application signals, but *also* those caused due to hardware faults. However, model construction for a supervised-learning classifier requires training labels in addition to training data. It has been shown [24], that if a temporary error-free detector can be implemented within the architecture, this can be used to estimate the training labels; though the resulting estimates deviate from ground truth, simulation experiments show that they can enable performance very close to that of the error-free system. It should be noted that the requirement of preserving the signal statistics implies that, generally, only static faults can be addressed through this approach; transient fault sources can cause the statistics to be altered, compromising the performance of the trained model.

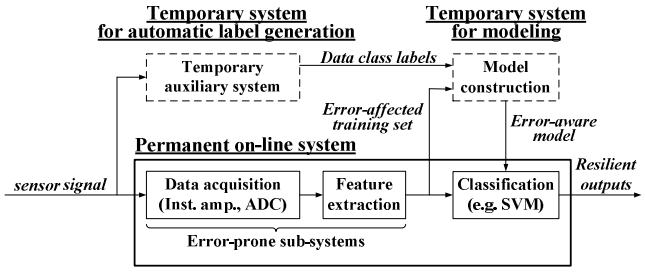

Fig. 4: Illustration of DDHR concept. Temporary systems for modeling and labeling enable construction of an error-aware model from error-affected training data. The error-aware model is then used by a real-time classifier.

Fig. 5 illustrates the impact of hardware faults on feature data and shows how an error-aware model addresses this. The data shown is obtained from gate-level simulations

performed on an EEG-based seizure-detection processor that has been synthesized to an ASIC standard-cell library. To simulate faults, the synthesized netlist has been edited to insert logic-gate switching faults and SRAM bit-cell faults at random. As shown, the original decision boundary, trained using error-free feature vectors, exhibits poor discrimination of the error-affected seizure and non-seizure feature data. An error-aware model, trained using the error-affected feature vectors is also shown, illustrating that good discrimination of the resulting data is now achieved.

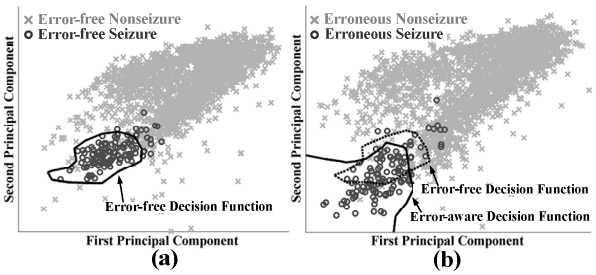

Fig. 5: EEG-signal feature data for a seizure detector (plotted in 2D using principle component analysis) showing (a) the error-free distribution and (b) the distribution after introducing errors (in 20% of the memory cells).

Fig. 6 illustrates the impact of the error-aware model in restoring the performance of the classification decision function. The first histogram shows the classifier output with an original model when using error-free feature data as the input. As shown, good separation between seizure and non-seizure classes is observed. The second histogram shows the impact of injected faults, illustrating the loss of classifier separability. Finally, the third histogram shows the output with an error-aware model when using error-affected feature data as the input. As shown separability performance close to that of the original system is restored.

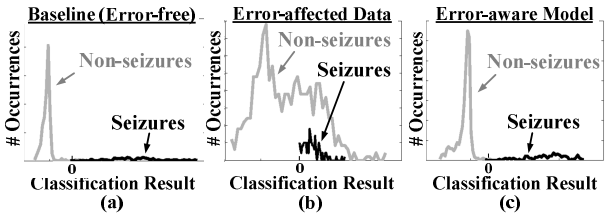

Fig. 6: Histograms of the classifier output for a seizure detector (a) without errors (baseline), (b) with memory errors, and (c) with errors, but using an error-aware model.

4.2 Analysis and Demonstrations

To demonstrate the DDHR concept, two biomedical sensing applications are considered: electroencephalogram (EEG) based epileptic seizure detection [25] and electrocardiogram (ECG) based arrhythmia detection [26]. The EEG and ECG signals used are obtained from the CHB-MIT Scalp EEG Database and the MIT-BIH Arrhythmia Database, respectively [27]. For both applications, the performance metrics considered are true positive rate (TP), true negative rate (TN), false positive rate (FP), and false negative rate (FN); these are widely used metrics for evaluating binary classification.

For both applications, an RTL description of the feature-extraction processor is developed and synthesized to an ASIC standard-cell library to obtain a gate-level netlist. Two types of hardware faults are then injected at a controlled rate into the netlist. The first fault type is SRAM bit-cell faults, wherein the data stored in the SRAM is statically assigned a value of logic '1' or '0' with 50% probability. The second fault type is logic-gate switching faults, wherein the output nodes of logic gates are statically tied to logic '1' or '0' with 50% probability. Since the precise location of faults can strongly influence the manifestations of errors, ten instances of error-injected netlists are simulated in all cases, with errors injected at random locations.

Fig. 7 shows the error statistics of the feature-data derived from gate-level simulations, illustrating sever impact of the injected faults. The first plot shows the magnitude of the resulting errors by plotting the RMS value of the errors normalized to the RMS value of the true features; in many cases, the errors exceed the actual feature values. The second set of plots show the error histogram distributions for four representative features; the histograms exhibit highly-irregular distributions.

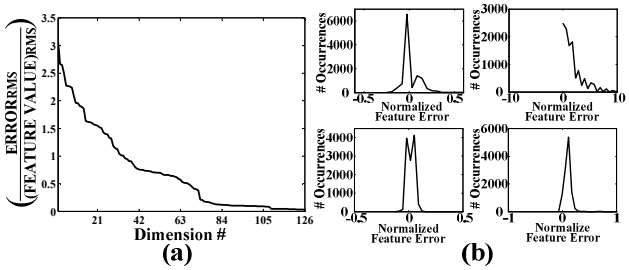

Fig. 7: (a) RMS error by feature (normalized to the RMS of the true feature value) and (b) representative error distributions from four features (cases shown correspond to 20% memory bit-cell errors in the seizure detector).

Despite the large error magnitudes and irregular error distributions in the resulting feature data, the overall performance of the systems is substantially restored thanks to the use of an error-aware model. Fig. 8 shows the performance (averaged over all ten cases at each fault rate) following SRAM bit-cell error injection. As shown, the performance of the original model rapidly degrades, while that of the error-aware model is essentially restored to error-free levels. Fig. 8 shows the performance following logic-gate error injection. The performance exhibits higher variance for this fault type; thus results for all ten cases are shown at the respective fault rates where the performance of the error-aware model begins to degrade. As shown, at these fault rates the original model exhibits degraded performance, while the error-aware model exhibits restored performance (note, TN and FN results are not shown as they exhibit no notable degradation).

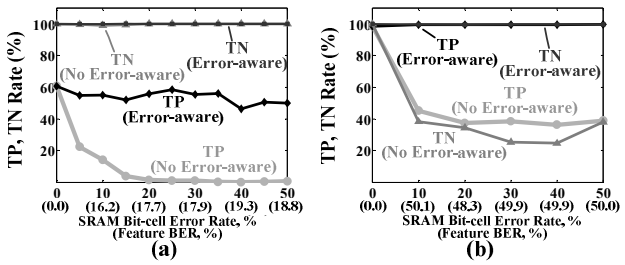

Fig. 8: Performance following SRAM error injection of (a) seizure detector and (b) arrhythmia detector.

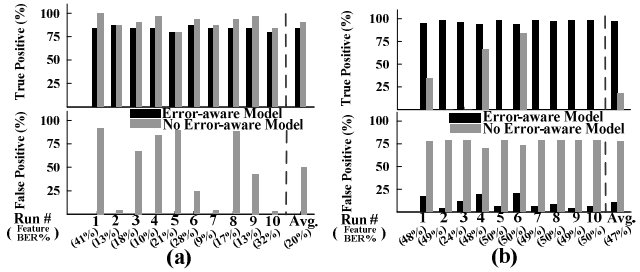

Fig. 9: Performance following logic-gate error injection of (a) seizure detector with fault rate of 10^{-4} errors/node (10 runs), and (b) arrhythmia detector with fault rate of $7x10^{-2}$ per node (10 runs).

5. CONCLUSIONS

Electronics for embedded applications face a challenging two-sided constraint. On one hand, applications are raising the need to perform analysis over increasingly complex sensor data. On the other hand, the hardware platforms themselves are plagued by system-resource limitations such as energy consumption, memory size, and hardware resilience. Given the opposing tradeoffs typically faced by platforms, methods that simultaneously address these constraints can have substantial impact. Machine-learning approaches can play such a role by enabling new algorithmic capabilities for handling application signals, while at the same time raising opportunities for hardware specialization due to algorithmic structure. Such specialization can be leveraged to not only enable substantial energy savings, but also to enable system-wide fault tolerance.

6. ACKNOWLEDGEMENTS

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

7. REFERENCES

1. Csavoy, G. Molnar, and T. Denison, "Creating support circuits for the nervous system: Considerations for "brain-machine" interfacing," *VLSI Symp. Circuits*, Jun. 2009, pp. 4–7.
2. G. Sartain and C. Purdy, "Transforming a 100-year-old industry through analytics," Teradata.
3. A. Maeda, "Technology innovations for smart cities," *VLSI Symp. Circuits*, June 2012, pp. 6-9.
4. N. Verma, K. H. Lee, and A. Shoeb, "Data-driven approaches for computation in intelligent biomedical devices: A case study of EEG monitoring for chronic seizure detection," *J. Low Power Electron. Appl.*, vol. 1, no. 1, pp. 150–174, Apr. 2011.
5. J. Park, et al., "A 92mW real-time traffic sign recognition system with robust illumination adaptation and support vector machine," *IEEE J. Solid State Circuits,* vol. 47, no. 11, pp. 2711-2723, Nov. 2012.
6. N. Verma, A. Shoeb, J. Bohorquez, J. Dawson, J. Guttag, and A. P. Chandrakasan, "A micropower EEG acquisition SoC with integrated feature extraction processor for a chronic seizure detection system," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 804–816, Apr. 2010.
7. A. S. Ganapathi, "Predicting and optimizing system utilization and performance via statistical machine learning," Ph.D. dissertation, Univ. Cal., Berkeley, CA, 2009.
8. Y. Ge and Q. Qui, "Dynamic thermal management for multimedia applications using machine learning" *DAC*, Jun. 2011, pp. 95-100.
9. A. DeOrio, et al., "Machine learning-based anomaly detection for post-silicon bug diagnosis," *DATE,* March 2013.
10. P. Dubey, "Recognition, mining and synthesis moves computers to the era of tera," *Technology at Intel Magazine*, Feb. 2005.
11. C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks* vol.13, no.2, pp. 415–425, Mar. 2002.
12. K. H. Lee, S.-Y. Kung, and N. Verma, "Improving kernel-energy tradeoffs for machine learning in implantable and wearable biomedical applications," *ICASSP*, May 2011, pp. 1597–1600.
13. D. Anguita, A. Boni, and S. Ridella, "A digital architecture for support vector machine: theory, algorithm, and FPGA implementation," *IEEE Tran. Neural Networks*, vol 14, no. 5, pp. 993-1009, Sep. 2003.
14. R. E. Schapire and Y. Freund *Boosting: Foundations and Algorithms*, Cambridge, MA, USA: MIT Press, 2012.
15. E. I. Shih, "Reducing the computational demands of medical monitoring classifiers by examining less data," Ph.D. dissertation, MIT, Cambridge, MA, 2010.
16. F. M. Khan, M. G. Arnold, and W. M. Pottenger. "Finite precision analysis of support vector machine classification in logarithmic number systems," *IEEE EUROMICRO Symp.Digital System Design (DSD)*, pp. 254-261, Sept. 2004.
17. K. H. Lee, S.-Y. Kung, and N. Verma, "Low-energy Formulations of Support Vector Machine Kernel Functions for Biomedical Sensor Applications" *J. Sign. Process. Syst.*, Apr. 2012. DOI 10.1007/s11265-012-0672-8.
18. K. H. Lee and N. Verma, "A 1.2–0.55 V general-purpose biomedical processor with configurable machine-learning accelerators for high-order, patient-adaptive monitoring," *ESSCIRC*, 2012, pp. 285–288.
19. K. Brinker, "Incorporating diversity in active learning with support vector machines," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE*, vol. 20, no. 1, 2003, p. 59.
20. Y. Yetim, M. Martonosi, and S. Malik, "Extracting useful computation from error-prone processors for streaming applications", *DATE*, March 2013.
21. L. Leem, et al., "Error-resilient system architecture for probabilistic applications," *IEEE/ACM DATE*, Mar. 2010.
22. J. Sartori and R. Kumar, "Architecting Processors to Allow Voltage/Reliability Tradeoffs," *CASES*, Oct. 2011.
23. N. Shanbhag, et al., "Stochastic computation," *DAC*, June 2010.
24. N. Verma, et al., "Enabling system-level platform resilience through embedded data-driven inference capabilities in electronic devices," *ICASSP*, March 2012.
25. A. Shoeb, et al., "Application of machine learning to epileptic seizure detection," *ICML*, June 2010.
26. E. Ubeyli, "ECG beats classification using multiclass support vector machines with error correcting output codes," *Digital Signal Processing*, May 2007.
27. PhysioNet. *http://www.physionet.org.*