Shannon-inspired Statistical Computing for the Nanoscale Era

Naresh R. Shanbhag, Naveen Verma, Yongjune Kim, Ameya D. Patil, and Lav R. Varshney

***Abstract*—Modern day computing systems are based on the von Neumann architecture proposed in 1945 but face dual challenges of: 1) unique data-centric requirements of emerging applications and 2) increased non-determinism of nanoscale technologies caused by process variations and failures. This paper presents a Shannon-inspired statistical model of computation (*statistical computing*) that addresses the statistical attributes of both emerging cognitive workloads and nanoscale fabrics within a common framework. Statistical computing is a principled approach to the design of *non* von Neumann architectures. It emphasizes the use of information-based metrics; enables the determination of fundamental limits on energy, latency, and accuracy; guides the exploration of statistical design principles for low signal-to-noise ratio (SNR) circuit fabrics and architectures such as *deep in-memory architecture* (DIMA) and *deep in-sensor architecture* (DISA); and thereby provides a framework for the design of computing systems that approach the limits of energy efficiency, latency, and accuracy. From its early origins, Shannon-inspired statistical computing has grown into a concrete design framework validated extensively via both theory and laboratory prototypes in both CMOS and beyond. The framework continues to grow at both of these levels, yielding new ways of connecting systems through architectures, circuits, and devices, for the semiconductor roadmap to march into the nanoscale era.**

***Index Terms*—computing, statistical computing, nanoscale devices, information theory, artificial intelligence, machine learning**

I. Introduction

The invention of the Turing machine [1] (1937) and its realization via the von Neumann architecture [2] (1945) helped spawn the modern information age. The discovery of the transistor by Bardeen, Brattain, and Shockley (1948), and subsequently, the integrated circuit by Kilby (1954) provided an efficient and cost-effective substrate to realize complex computing systems. Since then, computing systems, through various generations of microarchitecture and semiconductor technology (Moore's Law [3] (1965)) have fueled growth in nearly all industries and transformed health, energy, transportation, education, finance, entertainment, and leisure.

Computing systems, however, now face two major challenges. The first is the emergence of non-determinism in nanoscale fabrics due to increased stochasticity (variations and failures) as well as diminishing energy-delay benefits via CMOS scaling. Stochasticity in nanoscale fabrics is contrary to the deterministic Boolean switch required by the traditional von Neumann architecture. While many beyond-CMOS devices have been invented, none surpass the silicon MOSFET under metrics for a deterministic switch [4], [5]. The second is the emergence of data-centric cognitive applications that emphasize exploration-, learning-, and association-based computations using statistical representations and processing of massive data volumes [6]–[8]. The data-centric nature of these applications imposes stark challenges at the system-platform level. For instance, it places an undue burden on the memory-processor interface, aggravating the *memory wall* problem [9] in von Neumann architectures. Similarly, it gives rise to the sensor-processor interface, i.e., the *sensory wall*, in systems emphasizing large-scale sensing (e.g., IoT devices).

Although machines have recently begun to approach and exceed human performance in many complex inference tasks—AlexNet [10] and ResNet [11] achieving human-level accuracy in recognition tasks; AlphaGo [12] beating human champions in Go—due to the above-mentioned challenges, these successes have used energy four-orders-of-magnitude more than the human brain. Thus, we face a fundamental question:

> *How can we design intelligent machines that can proactively interpret and learn from data, solve unfamiliar problems using what has been learned, while operating with the energy efficiency of the human brain?*

This question can be comprehensively answered only by revisiting the very foundations of computing in light of the complexities of today's data-centric nanoscale era. This paper introduces Shannon-inspired statistical computing, based on the Shannon theory of communications [13], [14], as an outcome of such questioning undertaken by a community of researchers spanning systems, architectures, circuits, and devices.[1]

Shannon-inspired statistical computing, also referred to as *statistical computing* herein, is a principled approach to *non* von Neumann computing. It emphasizes: the use of information-based design metrics; enables the determination of fundamental limits on energy, latency, and accuracy; motivates and guides the exploration of statistical design principles and low signal-to-noise ratio (SNR) circuit fabrics and architectures; and thereby provides a methodology for the design of computing systems that approach the limits of energy efficiency, latency, and accuracy. Statistical computing is very

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

N. R. Shanbhag, Y. Kim, A. D. Patil, and L. R. Varshney are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (email: {shanbhag, yongjune, adpatil2, varshney}@illinois.edu).

N. Verma is with the Department of Electrical Engineering at Princeton University, Princeton, NJ 08544 USA (e-mail: nverma@princeton.edu).

[1] https://www.sonic-center.org/

well-suited to meet the needs of the data-centric nanoscale era as it is now well-recognized that statistical descriptors are most appropriate for both application data and nanoscale fabrics. Thus, statistical computing provides a consistent framework to jointly address design issues spanning from systems to devices. From its early origins [15], Shannon-inspired statistical computing has grown into a comprehensive design framework validated in both theory [16]–[20] and laboratory prototypes in both CMOS [21]–[29] and beyond [30], [31]. This paper provides an overview of the Shannon-inspired statistical computing framework.

A. Related Work

Moving beyond the deterministic von Neumann model of computation has been a long-standing aspiration in computing. Von Neumann himself [32] advocated the need to investigate a *stochastic* Shannon-inspired approach for designing computing systems on unreliable circuit/device fabrics, obtaining upper bounds on the error probability ϵ of an individual gate (ϵ-noisy logic gates) in order for a Boolean network composed of such gates to have an output probability of error less than 0.5. This work inspired a series of papers by information theorists [33]–[36] tightening the upper bound and proposing design methods based on expensive N-way gate-level replication. Information-theoretic results on noisy Boolean networks [33], [34], [37] have been largely negative, e.g., Elias [33] showed that the Shannon capacity for an ϵ-noisy AND gate is zero. We believe these negative results are primarily due to the restrictive definition of computation as a Boolean network. Shannon-inspired statistical computing avoids these pitfalls by defining computation more broadly as evaluating a function $f(X;\eta)$ of input X in the presence of noise in nanoscale fabrics η.

Other approaches such as fault-tolerant computing [38]–[40], approximate computing [41], [42], error-resilient computing [43]–[45], neuromorphic computing [46], stochastic computing [47], [48], and probabilistic computing [49] have also addressed the problem of designing reliable systems using unreliable components in different ways. These either exploit the inherent error tolerance of data-centric workloads to accommodate computational errors or employ expensive replication/replay techniques for compensating errors. For example, approximate computing [41], [42] employs approximation techniques to reduce the algorithmic computational complexity while meeting the application-level accuracy requirements on deterministic fabrics. While such approximations have been standard in the design of VLSI communication and signal processing systems, approximate computing makes their use pervasive across the various layers of design abstraction from the circuit level, to architecture, and even software. Similarly, neuromorphic computing with roots in the pioneering work of Carver Mead [46], strives to realize reliable computing systems via a bottom-up emulation of biological systems, e.g., the human brain, so that computational errors fall within the error tolerance envelope of the application. Classical fault-tolerant [32], [38]–[40] and error-resilient computing techniques such as RAZOR [43] and error detection sequential (EDS) circuits [44] employ replication or replay to compensate for errors. Such techniques can compensate for very small error rates, e.g., RAZOR I [50] (RAZOR II [51]) operates at an error rate of 0.1% (0.04%) while achieving an energy savings of 15% (5%), while EDS detection achieves 7% energy reduction over equivalent deterministic systems, i.e., systems operating at the point of first failure (PoFF).

None of the above-mentioned techniques leverage the unique opportunity to process *information* resident in the application data. Indeed, in establishing Shannon-inspired statistical computing, we make a critical distinction between *exploiting an application's tolerance to errors* versus *processing of an application's statistical information*. Exploiting an algorithm's error tolerance simply pits the widening of data distributions due to errors, against existing system margins. Statistical computing, on the other hand, engineers the impact of errors on these distributions to focus on preserving the information relevant to the task at hand.

Given this fact, we believe von Neumann's aspiration [32] to develop a Shannon-inspired framework for designing reliable systems using unreliable components has remained elusive; that Shannon theory [13], in spite of its promise, has had minimal impact on the design of reliable computing systems thus far; and a comprehensive computing framework based on Shannon-inspired mapping of applications from systems-to-devices is lacking. This paper strives to address this long-standing aspiration.

In the next section, we provide the background on Shannon theory. Section III describes the Shannon-inspired statistical computing framework. Statistical computing design techniques such as data-driven hardware resilience (DDHR), statistical error compensation (SEC), and hyper-dimensional (HD) computing are presented in Section IV. Shannon-inspired architectures such as the deep-in memory architecture (DIMA) (Section V-A) and deep-in sensor architecture (DISA) (Section V-B) are described next. Section VI concludes with a future outlook for computing in the nanoscale era.

II. Background on Shannon Theory

Shannon's 1948 paper [13] established information as a statistical quantity and laid out a complete mathematical theory of communication. That paper defined information as a statistical quantity and channel capacity as a function of noise statistics. The central result was that *reliable communication (error probability approaching zero) can be achieved only if the information transmission rate is less than the channel capacity*. Prior to Shannon's work, it was commonly believed that approaching zero error probability required a large (potentially infinite) number of repeated transmissions of the same information symbol (repetition code) which forced the information rate (rate at which information symbols are transmitted) also to approach zero or to transmit large signal power to overwhelm the receiver noise (high SNR). Note that the commonly used techniques of replication and replay strategies in fault-tolerant and error-resilient computing frameworks and the use of deterministic circuit fabrics are equivalent to the use of repetition codes and high-SNR channels, respectively,

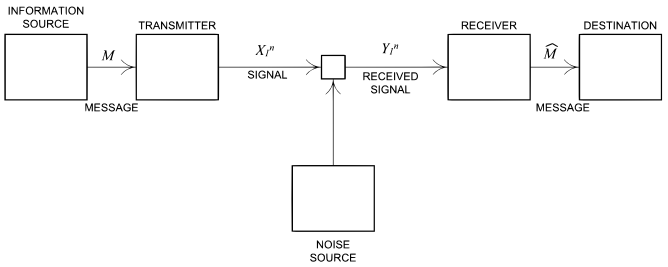


Fig. 1. Block diagram of a general communication system [13].

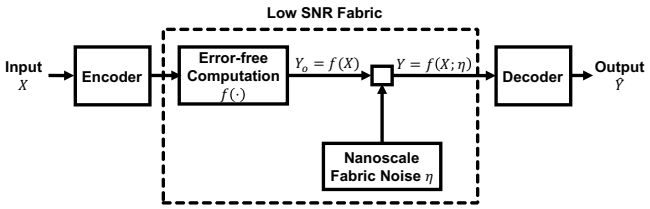


Fig. 2. The Shannon-inspired statistical model of computation.

from the pre-Shannon era. However, concurrent error detecting codes [44] can be considered to be more in the spirit of post-Shannon era error-control codes. Shannon also showed that capacity-achieving error control codes exist. Indeed, the discovery of turbo codes [52] and the re-discovery of low-density parity-check (LDPC) codes [53], [54] in the 1990s have all but eliminated the gap to Shannon capacity for point-to-point links. In this manner, Shannon theory completely revolutionized communications. Shannon-inspired statistical computing is based on a premise that a similar transformation is also possible in computing.

Consider the basic block diagram of a communication system, an annotated version of [13, Fig. 1] shown as Fig. 1. The goal is to reproduce the message M as $\widehat{M}$ with arbitrarily small probability of error $p_e = \Pr\{M \neq \widehat{M}\}$ by encoding M into the transmitted sequence $X_1^n = (X_1, \ldots, X_n)$ and decoding $\widehat{M}$ from the channel output sequence $Y_1^n = (Y_1, \ldots, Y_n)$. The *mutual information* (MI) between random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ with joint probability density function $p_{X,Y}(x,y)$ and corresponding marginals $p_X(x)$ and $p_Y(y)$ is given by

$$I(X;Y) = \iint p_{X,Y}(x,y) \log_2 \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} dxdy. \quad (1)$$

This can be used to define a quantity for memoryless channels called *channel capacity* as follows:

$$C(B) = \sup_{p_X(x): E[b(X)] \leq B} I(X;Y), \quad (2)$$

where $b(\cdot)$ is a single-letter function to measure resources and B is an average resource constraint.

By using a random coding argument (i.e. constructing a codebook at random and measuring average performance) for the achievability part, Shannon showed that there exists at least one code with normalized log codebook size (i.e., code rate) less than channel capacity (but perhaps asymptotically long in length) with arbitrarily small error probability.

The converse argument for finite alphabets $\mathcal{X}$ and $\mathcal{Y}$ relates the operational notion of error probability p_e to the MI via the conditional entropy. In particular, Fano's inequality [55] states that:

$$H(M|\widehat{M}) \leq h_b(p_e) + p_e \log(|\mathcal{X}| - 1) \quad (3)$$

where $h_b(p) = -p\log_2(p) - (1-p)\log_2(p)$ is the binary entropy function and $H(M|\widehat{M}) = H(M) - I(M;\widehat{M})$ where $H(M)$ is the source entropy. The final result of the converse argument is that error probability may not be arbitrarily small if code rate exceeds the channel capacity.

Unexpectedly, the achievability and converse results match and so channel capacity is precisely the fundamental limit of reliable communication over a noisy channel. Moreover, the MI emerges as the natural measure for point-to-point communication. Further, the extremizing $p_X(x)$ in (2) is in fact a description of an optimal codebook, i.e., it provides an optimal design principle for reliable communication.

III. Shannon-inspired Statistical Computing Framework and Fundamental Limits

In this section, we attempt provide a consistent framework to address design issues spanning from systems to devices. We rely on information-based metrics to address both the statistical attributes of application data as well as the statistical behavior of nanoscale fabrics and to obtain fundamental limits on system resources.

A. Shannon-inspired Statistical Computing

The Shannon-inspired model of computation, Fig. 2, comprises an *encoder*, a noise-free computation of the *correct output* $Y_o = f(X)$ over an ideal deterministic fabric being corrupted by noise in nanoscale fabrics parametrized by variable η to generate the *observed output* $Y = f(X;\eta)$ of the *low SNR fabric* (the channel), followed by a *decoder* that recovers the corrected output $\widehat{Y}$. In practice, Y_o is unavailable and may even be unknown. In Fig. 2, all variables (X, Y_o, η, Y, $\widehat{Y}$) are random variables.

Statistical computing is concerned with two primary information-based metrics: 1) $I(Y_o;\widehat{Y})$: the MI between the correct output and the final output, and (ii) $\Pr\{\widehat{Y} \neq Y_o\}$: the final application-level error probability. While the end user primarily cares about the application-level accuracy, the MI is a fundamental metric quantifying the uncertainty reduction of Y_o by knowing $\widehat{Y}$. These metrics emphasize that reliable computing is focused on maximizing the statistical dependence between the final output $\widehat{Y}$ and the correct output Y_o and not with the input X as in communications. Unlike communications, where the intent is to recover the transmitted data X, in computing, we care about the correct output Y_o. Furthermore, as $I(Y_o;\widehat{Y}) = H(Y_o) - H(Y_o \mid \widehat{Y})$, $I(Y_o;\widehat{Y})$ is maximized when $H(Y_o \mid \widehat{Y}) = 0$ implying that, in the best case, the knowledge of $\widehat{Y}$ completely removes any additional uncertainty in knowing Y_o other than its intrinsic entropy. Furthermore, for a given value of $I(Y_o;\widehat{Y})$ (equivalently for

a given $H(Y_o \mid \widehat{Y})$), Fano's inequality in (3) provides a lower bound on $\Pr\{\widehat{Y} \neq Y_o\}$.

Hence, the objective of Shannon-inspired statistical computing is to design computing systems where: (1) the MI $I(Y_o; \widehat{Y})$ is maximized in the presence of noise in nanoscale fabrics η; and (2) the application-level error probability $P_e = \Pr\{\widehat{Y} \neq Y_o\}$ is minimized for a given $I(Y_o; \widehat{Y})$. This objective is achieved by applying one or more of the following principles derived from communication systems:

- *Encoding*: induce a known structure into input data before computation on a low SNR nanoscale fabric to make it easier to recover from errors.
- *Channel engineering*: shape the impact of η on the observed output Y by engineering the error statistics of the low SNR fabric.
- *Decoding*: employ statistical estimation and detection techniques to recover $\widehat{Y}$ from Y and the knowledge of noise statistics, if available.

In this manner, Shannon-inspired statistical computing fundamentally optimizes systems, circuits, and nanoscale devices to preserve application-relevant information content at the output.

B. Fundamental Limits

Computational devices and components are fundamentally stochastic, but their level of noise is often governed by basic system resources such as energy, latency, and volume; using more resources typically reduces uncertainty. For example, thermodynamic arguments suggest that the error probability ϵ of an electronic switch decreases exponentially in energy [56]. For spintronic devices, physics arguments lead to a stretched exponential relationship [57]. The basic question in statistical computing is how to assemble and configure such stochastic components into circuits and systems that can achieve reliable computation—under appropriate performance criteria such as application-level error probability—in a way that uses resources in the most efficient manner possible.

Shannon-inspired statistical computing aims to establish fundamental limit theorems for relationships between resource consumption and application-level performance. This enables several outcomes. First, it establishes the playing field in terms of which resources and performance criteria are fundamental and which are largely unimportant. Second, it provides fundamental benchmarks that allow an evaluation of new technologies on an absolute scale, rather than only compared to previous technologies. Third, it provides ideals for pushing researchers to build computing systems that approach/achieve these limits. Fourth, it specifies desirable properties of nanoscale devices and provides a way to parameterize systems-level fundamental limits based on physical limits of nanodevices. Last and perhaps most importantly, in delineating what is possible from what is impossible, fundamental limits provide insights into operating at the boundary, i.e., optimal design techniques for statistical computing. The effectiveness of these design techniques in reducing energy, enhancing robustness, and increasing functional density can then be demonstrated via experimental realizations.

Next we will see two exemplary computing settings with fundamental limits and optimal designs. Separately note that analogies between communication and computation systems can also yield limit theorems for broad design techniques such as statistical error compensation [58].

1) Information-theoretic Limits of Data Storage in Emerging Memory Systems: Memory storage and recall aim to be an identity computation function (i.e., $f(X) = X$) such that what is stored is what is later retrieved, but due to noise this may be difficult. Recent experimental work [59] has explicitly characterized the stochastic distribution of phase change memories. Further, making direct application of Shannon's channel coding theorem and his end-to-end transmission theorem led to fundamental limits on storage capacity by maximizing MI. This yielded optimal energy-efficient analog signaling schemes that both approach the Shannon limit and also increase data storage capacity by 30% as compared to digital schemes.

Yet, more resources than just the storage elements themselves are needed to build memory systems, as well-noted in neurobiology [60]. In particular, computational elements that accompany the storage elements require energy and volume. As such, one can consider the problem of constructing a reliable memory system out of unreliable components (both storage elements and logic gates) where a new fundamental limit that explicitly takes the cost of these components into account is established. To determine fundamental limits for such *storage capacity*, $\mathfrak{C}$, we can develop an order-optimal achievable scheme based on a construction using LDPC codes and a noisy message-passing decoder [61], [62], and a converse argument that makes use of MI [18]. In particular, sphere-packing entropy production/dissipation arguments yield an upper bound on storage capacity of memories constructed from α-noisy gates and β-noisy bit-cells,

$$\mathfrak{C}(\alpha, \beta) \leq \frac{C(\alpha)}{1 + \frac{h_b(\alpha)}{2 - h_b(\beta/2 + 1/4)}} \tag{4}$$

where $C(\alpha) = \max_{p_X(x)} I(X;Y)$ is the Shannon capacity of the bit-cell. The basic design insight emerging from optimizing fundamental limits is in how to allocate resources to gates and bit-cells, respectively.

A related analysis can be used to find optimal resource allocation schemes for memory systems with missing wiring due to process variation [63]; intriguingly, a little bit of structural noise can improve performance, so-called *stochastic facilitation* [64].

2) Energy-Reliability Limits in Nanoscale Boolean Circuits and Deep Neural Networks: Consider the problem posed by von Neumann [32], i.e., a Boolean network constructed from ϵ-noisy gates with one difference being that ϵ is a function of energy; the more energy allocated to a given gate, the less the error. We wish to determine fundamental tradeoffs between energy and reliability at the circuit level as a function of the gate-level noise-energy relationship. As such, two interrelated problems of circuit redundancy and energy allocation must be solved simultaneously to find optimal energy-reliability tradeoffs. This is difficult to do directly, but we can establish information-theoretic limits that can then be optimized as a design principle [19] by using Fano's inequality and

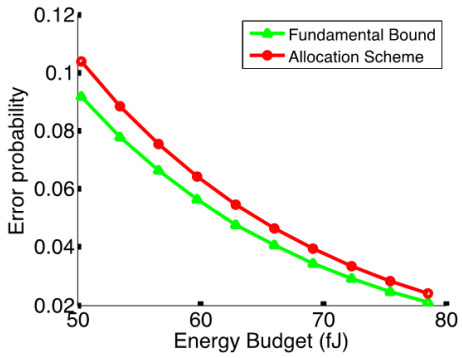


Fig. 3. Comparison of novel information-theoretic bound on energy-reliability tradeoff for disjunction of four Boolean variables and practical energy allocation to a Boolean circuit made from spin devices under the physical model of [70].

a strong data processing inequality that quantifies the loss in MI through multiple stages of computation. Let us first specifically consider tree-structured circuits that have fanout of one, i.e., what are called *formulas* in the fault-tolerant computing literature and are the main object of study there, e.g., [34], [35], [65]–[67], though this may increase circuit complexity. The basic result is that heterogeneous energy allocation is fundamentally superior to homogeneous energy allocation in terms of the scaling law for minimum energy per input bit of a Boolean computation while meeting any reasonable system-level reliability requirement. In fact, the optimal scaling behavior in the heterogeneous case is linear per input bit, just like for noiseless circuits! Furthermore, the use of Karush-Kuhn-Tucker (KKT) conditions from information-theoretic bounds gives the following design principle — if ϵ decays exponentially with energy, gates closer to the output should receive more energy in a geometric manner as one goes from one layer to the next (Fig. 3) [19]. This is a design principle that can be implemented in practice.

We extend our approach from tree-structured circuits to directed acyclic graph structures through a significant combinatorial extension of the basic mutual information propagation idea. Similar results can be derived for feedforward neural networks (multilayer perceptrons) where it can be shown that optimal energy allocation that assigns more energy to neurons closer to the output [68]. Interestingly, this is how energy allocation in mammalian sensory cortex seems to be [69].

Just as Shannon theory established the channel capacity as the fundamental limit for communications and suggested coding as a practical design technique to approach those limits, similarly, we next describe practical Shannon-inspired design techniques to build computing systems approaching these fundamental limits.

IV. Shannon-inspired Statistical Design Techniques

While Section III-A established the role of MI $I(Y_o;\widehat{Y})$ (see Fig. 2) in designing reliable systems and in obtaining fundamental bounds on energy and reliability, in this section, we present statistical design techniques for maximizing $I(Y_o;\widehat{Y})$.

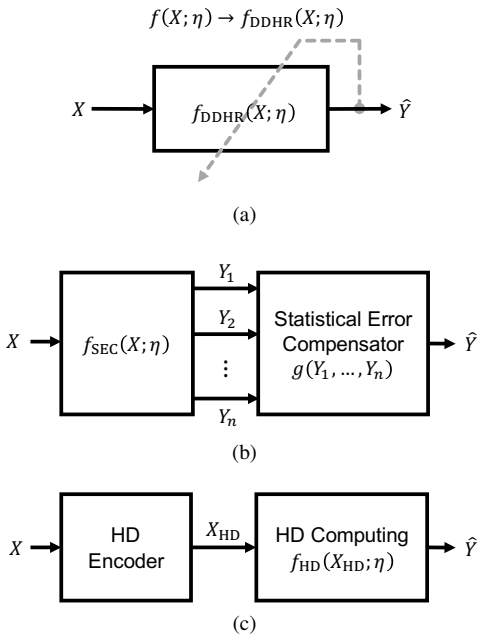


Fig. 4. Shannon-inspired statistical design techniques: (a) data-driven hardware resilience (DDHR), (b) statistical error compensation (SEC), and (c) high-dimensional (HD) computing.

Thus far, three distinct techniques have been discovered, and are shown in Fig. 4:

- *Data-driven hardware resilience (DDHR)* [24], [25]: engineers the error statistics of the low SNR fabric by adapting $f(X;\eta)$ into a new function $f_{\text{DDHR}}(X;\eta)$ such that the MI $I(Y_o;\widehat{Y})$ is maximized.
- *Statistical error compensation (SEC)* [16], [17], [21]–[23]: engineers the error statistics of the low SNR fabric as seen at the output Y to enhance $I(Y_o;Y)$ so that a low-complexity statistical error compensator $g(\cdot)$ (decoder) can be devised to correct for errors and thereby maximize $I(Y_o;\widehat{Y})$.
- *Hyper-dimensional (HD) computing* [20], [71]: uses a random code to map the input to a high-dimensional space and implement an associative algebra to obtain $\widehat{Y}$ thereby also maximizing $I(Y_o;\widehat{Y})$. Note: Shannon theory already proved that a high-dimensional random code is a good (capacity achieving) code in communications. HD computing overlays an associative algebra on top and directly computes in the HD space.

In all three cases, the MI $I(Y_o;\widehat{Y})$ is maximized.

A. Data-Driven Hardware Resilience (DDHR)

DDHR views the output distributions from a system block as originating, not only due to variances in the application-level signals, but also due to variances arising from stochastic noise of the nanoscale fabric [25]. For a binary inference problem, suppose that $Y_o \in \{+1,-1\}$ denotes the class label and the predicted output is $\widehat{Y} = f(X)$. The objective of training is to determine $f(\cdot)$ such that

$$f(\cdot) = \arg\min_{f(\cdot)} L(Y_o, f(X)) \tag{5}$$

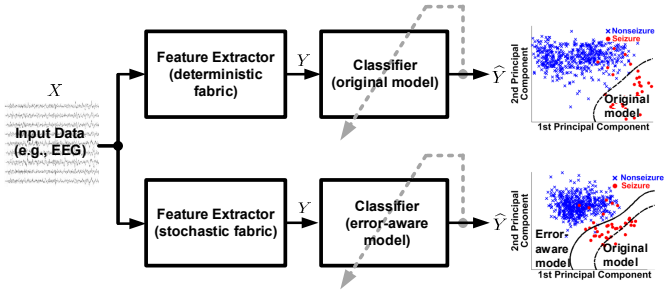


Fig. 5. Illustration of DDHR, using example of EEG-based seizure detector [25].

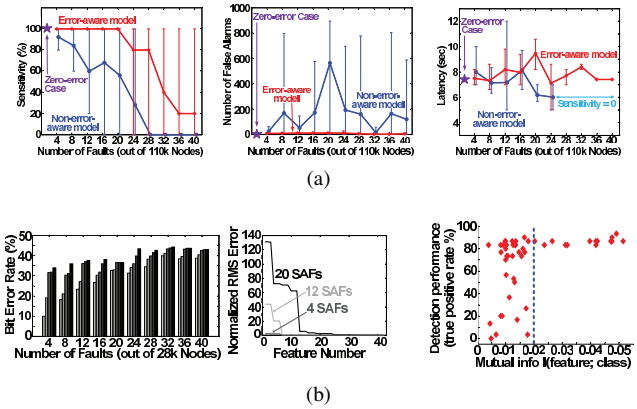


Fig. 6. Representative results from EEG-based seizure detector employing DDHR [25]: (a) performance restoration with error-aware model (error bars show minimum/maximum performance over 5 randomized mappings of faults at each rate), and (b) analysis of bit-level error rates, error magnitudes (measured as RMS error normalized to true feature value), and system performance (measured as detection sensitivity with respect to MI).

where $f(\cdot)$ depends on the statistics of the training data (X, Y_o), $L(\cdot)$ represents the loss function, which depends on the machine learning task. DDHR attempts to find $f_{\text{DDHR}}(\cdot)$ such that

$$f_{\text{DDHR}}(\cdot) = \arg\min_{f(\cdot)} L(Y_o, f(X;\eta)) \tag{6}$$

where $f_{\text{DDHR}}(\cdot)$ is adapted to the statistical attributes of data (X, Y_o) and that of the nanoscale fabric η as shown in Fig. 4(a). Next, we present two different scenarios for the application of DDHR.

Figure 5 illustrates the principle of DDHR via a canonical inference system, consisting of feature-extraction and classification stages. We assume that the feature-extraction stage is implemented on a stochastic fabric, and the classification stage is implemented on an ideal (deterministic) fabric. It is the job of feature-extraction to form data distributions that are well-separated with respect to a targeted inference, in order to enhance the generalization behavior of the learned classification model. The data distributions in Fig. 5 are derived from an actual hardware experiment described below, and show that, in the presence of errors, the distributions can be substantially altered. However, by employing data from the error-affected distributions, a new classification model, referred to as an *error-aware model*, can be learned using machine learning algorithms. So long as the resulting distributions maintain separation, inference performance can thus be restored to ideal levels.

To be more quantitative, we employed an FPGA platform, to which feature-extraction stages from a number of inference systems were mapped, and hardware faults were emulated by modifying the gate-level netlist to introduce static stuck-at-faults. This enabled controllable fault rate, and randomized mappings of faults to netlist nodes at each rate, since the ultimate error statistics are strongly dependent on precisely which nodes are affected by faults. Figure 6(a) shows representative results from an electroencephalogram (EEG)-based seizure detector (where features corresponding to the spectral-energy distribution of each EEG channel are extracted using FIR filters followed by energy accumulators [25]). We see that inference accuracy degrades without an error-aware model in spite of the presence of an inherent application-level error tolerance. But, with an error-aware model, accuracy is consistently restored even at high fault rates (i.e., no accuracy degradation is observed with faults on up to 18% of the feature-extractor circuit nodes, across 5 net lists with randomized fault injection). To understand this, Fig. 6(b) shows, first the resulting bit-level error rates and magnitudes, for different fault mappings, and then the system performance (measured as detection sensitivity) versus the empirical MI between feature vectors Y and class membership Y_o. Three important insights emerge: (1) even modest faults in the fabric can cause substantial bit-level errors, limiting the leverage of inherent application-level error tolerance; (2) on the other hand, even with significant fault rates, the required MI can often be retained in such inference applications; and (3) an error-aware model obtained by training to the error-affected distributions from faulty fabrics, enables performance corresponding to the MI.

Relating the emergent insights with system-level implications for adopting DDHR, two overheads can be considered. First is the retraining cost of constructing an error aware model. This is typically predictable for a particular application and can leverage techniques to enhance training efficiency, given an initial model (active, transfer learning) [25]. Second is the potential for increased complexity of the error-aware model. This is less predictable, dependent on the original and error-affected distributions, as well as the ability of the classifier model to adapt to different distributions. Previous experimental work has shown that the complexity may be impacted, but is not generally expected to increase [72].

Initially, it may seem that DDHR can only address errors in stages preceding the classification stage. But in fact, DDHR can be applied to the classification stage itself, allowing the classifier to also be implemented on stochastic fabrics. Further, adaptation to error-affected data distributions *after* the error-affected stage limits performance to a level corresponding to the MI retained by the error-affected stage. However, adaptation to an error-aware model *within* the error-affected stage itself raises the possibility of actively opposing the loss in MI due to those errors. A practical approach for doing this is error-adaptive classifier boosting (EACB) [24], illustrated in Fig. 7. EACB exploits iterative training of weak classifiers,

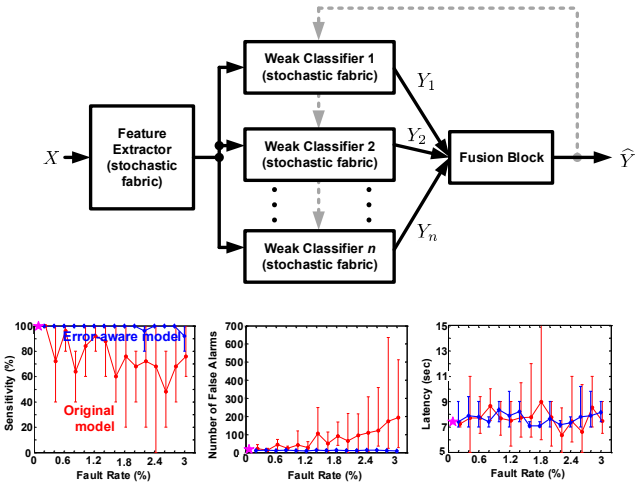


Fig. 7. Illustration of EACB and results from emulation of EEG-based seizure detector [24].

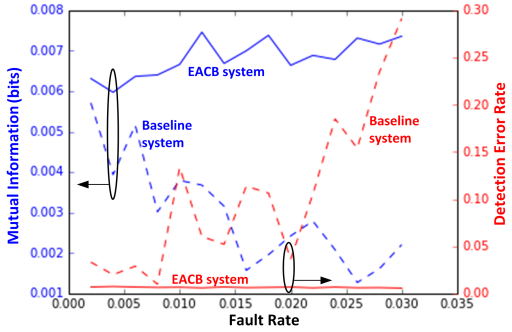


Fig. 8. Comparison of MI and detection error rate from FPGA emulation of a seizure-detection system.

as in adaptive boosting (AdaBoost) [73], but where model training at each stage is biased by both fitting errors from previous iterations and errors due to the stochastic fabric (i.e., no accuracy degradation is observed with faults on up to 2.8% of the classifier circuit nodes, across 5 net lists with randomized fault injection). We see that the output information is spread over multiple weak-classifier decisions. The need, potentially, for more such decisions incurs system-resource costs. However, we note that: 1) implementing very weak classifiers, as permitted by AdaBoost, enables energy-aggressive implementations; and 2) data-driven training at each iteration attempts to optimally minimize classification error, and thus maximize MI (Fig. 8). Work surrounding EACB has explored a number of relevant directions, including memory-efficient embedded training of the error-aware model, so that devices may adapt to their own errors [24], and analysis of the ability to overcome stochastic, uncorrelated non-idealities in the fabric (variations) versus systematic, correlated non-idealities (nonlinearity) [74], [75].

Section V provides further details of DDHR and EACB system realizations, and current work focuses on further increasing system scale and complexity of inference applications (particularly moving to realization of large deep-learning systems).

B. Statistical Error Compensation (SEC)

Statistical error compensation (SEC) [21]–[23] realizes a decoder (see Fig. 2) to recover output $\widehat{Y}$ from Y. As the complexity of the decoder is a direct function of the MI $I(Y_o;Y)$, SEC first enhances this MI by obtaining multiple (ideally independent) observations $Y=(Y_1,\ldots,Y_n)$ where $Y_i=Y_o+\eta_i$, and architects the computational fabric so that desirable (information preserving) error statistics are observed in Y. Then, SEC devises a low-complexity decoder $\widehat{Y}=g(Y)$ that strives to compute a maximum a posteriori (MAP) estimate $\widehat{Y}=\arg\max_Y P(Y_o|Y)$ thereby enhancing $I(Y_o;\widehat{Y})$. For independent noise η_is, it is straightforward to show that

$$I(Y_o;Y_i)\leq I(Y_o;\widehat{Y})\leq\sum_{i=1}^{n}I(Y_o,Y_i). \tag{7}$$

In our past experience, the complexity overhead of SEC ranges from 5% to 20% of the deterministic computation $f(X)$ when the low SNR fabric arises from voltage overscaling (VOS) [21] or frequency overscaling (FOS) [22] of digital architectures. However, as SEC allows the SNR of the circuit fabric to be reduced significantly, e.g., error rates of up to 89% can be compensated for in specific tasks, SEC results in significant energy savings.

A number of SEC techniques have been proposed [16], [76]–[78]. These differ in approach taken for generating multiple observations Y_i, methods for shaping error statistics, and decoder designs. For example, algorithmic noise tolerance (ANT) [16] augments the main block $Y_1=f(X;\eta)$ with a low-complexity estimator $Y_2=f_e(X)$ to generate $Y=(Y_1,Y_2)$. Stochastic sensor network-on-chip (SSNOC) [78] decomposes $f(X;\eta)$ into multiple low-complexity approximators realized using different arithmetic architectures. Soft N-modular redundancy (NMR) [76], like NMR, replicates the main block N-times but employs the knowledge of data and error statistics to design a soft voter to compute $\widehat{Y}$. Bit-level likelihood processing (BLP) [77] computes likelihoods of errors using the knowledge of data and error statistics. The effectiveness of SEC techniques has been demonstrated via multiple CMOS IC prototypes [21]–[23] using VOS to lower the circuit SNR and save energy.

Consider ANT [16] (Fig. 9(a)) consisting of a main block implemented on a low SNR fabric which carries out bulk of the computation given by $Y_1=f(X;\eta)$, and an estimator that approximates the error-free function $Y_o=f(X)$ to generate an output $Y_2=f_e(X)$, i.e.,

$$Y_1=Y_o+\eta \tag{8}$$
$$Y_2=Y_o+e \tag{9}$$

where η is the computational error due to unreliable hardware and e is the estimation error as $f_e(X)\neq f(X)$. The estimator is typically designed to be within 5–20% of the main block complexity using statistical estimation techniques, and is realized on deterministic hardware. By forcing the main block and estimator errors η and e, respectively, to emerge from

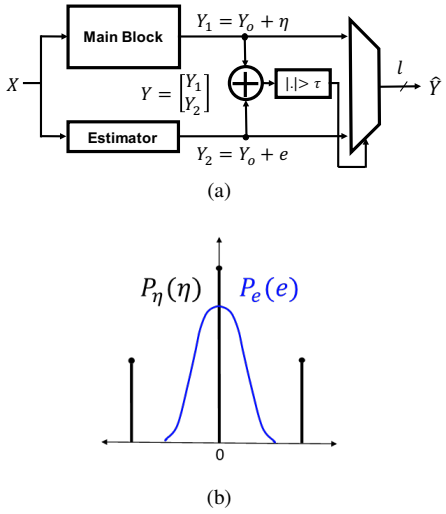


Fig. 9. Algorithmic noise tolerance (ANT): (a) block diagram, (b) distribution disparity in the ANT-based designed, exploited to achieve low-complexity error compensation.

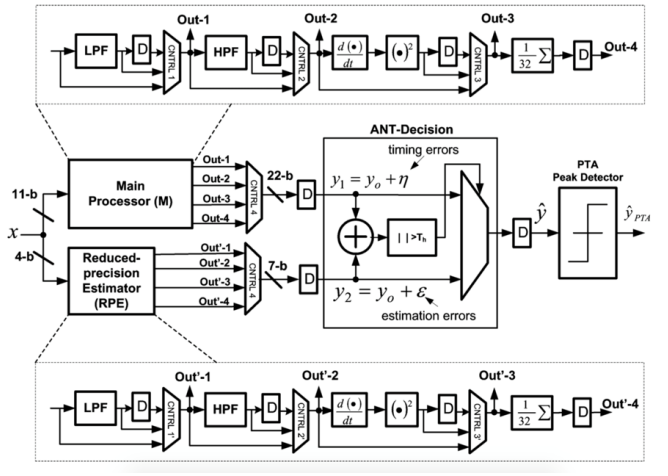
different sources, these errors tend to be *statistically independent* and exhibit *disparate probability density functions*. These two properties result in the following ANT decision rule to approximate a MAP estimator of Y_o using just a multiplexer and an adder:

$$\widehat{Y} = g(Y_1, Y_2) = \begin{cases} Y_1, & \text{if } |Y_1 - Y_2| < \tau, \\ Y_2, & \text{otherwise} \end{cases} \tag{10}$$

where τ is a design-specific parameter chosen to maximize the application-level accuracy $\Pr\{\widehat{Y} \neq Y_o\}$.

Several extensions to the original ANT technique have been proposed [79]–[82]. For example, [79] proposed to reuse some of the intermediate outputs of the main block as estimator outputs, in order to reduce the estimator overhead. [80] proposed a novel fusion block design, which under certain conditions, achieves perfect error compensation. The conditions for perfect error compensation are provided in [80], which guarantee $I(Y_o; Y_1, Y_2) = I(Y_o; \widehat{Y})$ and $I(Y_o; \widehat{Y}) = H(Y_o)$ as discussed in Section III-A.

In [22], an ANT-based subthreshold ECG processor (Fig. 10(a)) implements the Pan-Tompkins algorithm for real-time QRS detection in the ECG waveform. It consists of multiple stages of filtering followed by time-derivative, squaring, and time averaging operations. The ANT technique employs a reduced precision replica (4-bit) of the main block (11-bit) as the estimator. The complexity overhead due to ANT is approximately 32% (RPE+EC in Fig. 10(b)) but is able to compensate for a hardware error rate $\Pr\{\eta \neq 0\} \leq 0.58$ (Fig. 10(c)) while maintaining a detection accuracy $\geq 95\%$. This error compensation capability is $600\times$ higher than that of the conventional system which leads to a $16\times$ reduced sensitivity to voltage variations, which is critical for subthreshold designs. Furthermore, the ability to compensate for a 58% hardware error rate is equivalent to being able to reduce the energy consumption at the minimum energy operating point (MEOP) by 28%. In other words, the energy at MEOP for a deterministic design, which is commonly understood as the



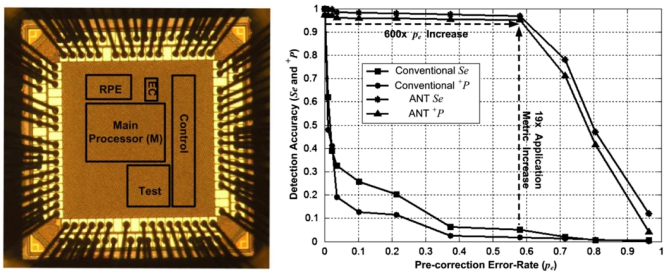


Fig. 10. Shannon-inspired SEC-based sub-threshold ECG processor in 45 nm CMOS [22]: (a) prototype chip architecture, (b) chip microphotograph, and (c) measured accuracy vs. pre-correction error rate for the conventional and Shannon-inspired implementations.

minimum energy required for computation, is reduced further by using ANT on a low SNR fabric.

The use of SEC for beyond CMOS computing was demonstrated in [70] where ANT was applied to an all-spin logic (ASL) implementation of a 120-dimensional linear support vector machine (SVM). As it turns out, ASL gates follow the ϵ-noisy gate model [32], and the tradeoff between energy E_s, delay T_g, and error probability ϵ is well-characterized by the relationship $\epsilon \approx \beta \exp(-\alpha\sqrt{E_s T_g})$ [57], [70]. This inverse relationship between energy-delay and ϵ implies that ASL devices can be made competitive to CMOS only when they are operated at high values of ϵ (e.g., at 1%). By reassigning individual gate delays, the probability distribution of errors η at the output of a 15-b ripple carry adder can be transformed from its naturally occurring dense form to a sparse distribution (Fig. 11(a)) [70]. This delay reassignment when applied to main block of the SVM (Fig. 11(b)) leads to a sparsification of the error distribution at its output. The main block operates at an average error rate of 1%, while the estimator operates much more reliably at an error rate of 10^{-6}. These techniques introduce the much desired disparity in the error distributions of the main block and the estimator, and hence to a simple fusion block consisting of a few adders leading to an overall ANT overhead of 11%. Fig. 11(c) shows that the ANT-based system achieves a decision error rate of $\approx$7% (same as that of the ideal error-free implementation) for average $\epsilon \leq 0.01$ for

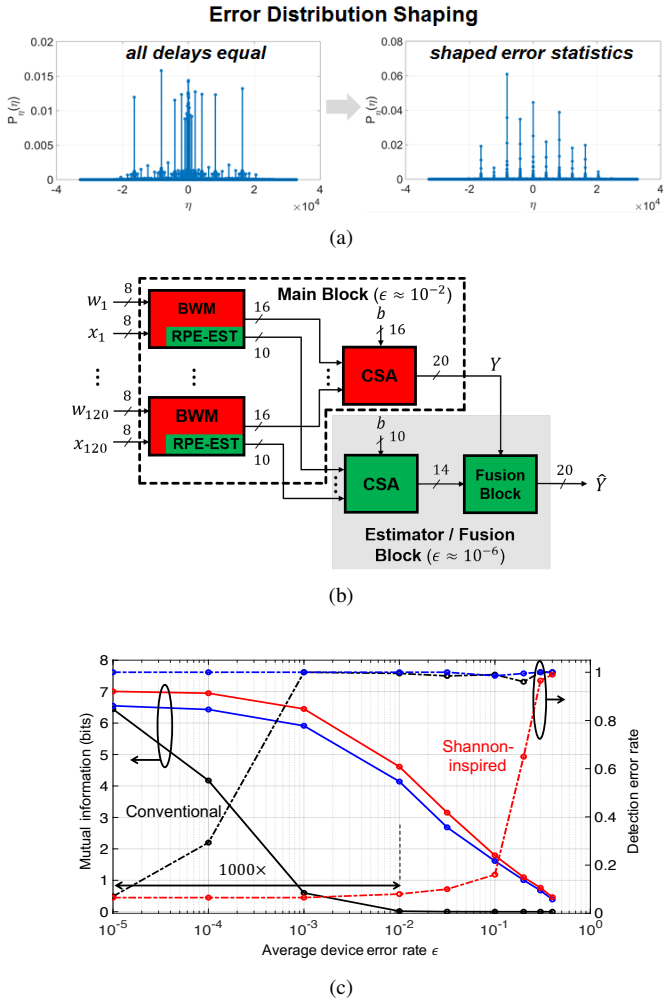


Fig. 11. ANT-based ASL-based linear SVM implementation [70]: (a) shaping the error distribution at the output of a 15-bit ripple carry adder via delay reassignment, (b) SVM architecture with a main block consisting of 120 Baugh-Wooley multipliers (BWM) and a carry save adder (CSA) operating and an estimator embedded into the main block, and (c) enhancement in the MI $I(Y_o;Y)$ at the output of the main block (blue) and the MI $I(Y_o;\widehat{Y})$ at the output of the fusion block (red), as compared to that of the conventional system (black) and the improvement in error compensation capability.

EEG-based seizure detection application [83]. This value of ϵ is $1000\times$ higher than the value of ϵ needed by the conventional architecture to achieve the same decision error rate. The reason for this error compensation capability is the enhancement in the MI of the main block and the fusion block as compared to that of the conventional system.

C. Hyper-dimensional (HD) computing

Kanerva [71] proposed the HD computing framework where binary input vectors are randomly mapped into a HD space, e.g., a k = 8-bit input being mapped to an n = 10,000-bit vector. The key idea underlying this mapping is that a random sampling of vectors from a very high n-dimensional space leads to a concentration of the probability measure, e.g., results in vectors having binomially distributed Hamming weights whose $\frac{\sigma}{\mu} = \sqrt{\frac{1}{n}}$ is small. Such an encoding generates with very high probability vectors that are *typical*, i.e., with a Hamming weight of $0.5n$, which are separate from each other by $0.5n$ bit flips. Furthermore, Kanerva defines an associative algebra involving *local* MAP operations for orthogonalization via local **M**ultiply (XOR), summarization via local **A**dd (OR) and ordered summarization via **P**ermutation. The results of these operations can be accurate even when they are error-prone individually. For example, [20] demonstrates the use of HD computing to identify the language from short strings or sentences with an accuracy of 94% even when the individual MAP computations make errors with probability 2.78×10^{-7}. Thus, similar to the use of random codes in Shannon theory [13], HD computing *encodes* the input message into an random HD vector (see Fig. 4(c)) to enhance robustness of computations, but unlike Shannon theory, HD computing does not decode to generate the input but instead computes an approximate version of the error-free output Y_o in Fig. 2. The dataflow in HD computing is massively parallel due to the local nature of the MAP operations combined with the high-dimensional space in which they are executed. Realizing HD computing systems requires one to efficiently implement the random mapping in the HD code and accommodate its massively parallel dataflow. Recently several HD computing systems have been demonstrated that exploit the inherent random variations in emerging devices such as resistive RAM (RRAM) and carbon nanotube field-effect transistors (CNFETs) to implement the random HD code, and in a monolithic 3D integration platform [84], [85] to exploit the massively parallel nature of HD computations. We point out that the code rate $\frac{k}{n}$ in HD computing is extremely low. This raises the possibility of exploring structured low-rate HD codes and computing with those to enhance the efficiency of HD computing further.

V. Statistical Information-Processing Architectures

The Shannon-inspired design principles and techniques described in Section IV have demonstrated their ability to handle very high error rates (in 10's of %) exhibited by low SNR circuit fabrics, which included circuits with permanent faults and defects [24], [25], circuits operated under VOS or FOS conditions [16], [21], [22], and ϵ-noisy ASL [86]. Armed with these techniques, we ask the question:

> *Are there other low SNR fabrics that exhibit favorable error rate vs. energy-delay trade-offs?*

To answer this question effectively, we ask a surrogate question: *what are the pain points in the today's inference architectures and why do they exist?* The answer to this question is clear—it is the separation ("the Wall") between the data source and computation. Such separation leads to severe communication costs, particularly as the scale of sensing and data storage increase in emerging sensor-rich, data-centric applications. The well-known *memory wall* in the von Neumann architecture and its counterpart, the *sensor wall*, in sensory architectures, which exemplify this separation, arise because data resides in a substrate whose material properties, native devices, and circuit architectures are very different from those for computation. While the memory wall in today's von Neumann architecture refers to the DRAM and CPU interface, we take a more

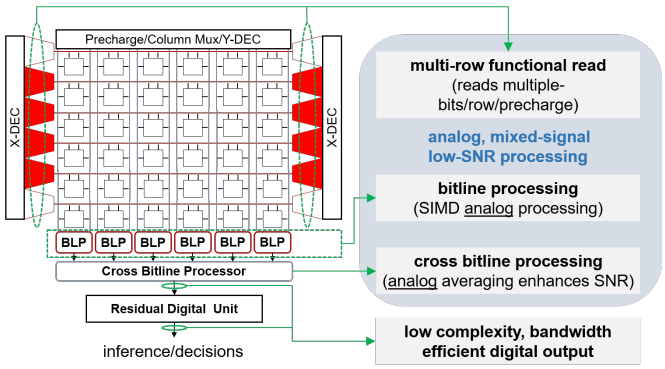


Fig. 12. The deep in-memory architecture (DIMA).

TABLE I
COMPARISON BETWEEN CONVENTIONAL AND DIMA ARCHITECTURES ASSUMING THAT THE DESIRED COMPUTATION REQUIRES ACCESSING A TOTAL OF D BITS OF DATA, THUS REQUIRING A $\sqrt{D} \times \sqrt{D}$ BCA [26].

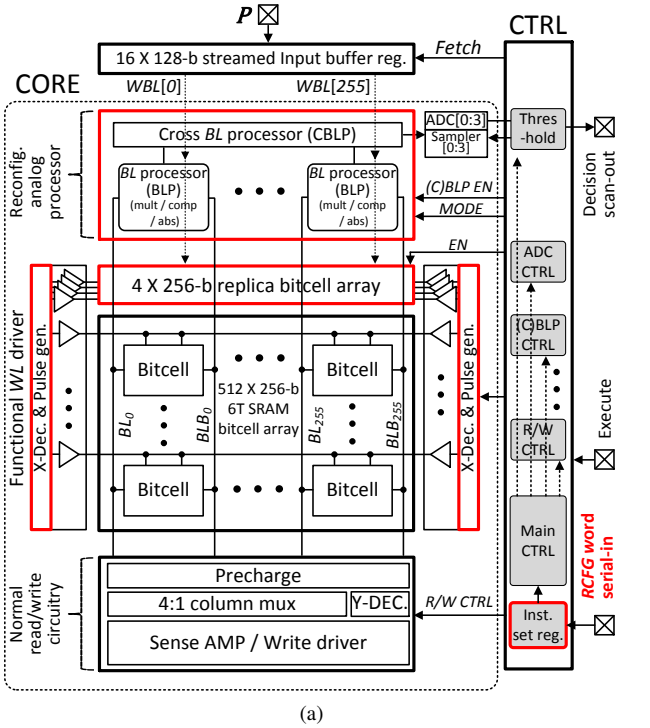
Metric	Conventional	DIMA
Bandwidth	$\frac{1}{\sqrt{D}}$	1
Latency	D	1
Energy	$D^{3/2}$	D
SNR	1	$\frac{1}{\sqrt{D}}$

fundamental view of the problem by observing that this wall exists wherever data is stored in sufficiently large volume separately from where it is processed, including the SRAM and data-path interface. Breaching this generalized notion of the memory wall leads to the *deep in-memory architecture (DIMA)* [26]–[29], [87], [88] and the sensor wall via the *deep in-sensor architecture (DISA)* [30], [31], both of which turn out to be low SNR fabrics and which fall directly into the Shannon-inspired statistical computing framework. This section describes DIMA and DISA.

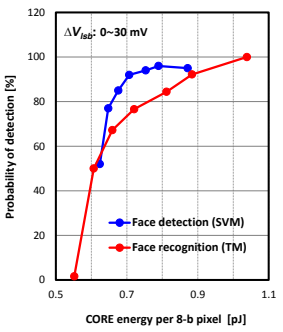
A. Deep In-Memory Architectures (DIMAs)

DIMA [26]–[29], [87], [88] *accesses multiple rows* of a standard 6T SRAM bitcell array (BCA) per pre-charge cycle via pulse width and amplitude modulated (PWAM) wordline (WL) enabling signals, and processes the resulting bitline (BL) voltage drops via *column pitch-matched low-swing analog circuits* in the periphery of the BCA. Thus, DIMA reduces the energy and latency cost of data access over conventional architectures while maintaining the storage density. DIMA has four sequentially executed processing stages as shown in Fig. 12:

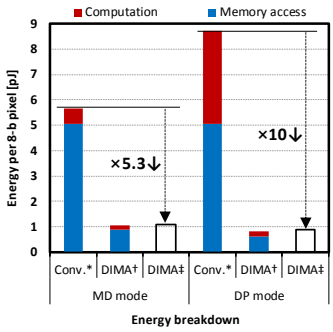
1) *multi-row functional read* (FR): fetches data stored in a column-major format by activating multiple rows per pre-charge cycle to generate an analog BL discharge voltage ΔV_{BL} that is a linearly weighted sum of the binary column-stored data, i.e., a coarse digital-to-analog conversion (D/A).
2) *BL processing* (BLP): computes word-level scalar arithmetic operations, e.g., addition, subtraction, or multiplication, by processing ΔV_{BL} in parallel on the BLs via column pitch-matched analog circuits.



(a)



(b)



(c)

Fig. 13. A mutli-functional DIMA IC (65 nm CMOS) prototype [27]: (a) architecture, (b) measured final classification accuracy versus energy trade-off, and (c) CORE energy comparison of DIMA (†post-layout simulations, ‡measured) versus conventional (Conv.) digital architecture (*SRAM energy measured and digital computation energy from post-layout simulations)

3) *cross BL processing* (CBLP): aggregates multiple BLP outputs via charge-sharing to obtain a scalar output.
4) *residual digital logic* (RDL): converts the analog output of CBLP into a digital word using an ADC, and implements any residual computations in the digital domain.

Table I, which compares DIMA and the conventional architecture [26], indicates that DIMA is superior to the conventional architectures in terms of bandwidth (data access rate), latency, and energy, but at the cost of circuit SNR. This clearly indicates that DIMA is a manifestation of a low SNR circuit fabric.

A multi-functional DIMA chip [27] realizing four machine learning algorithms — SVM, template matching, k-nearest neighbor, and matched filtering — was implemented in 65 nm CMOS comprising a 16KB standard 6T SRAM BCA (Fig. 13). The chip (Fig. 13(a)) consists of a DIMA CORE realiz-

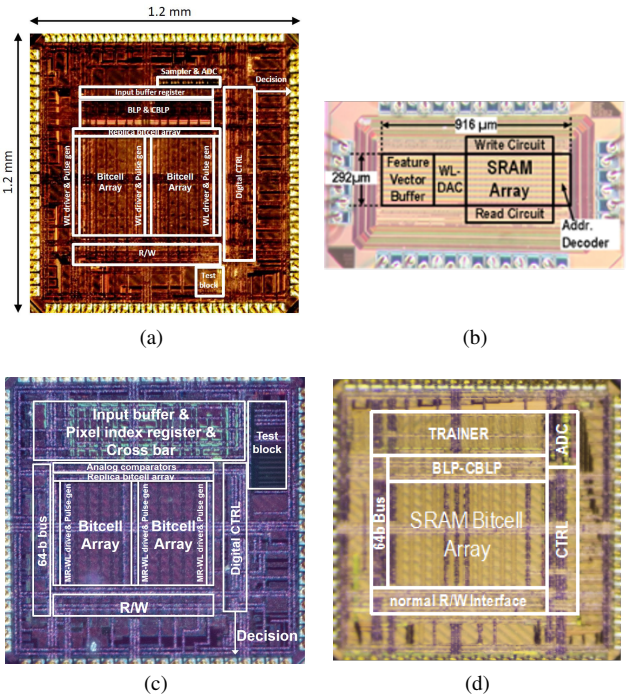


Fig. 14. Recent DIMA IC prototypes: (a) multi-functional (65 nm CMOS) [27], (b) boosted classifier (130 nm CMOS) [26], (c) random forest (65 nm CMOS) [28], and (d) with on-die learning (65 nm CMOS) [29].

ing programmable in-memory analog computations sequenced via a digital controller (CTRL). The DIMA CORE includes functional wordline drivers, BLP, and CBLP at the periphery of the BCA, each implementing one of the first three processing stages of DIMA, respectively. Fig. 13(b) shows that DIMA achieves up to 10× energy savings, 5.3× delay reduction, and hence 50× reduction in the energy-delay product (EDP) compared to a conventional function-specific digital implementation for face detection and recognition applications. In addition, the multi-functional DIMA chip (Fig. 14(a)) exhibits minimal, i.e., ≤1% accuracy degradation in all four algorithms.

Recently DIMA prototype ICs in Fig. 14 have shown energy savings ranging from 10× to 113×, throughput gains of up to 6×, and an energy-delay product reduction of 52× to 162× over custom von Neumann architectures [26]–[29]. Specifically, [26] (Fig. 14(b)) realizes a boosted classifier in 130 nm CMOS, using DDHR and EACB to embed computation in the memory array, resulting in a 113× energy reduction over a conventional von Neumann architecture for a 10-class MNIST dataset. It employs the Shannon-inspired design technique EACB (see Section IV-A) to overcome DIMA's low SNR property. A DIMA-based random forest algorithm IC was implemented in 65 nm CMOS [28] (Fig. 14(c)). Detection accuracies of 94% (8-class traffic sign recognition) and 97.5% (face detection) were achieved along with energy savings of 3.1×, a delay reduction of 2.2×, and an EDP reduction of 6.85× compared to a custom digital implementation. Similarly, DDHR was employed by the DIMA IC in [29] (Fig. 14(d)) to adapt to temperature and voltage fluctuations, changing input statistics, and usage patterns and thereby achieve energy savings of 2.4× over conventional DIMA with an accuracy close to that of a floating point algorithm in spite of DIMA's low SNR. A mixed-signal binarized neural network (MSBNN) IC in 28 nm CMOS [89] implements an 8-layer convolutional neural network (CNN) achieving a record-low energy efficiency of 3.8 μJ/decision at 237 frames/sec.

These realizations correspond to accelerators, implementing complete inference kernel, as have been widely considered in typical benchmarking (digital feature-vector or sensor-data input, classification decision output). Further, the baseline implementations correspond to optimized digital accelerators whose energies are estimated from silicon measurements of memory blocks and post-layout simulations of compute blocks.

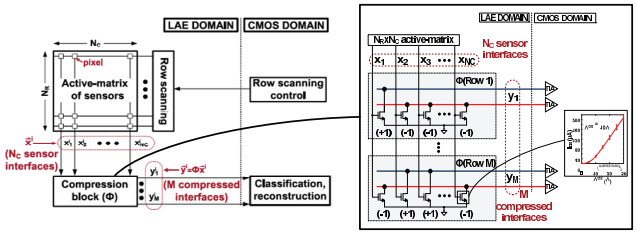
Thus, DIMA simultaneously enhances energy, latency, and accuracy by trading-off circuit SNR, and therefore is a low SNR fabric required by Shannon-inspired statistical computing framework. Indeed, it is possible to leverage Shannon's water-filling argument [14] to assign energy-optimal BL swings in a SRAM for a fixed accuracy [90], and further reduce the energy consumption of DIMA. To enhance the engineering design space based on these principles, the FR/BLP/CBLP/RDL steps enable DIMA-specialized signal processing algorithms to manage SNR by delaying hard-decision making stages (e.g., ADCs) [27], and recent work explores architectures and algorithms where computations involving very high dimensionalities can be broken and mapped across multiple DIMAs, to enable the bandwidth/latency/energy vs. SNR trade-off shown in Table I to be applied selectively [91].

Additionally, the similarity between DIMA and the dataflow of commonly encountered inference kernels makes DIMA well-matched to machine learning and cognitive applications. In addition to the energy efficiency and throughput gains, these prototypes have affected a fundamental transformation of the system architecture and the resulting trade-offs.

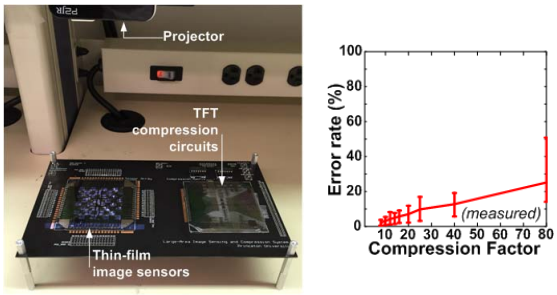
B. Deep In-Sensor Architectures (DISAs)

Consider a canonical sensor-inference system, which employs feature-extraction and classification stages to make decisions. DISAs aim to realize the functionality of these stages *within the sensing fabric*. This enables efficient architectures by reducing the communication requirements, but also by enabling algorithmic tools from statistical signal processing and machine learning to provide a bridge to the statistical computing principles above (Section IV).

We describe DISA in the context of an emerging technological fabric for sensing, namely large-area electronics (LAE). LAE is based on low-temperature processing of thin films, which makes it compatible with diverse materials, and large-area deposition methods and substrates, thus enabling diverse, expansive, and form-fitting arrays of transducers on glass, plastic, paper, and others. LAE is thus considered as a platform technology for next-generation, large-scale embedded sensing [92], [93]. However, low-temperature processing leads to orders-of-magnitude lower performance and energy efficiency of thin-film transistors (TFTs), making it necessary to employ CMOS technologies for processing sensory data and



(a)



(b)

Fig. 15. Demonstration of DISA for feature extraction from LAE image sensors [31]: (a) hardware architecture and (b) experimental results.

thereby leading to *hybrid systems* combining LAE and silicon CMOS. In such systems, it is the sensor-processor interface between the technology domains that limits the scalability and efficiency of systems [30]. Thus, DISAs can enhance efficiency by communicating features of classification decisions over interfaces, rather than raw data.

Figure 15(a) shows an architecture where TFTs are used to reduce data from an active-matrix array of LAE image sensors to features [31]. This architecture exploits the construct of compressive random-projection matrices from statistical signal processing, which have the property of preserving vector inner products [94]. Doing so enables the computation of similarity metrics such as vector distances. Importantly, random-projection matrices can be mapped to computational models well-aligned to the attributes of TFTs. A highly parallel reduction architecture simply feeds sensor data to the TFT gates in order to implement current summation on shared positive/negative nodes. Such a parallel architecture overcomes the performance limitations of TFTs, while the random nature of the projection itself enables tolerance to random device variations. Figure 15(b) shows an experimental demonstration of the system, illustrating the high-level of classification performance achieved for MNIST image recognition, even with aggressive matrix-compression ratios (error bars show min/max performance across ten 1-vs.-all binary classifiers for each digit), and further analysis, employing statistical device models calibrated to fabricated TFTs, shows high tolerance as TFT variation sources scale [31].

Second, Fig. 16(a) shows an architecture where TFTs are used to reduce data from an array of LAE image sensors to weak-classifier decisions [30]. This architecture exploits the EACB algorithm (see Section IV) to enable the realization of weak classifiers using TFTs. In particular, appropriate biasing of the two-TFT stacks shown leads to an output current approximating multiplication of the two gate-input voltages. Nonvolatile charge storage in the gate dielectric of the bottom TFT enables storage of a weight from training. Thus driving the top TFT with a sensor signal enables multiplication with the stored weight, and summation of the output currents from all pseudo-differential TFT stacks, thereby enables an approximate linear classifier. Measurements of the TFT-stack transfer function reveal significant non-linearity and variation (error-bars show standard deviation over 10 circuits). However, the experimental demonstration in Fig. 16(b) shows that EACB enables image-recognition performance at the level of an ideal classifier (SVM, implemented in MATLAB), reducing 36 sensor outputs to just 3-5 weak-classifier decisions for a simple dataset employed [30].

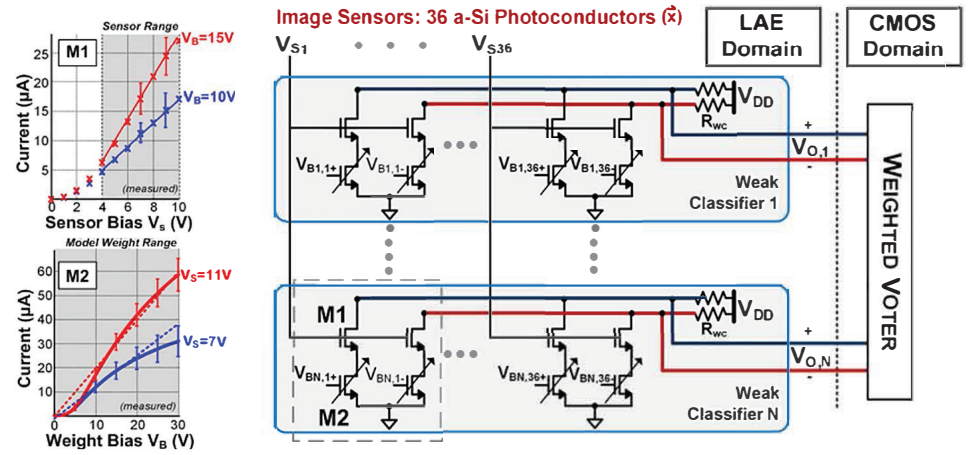
Beyond emerging sensing fabrics, the design principles above (Section IV) have been demonstrated within CMOS fabrics as well. For instance, in [95] multiplication operations were integrated within the analog-to-digital conversion process, enabling mapping of linear feature extraction and boosted linear classifiers. Direct conversion of sensor signals to weak-classifier decisions in this way demonstrated 13× and 29× energy reduction in a medical-sensor and image-recognition application, respectively. Going further, in [96] clocked-comparators with an array of transconductance-configurable inputs were used to directly convert analog sensor inputs to weak linear-classifier decisions. Utilizing EACB, this enabled a strong classifier, while replacing instrumentation amplifiers, ADCs, and digital processors. This leads to 34× energy reduction demonstrated for an MNIST image-recognition application, but more importantly demonstrates how the fundamental architectural tradeoffs can be altered. For instance, by directly deriving classification decisions, clocked comparators avoid the need for linear settling, as conventionally required within instrumentation amplifiers. This enables narrow noise-bandwidths and superior noise-power efficiency [96].

The more well-established architectures of such CMOS systems have enabled evaluation of energy savings in end-to-end sensor inference systems, where for instance the baseline architectures consist of analog front-end (instrumentation, ADC), feature extraction, and classification stages (whose energies are measured from prototypes or estimated using post-layout simulations).

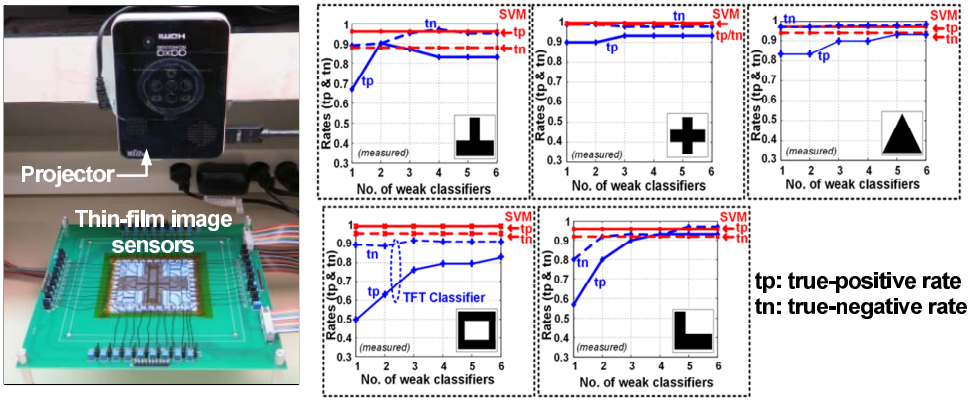
VI. Future Prospects

This paper has described the key elements of a Shannon-inspired statistical computing framework encompassing fundamental limits, design principles, and prototypes. These elements make it abundantly clear that this framework leads to unique system architectures and trade-offs, provides a road-map for operating such systems on the Pareto-optimal energy-delay-accuracy surface, and is particularly well-suited for today's data-centric workloads and nanoscale technologies. Some interesting avenues for further investigation are described next.

To date, the three statistical design principles described in Section IV (DDHR, SEC and HD computing) have been applied independently to various problems. There is much



(a)



(b)

Fig. 16. Demonstration of DISA for classification from LAE image sensors [30]: (a) hardware architecture and (b) experimental results.

value in finding effective ways to combine these, and other yet to be discovered techniques, to design large-scale computing systems in a manner similar to the design of advanced communication systems today. The latter includes the joint selection and optimization of the channel code (coding) in conjunction with the shaping of the channel (channel engineering) to enable low-complexity data recovery and decoding techniques to be employed at the receiver, all geared for the singular goal of minimizing the bit error rate with minimal complexity. Indeed, we believe that lessons from communication system design may prove to be useful in our quest.

The Shannon-inspired architectures described in Section V (DIMA and DISA) have come from our search for low SNR fabrics. By taking the radical step of embedding computation at the data source, these architectures have already shown orders-of-magnitude gains in the energy-delay product over von Neumann architectures. Much work remains in finding circuit and architectural techniques to make their energy-delay-accuracy trade-offs even more favorable. More importantly, it is expected that application of statistical design principles from Section IV to DIMA and DISA will enable computing at the limits of these trade-offs.

Much work remains to develop a comprehensive design methodology that weaves together various design principles and methods, with modeling and simulation techniques, to explore instruction set architectures for DIMA and DISA, compiler techniques to map applications on such architectures, and developing programming models that comprehend their intrinsic statistical nature. As Shannon-inspired statistical computing ties systems-to-devices, it presents many opportunities for systems researchers, architects, circuit designers, and device researchers to innovate in their specific domain using information-based metrics. It also also creates opportunities for systematic cross-layer optimization so that large-scale inference systems operating at the limits of energy-latency-accuracy, can be designed on deeply scaled semiconductor process technologies.

Acknowledgment

The authors gratefully acknowledge funding from NSF, DARPA, and SRC over the years. Much of the presented work was conducted within the Systems on Nanoscale Information fabriCs (2013–2017), and earlier in the Gigascale Systems Research Center (2003–2012), through collaborations with the researchers from both industry and academia brought together by these centers.

References

[1] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *Proc. London Math. Soc.*, vol. 2, no. 1, pp. 230–265, Nov. 1937.

[2] J. von Neumann, "First draft of a report on the EDVAC," Univ. of Pennsylvania, Tech. Rep., Jun. 1945.

[3] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, Apr. 1965.

[4] D. E. Nikonov and I. A. Young, "Overview of beyond-CMOS devices and a uniform methodology for their benchmarking," *Proc. IEEE*, vol. 101, no. 12, pp. 2498–2533, Jun. 2013.

[5] D. Nikonov and I. Young, "Benchmarking of beyond-CMOS exploratory devices for logic integrated circuits," *IEEE J. Explor. Solid-State Computat. Devices Circuits*, pp. 3–11, Apr. 2015.

[6] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[8] J. Dean *et al.*, "Large scale distributed deep networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2012, pp. 1223–1231.

[9] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," *SIGARCH Comput. Archit. News*, vol. 23, no. 1, pp. 20–24, Mar. 1995.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2012, pp. 1097–1105.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778.

[12] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[13] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[14] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[15] N. R. Shanbhag, "A mathematical basis for power-reduction in digital VLSI systems," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 11, pp. 935–951, Nov. 1997.

[16] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Trans. VLSI Syst.*, vol. 9, no. 6, pp. 813–823, Dec. 2001.

[17] N. R. Shanbhag, R. A. Abdallah, R. Kumar, and D. L. Jones, "Stochastic computation," in *Proc. Des. Autom. Conf (DAC)*, Jun. 2010, pp. 859–864.

[18] L. R. Varshney, "Toward limits of constructing reliable memories from unreliable components," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2015, pp. 114–118.

[19] A. Chatterjee and L. R. Varshney, "Energy-reliability limits in nanoscale circuits," in *Proc. Inf. Theory Appl. Workshop (ITA)*, Jan. 2016, pp. 1–6.

[20] A. Rahimi, S. Datta, D. Kleyko, E. P. Frady, B. Olshausen, P. Kanerva, and J. M. Rabaey, "High-dimensional computing as a nanoscalable paradigm," *IEEE Trans. Circuits Syst. I*, vol. 64, no. 9, pp. 2508–2521, Sep. 2017.

[21] R. Hegde and N. R. Shanbhag, "A voltage overscaled low-power digital filter IC," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, pp. 388–391, Feb. 2004.

[22] R. A. Abdallah and N. R. Shanbhag, "An energy-efficient ECG processor in 45-nm CMOS using statistical error compensation," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2882–2893, Nov. 2013.

[23] E. P. Kim, D. J. Baker, S. Narayanan, N. R. Shanbhag, and D. L. Jones, "A 3.6-mW 50-MHz PN code acquisition filter via statistical error compensation in 180-nm CMOS," *IEEE Trans. VLSI Syst.*, vol. 23, no. 3, pp. 598–602, Mar. 2015.

[24] Z. Wang, R. E. Schapire, and N. Verma, "Error adaptive classifier boosting (EACB): Leveraging data-driven training towards hardware resilience for signal inference," *IEEE Trans. Circuits Syst. I*, vol. 62, no. 4, pp. 1136–1145, Apr. 2015.

[25] Z. Wang, K. H. Lee, and N. Verma, "Overcoming computational errors in sensing platforms through embedded machine-learning kernels," *IEEE Trans. VLSI Syst.*, vol. 23, no. 8, pp. 1459–1470, Aug. 2015.

[26] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, Apr. 2017.

[27] M. Kang, S. K. Gonugondla, A. Patil, and N. R. Shanbhag, "A multi-functional in-memory inference processor using a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 53, no. 2, pp. 642–655, Feb. 2018.

[28] M. Kang, S. K. Gonugondla, and N. R. Shanbhag, "A 19.4 nJ/decision 364K decisions/s in-memory random forest classifier in 6T SRAM array," in *Proc. European Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2017, pp. 263–266.

[29] S. K. Gonugondla, M. Kang, and N. R. Shanbhag, "A 42 pJ/decision 3.12 TOPS/W robust in-memory machine learning classifier with on-chip training," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Pap.*, Feb. 2018, pp. 490–491.

[30] W. Rieutort-Louis, T. Moy, Z. Wang, S. Wagner, J. Sturm, and N. Verma, "A large-area image sensing and detection system based on embedded thin-film classifiers," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 281–290, Jan. 2016.

[31] T. Moy, W. Rieutort-Louis, S. Wagner, J. C. Sturm, and N. Verma, "A thin-film, large-area sensing and compression system for image detection," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 11, pp. 1833–1844, Nov. 2016.

[32] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata stud.*, vol. 34, pp. 43–98, 1956.

[33] P. Elias, "Computation in the presence of noise," *IBM J. Res. Develop.*, vol. 2, no. 4, pp. 346–353, Oct. 1958.

[34] N. Pippenger, "Reliable computation by formulas in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 194–197, Mar. 1988.

[35] B. Hajek and T. Weller, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 388–391, Mar. 1991.

[36] W. S. Evans and L. J. Schulman, "Signal propagation and noisy circuits," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2367–2373, Nov. 1999.

[37] S. Winograd and J. D. Cowan, *Reliable Computation in the Presence of Noise*. MIT Press Cambridge, Mass., 1963, no. 22.

[38] W. G. Brown, J. Tierney, and R. Wasserman, "Improvement of electronic-computer reliability through the use of redundancy," *IEEE Trans. Comput.*, vol. EC-10, no. 3, pp. 407–416, Sep. 1961.

[39] I. Koren and S. Y. H. Su, "Reliability analysis of N-modular redundancy systems with intermittent and permanent faults," *IEEE Trans. Comput.*, no. 7, pp. 514–520, Jul. 1979.

[40] N. Vaidya and D. K. Pradhan, "Fault-tolerant design strategies for high reliability and safety," *IEEE Trans. Comput.*, vol. 42, no. 10, pp. 1195–1206, Oct. 1993.

[41] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Architecture support for disciplined approximate programming," in *Proc. Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, Mar. 2012, pp. 301–312.

[42] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *Proc. IEEE European Test Symp. (ETS)*, May 2013, pp. 1–6.

[43] D. Ernst *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. IEEE/ACM Annu. Int. Symp. Microarchitecture (MICRO)*, Dec. 2003, pp. 7–18.

[44] K. A. Bowman, J. W. Tschanz, N. S. Kim, J. C. Lee, C. B. Wilkerson, S.-L. L. Lu, T. Karnik, and V. K. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 49–63, Jan. 2009.

[45] J. Tschanz, K. Bowman, C. Wilkerson, S.-L. Lu, and T. Karnik, "Resilient circuits—enabling energy-efficient performance and reliability," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2009, pp. 71–73.

[46] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[47] B. R. Gaines, "Stochastic computing systems," *Adv. Inf. Syst. Sci.*, vol. 2, no. 2, pp. 37–172, 1969.

[48] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2s, pp. 92:1–92:19, May 2013.

[49] J. George, B. Marr, B. E. S. Akgul, and K. V. Palem, "Probabilistic arithmetic and energy efficient embedded signal processing," in *Proc. Int. Conf. Compilers, Archit. Synthesis for Embedded Syst. (CASES)*, Oct. 2006, pp. 158–168.

[50] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "A self-tuning DVS processor using delay-error detection and correction," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 792–804, Apr. 2006.

[51] S. Das, C. Tokunaga, S. Pant, W. H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In situ error detection and correction for PVT and SER tolerance," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, Jan. 2009.

[52] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 1993, pp. 1064–1070.

[53] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[54] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[55] R. Fano, *Transmission of Information: A Statistical Theory of Communications*. Cambridge, MA: MIT. Press, 1961.

[56] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Develop.*, vol. 5, no. 3, pp. 183–191, Jul. 1961.

[57] W. H. Butler, T. Mewes, C. K. A. Mewes, P. B. Visscher, W. H. Rippard, S. E. Russek, and R. Heindl, "Switching distributions for perpendicular spin-torque devices within the macrospin approximation," *IEEE Trans. Magn.*, vol. 48, no. 12, pp. 4684–4700, Dec. 2012.

[58] D. Seo and L. R. Varshney, "Information-theoretic limits of algorithmic noise tolerance," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Oct. 2016, pp. 1–4.

[59] J. H. Engel, S. Eryilmaz, S. Kim, M. BrightSky, C. Lam, H.-L. Lung, B. A. Olshausen, and H.-S. Wong, "Capacity optimization of emerging memory systems: A shannon-inspired approach to device characterization," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Dec. 2014, pp. 29.4.1–29.4.4.

[60] L. R. Varshney, P. J. Sjöström, and D. B. Chklovskii, "Optimal information storage in noisy synapses under resource constraints," *Neuron*, vol. 52, no. 3, pp. 409–423, Nov. 2006.

[61] M. G. Taylor, "Reliable information storage in memories designed from unreliable components," *Bell Syst. Tech. J.*, vol. 47, no. 10, pp. 2299–2337, Dec. 1968.

[62] L. R. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4427–4444, Jul. 2011.

[63] L. Chang, A. Chatterjee, and L. R. Varshney, "Performance of LDPC decoders with missing connections," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 511–524, Feb. 2017.

[64] H. Chen, L. R. Varshney, and P. K. Varshney, "Noise-enhanced information systems," *Proc. IEEE*, vol. 102, no. 10, pp. 1607–1621, Oct. 2014.

[65] W. Evans and N. Pippenger, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1299–1305, May 1998.

[66] W. S. Evans and L. J. Schulman, "On the maximum tolerable noise of k-input gates for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 3094–3098, Nov. 2003.

[67] A. Mozeika and D. Saad, "On reliable computation by noisy random boolean formulas," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 637–644, Jan. 2015.

[68] A. Chatterjee and L. R. Varshney, "Energy-reliability limits in nanoscale neural networks," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.

[69] ——, "Optimal energy allocation in reliable neural sensory processing," in *Proc. Conf. Cognitive Computational Neurosci. (CCN)*, Sep. 2017.

[70] A. D. Patil, S. Manipatruni, D. Nikonov, I. A. Young, and N. R. Shanbhag, "Shannon-inspired statistical computing to enable spintronics," *arXiv preprint arXiv:1702.06119*, Feb. 2017.

[71] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Comput.*, vol. 1, no. 2, pp. 139–159, Jan. 2009.

[72] H. Jia and N. Verma, "Exploiting approximate feature extraction via genetic programming for hardware acceleration in a heterogeneous microprocessor," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 1016–1027, Apr. 2018.

[73] R. E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms*. MIT Press, 2012.

[74] Z. Wang and N. Verma, "Enabling hardware relaxations through statistical learning," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, Sep. 2014, pp. 319–326.

[75] Z. Wang, "Relaxing the implementation of embedded sensing systems through machine learning and statistical optimization," Ph.D. dissertation, Princeton University, 2017.

[76] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 323–336, Mar. 2012.

[77] R. A. Abdallah and N. R. Shanbhag, "Robust and energy efficient multimedia systems via likelihood processing," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 257–267, Feb. 2013.

[78] G. V. Varatkar, S. Narayanan, N. R. Shanbhag, and D. L. Jones, "Variation-tolerant, low-power PN-code acquisition using stochastic sensor NOC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2008, pp. 380–383.

[79] S. Zhang and N. R. Shanbhag, "Embedded algorithmic noise-tolerance for signal processing and machine learning systems via data path decomposition," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3338–3350, Jul. 2016.

[80] S. K. Gonugondla, B. Shim, and N. R. Shanbhag, "Perfect error compensation via algorithmic error cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar. 2016, pp. 966–970.

[81] B. Shim, "Error-tolerant digital signal processing," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2005.

[82] Y. Lin, S. Zhang, and N. R. Shanbhag, "Variation-tolerant architectures for convolutional neural networks in the near threshold voltage regime," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2016, pp. 17–22.

[83] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[84] H. Li *et al.*, "Hyperdimensional computing with 3D VRRAM in-memory kernels: Device-architecture co-design for energy-efficient, error-resilient language recognition," in *Proc. IEEE Int. Electron Devices Meeting (IEDM)*, Dec. 2016, pp. 16.1.1–16.1.4.

[85] T. Wu, P.-C. Huang, A. Rahimi, H. Li, M. Shulaker, J. Rabaey, H.-S. P. Wong, and S. Mitra, "Brain-inspired computing exploiting carbon nanotube FETs and resistive RAM: Hyperdimensional computing case study," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Pap.*, Feb. 2018.

[86] S. Manipatruni, D. E. Nikonov, and I. A. Young, "Modeling and design of spintronic integrated circuits," *IEEE Trans. Circuits Syst. I*, vol. 59, no. 12, pp. 2801–2814, Dec. 2012.

[87] M. Kang, M.-S. Keel, N. R. Shanbhag, S. Eilert, and K. Curewitz, "An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2014, pp. 8326–8330.

[88] N. Shanbhag, M. Kang, and M.-S. Keel, "Compute memory," U.S. Patent 9 697 877, Jul. 4, 2017.

[89] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, "An always-on 3.8μJ/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28nm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Pap.*, Feb. 2018, pp. 222–223.

[90] Y. Kim, M. Kang, L. R. Varshney, and N. R. Shanbhag, "Generalized water-filling for source-aware energy-efficient SRAMs," *IEEE Trans. Commun.*, 2018, accepted.

[91] Y. Tang, J. Zhang, and N. Verma, "Scaling Up In-Memory-Computing Classifiers via Boosted Feature Subsets in Banked Architectures," *IEEE Trans. Circuits Syst. II*, 2018, accepted.

[92] S. Wagner, S. P. Lacour, J. Jones, P. I. Hsu, J. C. Sturm, T. Li, and Z. Suo, "Electronic skin: architecture and components," *Physica E: Low-dimensional Syst. Nanostructures*, vol. 25, no. 2, pp. 326–334, Jul. 2004.

[93] T. Someya, B. Pal, J. Huang, and H. E. Katz, "Organic semiconductor devices with enhanced field and environmental responses for novel applications," *MRS Bulletin*, vol. 33, pp. 690–696, Jul. 2008.

[94] S. Dasgupta and A. Gupta, "An elementary proof of the Johnson and Lindenstrauss lemma," *Random Structures and Algorithms*, vol. 22, no. 1, pp. 60–65, Nov. 2002.

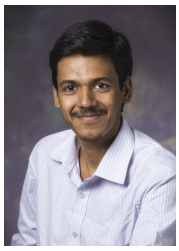
[95] Z. Wang, J. Zhang, and N. Verma, "Realizing low-energy classification systems by implementing matrix multiplication directly within an ADC," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 6, pp. 825–837, Dec. 2015.

[96] Z. Wang and N. Verma, "A low-energy machine-learning classifier based on clocked comparators for direct inference on analog sensors," *IEEE Trans. Circuits Syst. I*, vol. 64, no. 11, pp. 2954–2965, Nov. 2017.

Naresh R. Shanbhag (F'06) received the Ph.D. degree in Electrical Engineering from the University of Minnesota, Minneapolis, MN, USA, in 1993. From 1993 to 1995, he was with the AT&T Bell Laboratories, Murray Hill, NJ, USA, where he led the design of high-speed transceiver chip-sets for very high-speed digital subscriber line. In 1995, he joined the University of Illinois at Urbana-Champaign, Urbana, IL, USA. He has held visiting faculty appointments at the National Taiwan University, Taipei, Taiwan, in 2007, and at Stanford University, Stanford, CA, USA, in 2014. He is currently the Jack Kilby Professor of Electrical and Computer Engineering with the University of Illinois at UrbanaChampaign. His current research interests include the design of energy-efficient integrated circuits and systems for communications, signal processing, and machine learning. He has authored or co-authored more than 200 publications in this area and holds 13 U.S. patents. Dr. Shanbhag was a recipient of the National Science Foundation CAREER Award in 1996, the IEEE Circuits and Systems Society Distinguished Lecturership in 1997, the 2010 Richard Newton GSRC Industrial Impact Award, and multiple Best Paper Awards. In 2000, he co-founded and served as the Chief Technology Officer of Intersymbol Communications, Inc., (acquired in 2007 by Finisar Corporation) a semiconductor startup that provided DSP-enhanced mixed-signal ICs for electronic dispersion compensation of OC-192 optical links. From 2013 to 2017, he was the founding Director of the Systems On Nanoscale Information fabriCs Center (SONIC), a 5-year multi-university center funded by DARPA and SRC under the STARnet program.

Naveen Verma received the B.A.Sc. degree in Electrical and Computer Engineering from the UBC, Vancouver, Canada in 2003, and the M.S. and Ph.D. degrees in Electrical Engineering from MIT in 2005 and 2009 respectively. Since July 2009 he has been with the Department of Electrical Engineering at Princeton University, where he is currently an Associate Professor. His research focuses on advanced sensing systems, exploring how systems for learning, inference, and action planning can be enhanced by algorithms that exploit new sensing and computing technologies. This includes research on large-area, flexible sensors, energy-efficient statistical-computing architectures and circuits, and machine-learning and statistical-signal-processing algorithms. Prof. Verma has served as a Distinguished Lecturer of the IEEE Solid-State Circuits Society, and currently serves on the technical program committees for ISSCC, VLSI Symp., DATE, and IEEE Signal-Processing Society (DISPS). Prof. Verma is recipient or co-recipient of the 2006 DAC/ISSCC Student Design Contest Award, 2008 ISSCC Jack Kilby Paper Award, 2012 Alfred Rheinstein Junior Faculty Award, 2013 NSF CAREER Award, 2013 Intel Early Career Award, 2013 Walter C. Johnson Prize for Teaching Excellence, 2013 VLSI Symp. Best Student Paper Award, 2014 AFOSR Young Investigator Award, 2015 Princeton Engineering Council Excellence in Teaching Award, and 2015 IEEE Trans. CPMT Best Paper Award.

Yongjune Kim (S'12–M'16) received the B.S. and M.S. degrees in Electrical and Computer Engineering from Seoul National University, Seoul, South Korea, in 2002 and 2004, respectively, and the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2016. From 2007 to 2011, he was with Samsung Electronics and Samsung Advanced Institute of Technology, South Korea. He is currently a Post-Doctoral Scholar with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA. His research interests include coding and information theory for nanoscale devices, energy-efficient computing, and machine learning. He received the Best Paper Award at the IEEE International Conference on Communications (ICC), the Student Paper Award at the IEEE International Symposium on Circuits and Systems (ISCAS), and the Best Paper Award at the Samsung Semiconductor Technology Symposium.



Ameya D. Patil (S'15) received the B.Tech. degree in 2014 from the department of Electrical Engineering at Indian Institute of Technology (IIT) Hyderabad, India. He received his M.S. degree in 2016 from the department of Electrical and Computer Engineering (ECE) at the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, where he is currently pursuing his Ph.D. degree. His research interests lie at the intersection of machine learning, circuits, and architecture. He is a recipient of the Joan and Lalit Bahl Fellowship from the ECE department at UIUC in 2015-16 and 2016-17.

Lav R. Varshney (S'00–M'10–SM'15) received the B.S. degree (*magna cum laude*) in electrical and computer engineering with honors from Cornell University, Ithaca, New York, in 2004. He received the S.M., E.E., and Ph.D. degrees, all in electrical engineering and computer science, from the Massachusetts Institute of Technology, Cambridge, in 2006, 2008, and 2010, where his theses received the E. A. Guillemin Thesis Award and the J.-A. Kong Award Honorable Mention. He is an assistant professor in the Department of Electrical and Computer Engineering, the Department of Computer Science (by courtesy), the Coordinated Science Laboratory, the Beckman Institute, and the Neuroscience Program at the University of Illinois at Urbana-Champaign. He is also leading curriculum initiatives for the new B.S. degree in Innovation, Leadership, and Engineering Entrepreneurship in the College of Engineering. During 2010–2013, he was a research staff member at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York. His research interests include information and coding theory; limits of nanoscale, human, and neural computing; human decision making and collective intelligence; and creativity. Dr. Varshney is a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi. He received the IBM Faculty Award in 2014 and was a Finalist for the Bell Labs Prize in 2014 and 2016. He and his students have won several best paper awards. His work appears in the anthology, *The Best Writing on Mathematics 2014* (Princeton University Press). He currently serves on the advisory board of the AI XPRIZE.