Moving Toward Intelligence: Detecting Symbols on 5G Systems Through Deep Echo State Network

Kangjun Bai, *Student Member, IEEE*, Yang Yi, *Senior Member, IEEE*, Zhou Zhou, *Student Member, IEEE*, Shashank Jere, *Student Member, IEEE*, and Lingjia Liu, *Senior Member, IEEE*

***Abstract*— Due to the nonlinear distortion caused by radio-frequency (RF) components in the transceiver, detecting transmitted symbols for multiple-input and multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM) systems can be challenging and resource consuming. In this work, we introduce a Deep Echo State Network (DESN) to serve as the symbol detector for 5G communication networks. Our DESN employs memristive synapses as the dynamic reservoir layer to accelerate the learning algorithm and computation. By cascading multiple dynamic reservoir layers in a hierarchical processing structure, our DESN processes received signal in both spatial and temporal domains. The resulting hybrid memristor-CMOS co-design provides the nonlinear computation required by the reservoir layer while significantly reduces the power consumption. From the benchmark on nonlinear system prediction, our DESN exhibits 10.31X reduction on the prediction error compared to state-of-the-art neural network designs. Moreover, our DESN records a bit error rate (BER) of 5.76×10^{-2} on the high-speed transmitted symbol detection task for MIMO-OFDM systems, yielding 47.73% more precise than state-of-the-art techniques in the literate for 5G communication networks.**

***Index Terms*— Deep learning, reservoir computing, echo state network, memristive crossbar, 5G/beyond-5G system, MIMO-OFDM, symbol detection.**

I. Introduction

THE fifth generation (5G) communication networks will not only interconnect people, but also interconnect machines and devices. The new level of performance and efficiency of 5G communication networks will empower new user experiences, delivering multi-Gbps (Gigabits per second) peak rate, ultra-low latency, and massive capacity [1].

The enhanced mobile broadband (eMBB), the ultra-reliable low latency communications (URLLC), and the massive machine type communications (mMTC) are the three primary sets of cases defined for 5G New Radio (NR). In particular, eMBB supports stable wireless connections with high data rates across a wide coverage area; URLLC ensures ultra-low latency connections for mission critical communications, such as remote surgery and autonomous vehicles; mMTC reinforces durable connections for an enormous number of simultaneous devices within a small area. Consequently, the corresponding signal processing in 5G communication networks is challenging and resource consuming due to the nonlinear distortion caused by practical radio frequency (RF) components in the transceiver chain, as well as the noise interference introduced by wireless connections.

In scenarios where seamless wide-area coverage is needed, 5G communication networks support the high-speed mobility up to 500km/hr [2]. As such, a high-speed and reliable receiver is needed to conduct the symbol detection under different wireless propagation characteristics. In recent years, bio-inspired artificial neural networks (ANNs) have provided effective solutions for various nonlinear dynamic systems [3]. Based on the framework of supervised learning, ANNs can learn to reconstruct corrupted symbols from the aforementioned distortion, interference, and noise at the receiver. In particular, recurrent neural networks (RNNs), a subset of ANNs, allow effective processing and learning of nonlinear signals with recurrent memory [4]. Due to the nonlinear sequential feature of communication signals, RNNs could be one possible deep learning architecture candidate for the task of symbol detection in wireless systems.

In this work, we introduce a Deep Echo State Network (DESN) to serve as the symbol detector for 5G multiple-input and multiple-out orthogonal frequency-division multiplexing (MIMO-OFDM) systems. Major contributions of our work are summarized as follows:

- By fashioning multiple reservoir layers in a hierarchical processing structure, our DESN enables the learning behavior on both temporal and spatial domains;
- In the endeavor to accelerate the learning operation, the hybrid memristor-CMOS co-design facilitates the in-memory computing capability, significantly improving the computing and power efficiency;
- Benchmark result on the nonlinear system prediction exhibits 10.31X reduction on the prediction error compared to state-of-the-art neural network designs;
- Experimental results on the 5G MIMO-OFDM symbol detection record a bit error rate of 5.76×10^{-2}, yielding 47.73% more precise compared to state-of-the-art techniques in the literate for 5G communication networks.

Manuscript received December 29, 2019; revised March 20, 2020; accepted April 19, 2020. Date of publication May 4, 2020; date of current version June 12, 2020. This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF-1750450, Grant ECCS-1811497, and Grant CCF-1937487. This article was recommended by Guest Editor C. Zhang. *(Corresponding author: Yang Yi.)*

The authors are with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: kangjun@vt.edu; yangyi8@vt.edu; zhou89@vt.edu; shashankjere@vt.edu; llj@vt.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2020.2992238

In this paper, the general architecture of memristive-based deep neural networks and the symbol detection technique in the literate for 5G MIMO-OFDM systems are introduced in Section II and Section III, respectively. The design methodology and experimental evaluations on our DESN are demonstrated in Section IV, followed by benchmark and application evaluations in Section V. The paper is then concluded in Section VI.

II. Memristive-Based Deep Neural Network

A. Deep Neural Networks

Deep neural networks (DNNs), also known as the deep learning, provide systems the ability to automatically learn and improve from data through a general-propose learning algorithm, rather than designed by human engineers [5]. The major structure of DNNs can be summarized into two categories: (1) the depth-in-space structure represented by feedforward neural networks (FNNs), and (2) the depth-in-time structure represented by RNNs.

The depth-in-space structure creates a network through multiple hidden layers to learn the representation of data with different levels of abstractions [6], enabling the spatial-based learning capability. For instance, in the natural image recognition task [7], learned features in the first hidden layer typically represent edges at particular orientations and locations of an image, while rest of hidden layers may detect motifs by recognizing particular arrangements of edges. On the other hand, the major design challenge in the depth-in-space structure is that the learning time, the inference accuracy, and the computational power are toughly affected by the number of hidden layers and their associated neurons.

At the meantime, the depth-in-time structure creates a network with recurrent memory, establishing the context of data and allowing the system to learn from its previous knowledge, and thus, enabling the temporal-based learning capability. For instance, Hochreiter and Schmidhuber introduce a long short-term memory (LSTM) [8] with memory cells and forget gate, such that values would be remembered over arbitrary time intervals within memory cells and information would be regulated in the forget gate. However, due to the state of the network with recurrent memory, the present neural state can possibly depend on all previous learned information. In other words, all internal weight matrices and bias vector parameters are needed to be learned, resulting in a computational-expensive learning process.

The reservoir computing, representing a unified computing framework divided from conventional RNNs as demonstrated in Fig. 1, utilizes a dynamic reservoir layer having the short-term memory for high-dimensional feature projection. The node state of the generic reservoir computing model at the present time step, $s(t)$, can be expressed as

$$s(t) = f(x(t) \cdot W_{in} + s(t-1) \cdot W_{res} + y(t-1) \cdot W_{fb}), \quad (1)$$

where $f()$ is a nonlinear activation function; $x(t)$ represents the input at the present time step; $s(t-1)$ and $y(t-1)$ are the internal state and output state of the network, respectively, from the previous time step; W_{in}, W_{res}, and W_{fb} denote input weights, internal weights within the reservoir layer, and feedback weights from the output to the reservoir, respectively. The output state of the network at the present time step can be then expressed in terms of the internal state of the network and output weighted elements, W_{out}, which can be written as

$$y(t) = s(t) \cdot W_{out}. \quad (2)$$

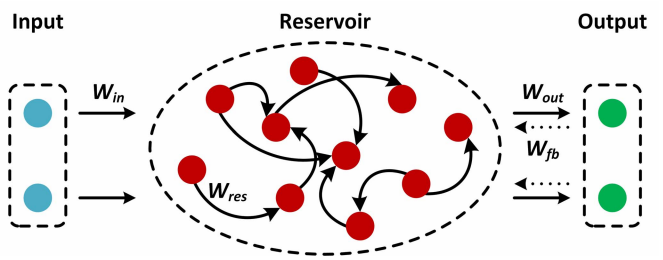


Fig. 1. Generic architecture of reservoir computing model.

In general, the echo state network (ESN) [9] and the liquid state machine (LSM) [10] are the two representations of the reservoir computing model. The methodology of internal signal processing is the major characteristic that sets these two models apart. For instance, in the ESN, actual numerical number from the input are adapted for the computation, while spiking signals are examined in the LSM during the operation. The major characteristic of the reservoir computing is that input weights and weights within the internal reservoir layer are fixed at all time, and thus, training operations to W_{in}, W_{res}, and W_{fb} are not required. More specifically, the role of the reservoir layer is to project the sequential input onto a higher dimensional space, such that crucial features of input information can be efficiently readout by a simple learning algorithm with output weighted elements.

B. Memristive Synapse With In-Memory Computing Capability

Together with the development of DNNs, the crossbar with emerging nonvolatile memory (NVM) has been considered as a promising candidate for intensive vector-matrix computation as in the neural network design, e.g., the resistive random-access memory (ReRAM) [11]. The ReRAM, a type of memristor, is a two-terminal metal-oxide-based nano-scale device, which performs the same functionality as a variable resistor with the non-volatility characteristic. Due to the formation of conductive filaments in the insulating material between two terminals, the resistance of a ReRAM cell can be switched from its high resistance state (HRS) to its low resistance state (LRS) when the stimulus across the device excesses a specific threshold, or vice versa.

The general mathematical representation of the neural computation at the j-th output neuron can be written as

$$y_j = \sum_{i=1}^{m} x_i \cdot W_{ij}, \quad (3)$$

where x_i and m represent the input vector and its data length, respectively, and W_{ij} denotes the particular weighted element

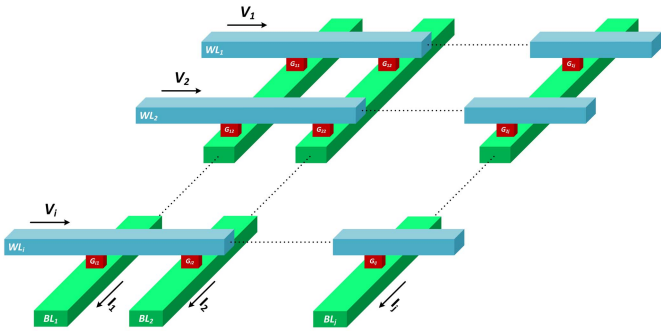


Fig. 2. Representation of vector-matrix computation on a memristive crossbar.

located between the i-th input neuron and the j-th output neuron. By mapping the input vector to analogue voltage and the weight matrix to memristive crossbar as depicted in Fig. 2, the vector-matrix computation as in the neural network design can be realized by sampling the total output current on each bit-line, I_j, which can be expressed as

$$I_j = \sum_{i=1}^{m} V_i \cdot G_{ij}, \tag{4}$$

where V_i is the i-th input voltage, and G_{ij} denotes the conductance ($G = \frac{1}{R}$) of a ReRAM cell located between the i-th word-line and the j-th bit-line. The memristor is naturally adapt in the historical behavior [12]. Firstly, the memristor-based crossbar supports a numerous amount of signal propagation within a small silicon area, mimicking synaptic connections in a neurological system. Secondly, the memristor-based crossbar inherently provides the vector-matrix computation with the intrinsic parallel-computing capability, imitating the operation of dendrite potential [13].

Due to the high access latency to memory unit, the conventional computing architecture can no longer offer timely response [14]. Through the crossbar structure, information within each column of memory cells can be readout directly via the sensing element at the end of each bit-line, as expressed in (4). Such computing structure eliminates the use of external memory storage and power-hungry peripheral devices, significantly improving the computing and energy efficiency.

III. 5G Communication Systems

In 5G communication networks, one of the major challenges is to conduct the detection on transmitted symbols for MIMO-OFDM systems under heterogeneous environments and channel conditions. This is because the received signal in a MIMO-OFDM system is the superposition of all modulation symbols associated with its sub-carriers, in which modulation symbols refer to the character selected from a predefined finite alphabet table [15]. Increasing the size of the alphabet table can convey more information through one modulation symbol.

Fig. 3 demonstrates general transmitting and receiving operations in MIMO-ODFM systems. In the conventional receiving operation, channel estimation is firstly conducted, followed by the symbol detection based on the estimated channel. However, such approach requires accurate channel estimation, which is usually challenging and resource consuming. In fact, there is a clear trade-off between the performance of channel estimation and resources used for data transmission, showing that more accurate channel estimation will require more resources to be allocated for this process, which results in less resources available for data transmission. On the other hand, it is crucial for a 5G communication network to be able to conduct MIMO symbol detection methods under heterogeneous channel knowledge (e.g., limited or even without available channel knowledge). In this section, we first introduce conventional MIMO-OFDM symbol detection techniques in literature with their associated hardware challenges.

A. Conventional MIMO-OFDM Symbol Detection

The MIMO-OFDM technology is the foundation of modern cellular networks and wireless local area networks. The Maximum Likelihood (ML) detector is one of the optimal solutions to the MIMO detection problem [16]. However, a brute-force ML detector implementation involves an exhaustive search over the space of all possible transmitted symbols, and thus, such detection method does not scale properly with the modulation order and the number of antennas, making the hardware complexity prohibitively high. The soft-output Sphere Decoding (SD) [17] is another optimal decoding technique that can achieve ML or near-ML Bit Error Rate (BER). However, such computational complexity also scales exponentially with the number of transmitted data streams [18].

The massive MIMO, a fundamental technology that forms a basis for modern wireless standards, proposes employing antenna arrays with significantly larger number of antenna elements than conventional MIMO systems. In a massive MIMO uplink, complexity and power consumption are key considerations in data detection tasks at a Base Station (BS) with hundreds of antenna elements and a large number of users. Consequently, low-complexity albeit sub-optimal detection schemes such as Zero Forcing (ZF), Minimum Mean Squared Error (MMSE), and Successive Interference Cancellation (SIC)-based schemes [19], [20] have been introduced. Although such linear detection schemes result in reduced BER performance compared to optimal detectors, such hardware implementation complexity is much lower.

Low-order MIMO symbol detectors based on linear detection schemes have been introduced, e.g., a FPGA-based MMSE detector for a 4 x 4 MIMO system [21]. Moreover, a sphere decoder based on Djistra's algorithm for a 4 x 4 MIMO system has implemented using the 180nm CMOS process [22]. To this end, several Lattice Reduction (LR) algorithms that only increase the pre-processing complexity have been used in conjunction with linear or SIC detectors with available FPGA implementations [23]. An MMSE-based detector for a 3GPP LTE-based 128 antenna, 8 user massive MIMO system had its first FPGA implementation demonstrated in [24]. However, such model-based approaches to MIMO symbol detection rely on simplification techniques, e.g., a matrix inversion to achieve correspondingly lower symbol-rate hardware complexity and power consumption.

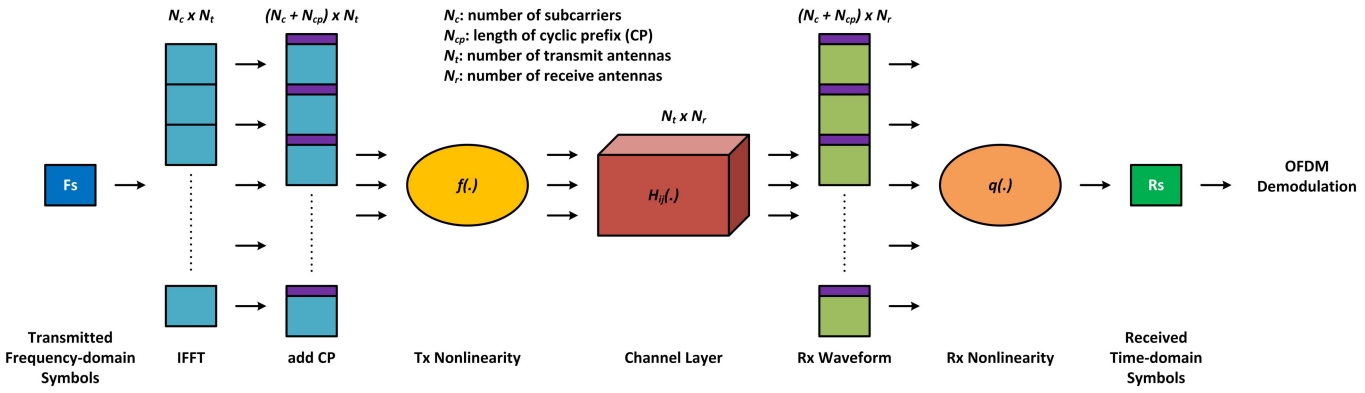


Fig. 3. General transmitting and receiving operations of MIMO-OFDM systems.

Alternatively, data-driven approaches powered by DNNs may pave the way towards large-scale MIMO detection with acceptable hardware implementation costs.

B. Neural Network-Based Symbol Detection

The conventional symbol detection method in MIMO-OFDM systems relies on modeling the convolutional feature of the transmission channel and solving the formulated problem based on the model. Recent advances in DNNs offer a solution to the symbol detection problem without relying on such model-based assumptions. First, DNNs can be used to perform parameter tuning based on existing symbol detection methods. For instance, in [25], a DNN is used to conduct MIMO symbol detection based on the estimated Channel State Information (CSI) and the received symbols as inputs. Moreover, in [26], a DNN is constructed based on iterative soft-thresholding algorithms to fine tune the parameters for MIMO symbol detection. However, such methods usually require a large training dataset along with explicit CSI availability. In addition, they do not consider the effect of OFDM on MIMO symbol detection. Second, the MIMO symbol detection can be formulated as a classification problem, whereby DNNs can be directly utilized. For instance, in [27], an ESN is applied for MIMO-OFDM symbol detection without relying on explicit CSI. The effectiveness of such method is evaluated via comparison with conventional model-based methods with the consideration of RF impairments. Furthermore, [28] presents an energy efficient perspective, showing the energy efficiency of the ESN-based detection technique to be better than the popular linear MMSE (LMMSE)-based approach. By taking the ESN-based detection approach further, [29] introduced a windowed ESN (WESN) by adding a sliding window to the ESN's input to enhance the short-term memory of the underlying ESN, showing that the performance of WESN to be better than the standard ESN when using LTE/LTE-Advanced compatible reference/pilot signals as the training set.

For such single-layer ESNs, the memory capacity, which represents the amount of input data an ESN can store, is limited. As the complexity of input dataset scales up, the learning capability of ESNs from short-term memory reduces, and thus, increasing the prediction error. In [30], a series deep ESN architecture is introduced by cascading multiple ESN in series, allowing such system to capture more features between input and output sequences, and thus, improving the overall prediction accuracy. Moreover, in [31], a deep ESN with stacked hierarchy of ESN is introduced to achieve multiple temporal representation of input sequence and enhance the richness of reservoir states as well as the memory capacity. In recent years, the concept of ESN is also implemented with the federated learning for mobile edge applications, e.g., the cyber-security [32] and the virtual reality [33].

Due to the nonlinear sequential feature of communication signals, RNNs could be one possible deep learning architecture candidate for the task of symbol detection in wireless systems. To this end, we investigate a computational- and energy-efficient RNN-based symbol detector, in particular, a symbol detector based on the topology of deep ESN.

IV. Design Methodology

A. Deep Echo State Network

It can be observed that transmitted signals undergo attenuation and delay due to the nonlinear distortion of wireless transmission. As discussed in Section III, an accurate channel estimation, required by the symbol detection on MIMO systems, relies in obtaining accurate CSI estimation and channel equalization. Unlike conventional detection techniques, our DESN-based symbol detector can learn to reconstruct corrupted symbols from the aforementioned distortion, interference, and noise at the receiver based on the framework of supervised learning algorithm.

The general architecture of our DESN is demonstrated in Fig. 4. In general, our DESN contains three major computing layers, namely, the input layer, the cascaded dynamic reservoir layer with intermediate I/Os, and the output layer. During the computation, a set of complex time-domain symbols of binary digits with both real and imaginary elements are applied as the global input signal, which can be defined as $u(t) = x^{(1)}(t) \in N_U$, where N_U is the global input dimension. Such input layer is associated with the anterior reservoir layer through global input weights, $W_{in}^{(1)} \in [N_R \times N_U]$, where N_R is the number of neurons in each reservoir layer. Within

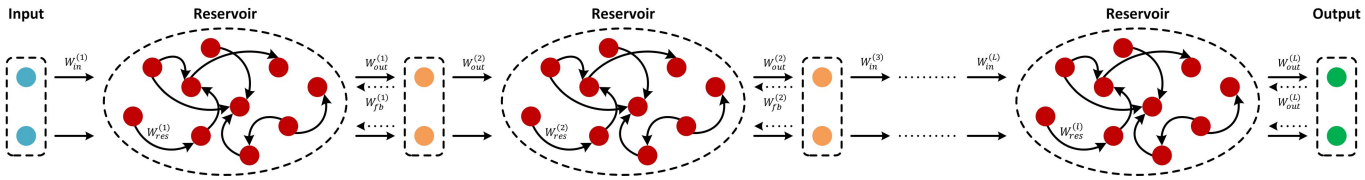


Fig. 4. Architecture of Deep Echo State Network (DESN).

the dynamic hidden layer, each hidden reservoir layer adopts the intermediate output, generated from the previous module, as its input to compute the corresponding output signal. Based on (1), by denoting the total number of hidden dynamic reservoir layer as N_L, the state of the network at the l-th hidden reservoir layer at the present time step can be written as

$$s^{(l)}(t) = f(x^{(l)}(t) \cdot W_{in}^{(l)} + s^{(l)}(t-1) \cdot W_{res}^{(l)} + y^{(l)}(t-1) \cdot W_{fb}^{(l)}), \quad (5)$$

where $x^{(l)}(t) = y^{(l-1)}(t)$ represents the l-th local input at the current time step; $W_{in} \in [N_R \times N_U]$, $W_{res} \in [N_R \times N_R]$, and $W_{fb} \in [N_R \times N_Y]$ denote input weights, internal weights, and feedback weights, respectively; $y^{(l)}(t-1)$ is the l-th local output at the previous time step. From (2), the l-th output state of the network at the present time step can be rewritten as

$$y^{(l)}(t) = s^{(l)}(t) \cdot W_{out}^{(l)}, \quad (6)$$

where $W_{out} \in [N_Y \times N_R]$ is output weights, and N_Y represents the output dimension. Unlike conventional RNNs, W_{in}, W_{res}, and W_{fb} in each hidden reservoir layer remain fixed at all times, while W_{res} is sparsely connected. Such structure significantly reduces the learning cost, and decompose various levels of interference for the received OFDM signal.

By transforming the processing structure into a hierarchy of stacked reservoir layers, the learning behavior is carried out layer by layer. For instance, the intermediate output of the first hidden reservoir layer is learned based on the input OFDM signal, while each latter hidden reservoir layer is learned based on the computed results from its previous layer. To reduce the design complexity, the teacher forcing for each hidden reservoir layer is the same. Correspondingly, the final output, $y^{(L)}(t)$, generated from the last hidden reservoir layer, estimates the desired OFDM symbol through global output weights, $W_{res}^{(L)}$; such operation can be achieved by minimizing the $L2$ norm distance between the computed output, $\hat{y}(t)$, and the targeted output, $y(t)$, which can be expressed as

$$\min_{W_{out}} \sum_{0}^{N_U-1} \left\| y(t) - \hat{y}(t) \right\|_2^2. \quad (7)$$

As such, readout weights can be then updated by the following closed-form expression

$$W_{out} = ([S_0^T, \cdots, S_{N_U-1}^T])^+ \cdot [\hat{y}_0^T, \cdots, \hat{y}_{N_U-1}^T]^T, \quad (8)$$

where $(S)^+$ is the pseudo-inverse of matrix of $s(t)$. The learning operation can be summarized as in Algorithm 1.

Algorithm 1 DESN-Based MIMO-OFDM Symbol Detection

Data: $x(t)$
Result: $\hat{y}(t)$
initialization;
for $l \leftarrow 0$ **to** $l-1$ **do**
  Generate the state matrix based on (5):
  $s^{(l)}(t) = f(x^{(l)}(t) \cdot W_{in}^{(l)} + s^{(l)}(t-1) \cdot W_{res}^{(l)}$
end
return $s^{(l)}(t)$;
Calculate the output matrix according to (6):
$y^{(l)}(t) = s^{(l)}(t) \cdot W_{out}^{(l)}$;
Determine the loss between outputs:
$loss = \left\| y(t) - \hat{y}(t) \right\|_2^2$;
Minimize the L2 norm distance according to (7):
$loss\_min = \min \sum_0^{N_U-1} loss$;
Update output weights according to (8):
$W_{out} = (S)^+ \cdot (\hat{y}^T)^T$;

B. Reservoir Layer on a Memristive Crossbar

A generic model of the reservoir layer within our DESN is deployed on a memristive crossbar, as depicted in Fig. 5. As discussed in the previous subsection, the internal state of the hidden reservoir layer at the present time step can be written as in (5), while the output state is expressed as in (6). In the mathematical point of view, such operation can be realized by the sum-of-product computation. By mapping the sequential input to analogue voltage and weighted elements to conductance, such sum-of-product computation can be implemented by a memristive crossbar.

As depicted in Fig. 5, the generic reservoir layer contains two crossbar arrays, where the major crossbar determines the internal state of the network while the output crossbar computes the desired output. The major crossbar can be further divided into three groups of memory cells, which represent the fully-connected W_{in}, the sparsely-connected W_{res}, and the fully-connected W_{fb}, respectively.

During the operation, the input, $x(t)$, represented by the analogue voltage, is applied to horizontal word-lines of the crossbar. Consequently, an intermediate current is generated at each vertical bit-line by multiplying the input voltage and the conductance of the corresponding ReRAM cell as explicated in (4). Similarly, the corresponding current signal within the reservoir layer and the feedback unit can be computed by adopting the feedback signal from the previous internal state and output state of the network, and thus, the total j-th bit-line

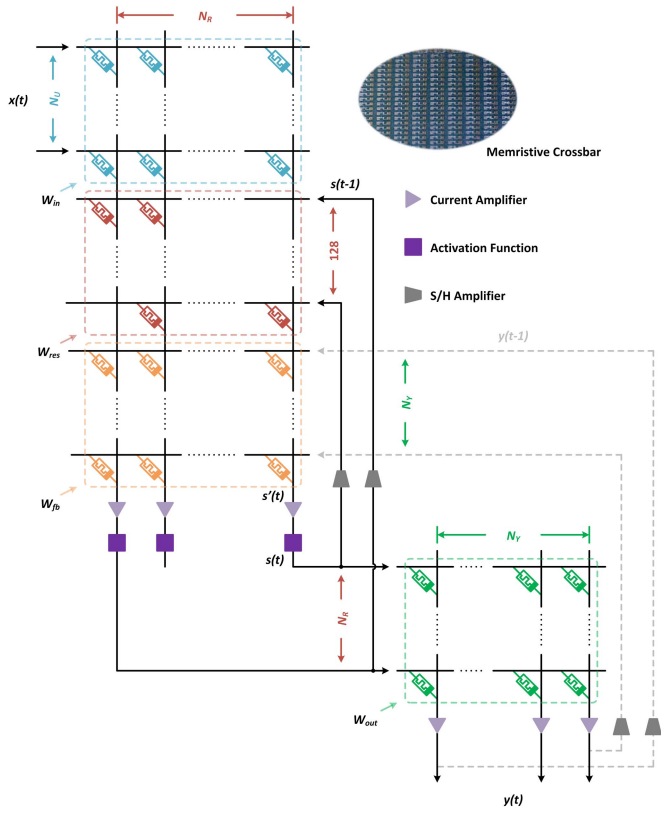


Fig. 5. Deploying the hidden reservoir layer on a memristive crossbar.

current generated from the reservoir layer at the present time step can be defined as

$$s'_j(t) = \sum_{i=1}^{N_U} V_i(t) \cdot G_{ij} + \sum_{i=N_U+1}^{N_R} V_i(t-1) \cdot G_{ij} + \sum_{i=N_R+1}^{N_Y} V_i(t-1) \cdot G_{ij}. \quad (9)$$

The state of the network, $s'(t)$, in the format of analogue current, is accumulated in the linear current amplifier with the inlaid current-to-voltage converter. The converted voltage output is then projected onto a higher dimensional space through the nonlinear activation function. Consequently, the output from the reservoir layer can be then determined by multiplying the transferred state of the network, $s(t)$ in the analogue voltage domain, and the output crossbar. To establish recurrent connections from the internal state and the output state of the network, $s(t)$ and $y(t)$ are fed back to the major crossbar respectively through the sample/hold amplifier. Such $y(t)$ is also used as the input for the following layer.

Our DESN closely emulates the recurrent connections as required by the reservoir layer with the in-memory computing capability relied on the memristive crossbar. In our hardware implementation, each element in the crossbar is composed of the discrete ReRAM cell [34], where the resistance range $R_{mem} \in [20\text{k}\Omega, 1\text{M}\Omega]$. In practice, the ReRAM cell is known to have large device-to-device and cycle-to-cycle variations as the system is scaled up [35]. In order to properly preserved

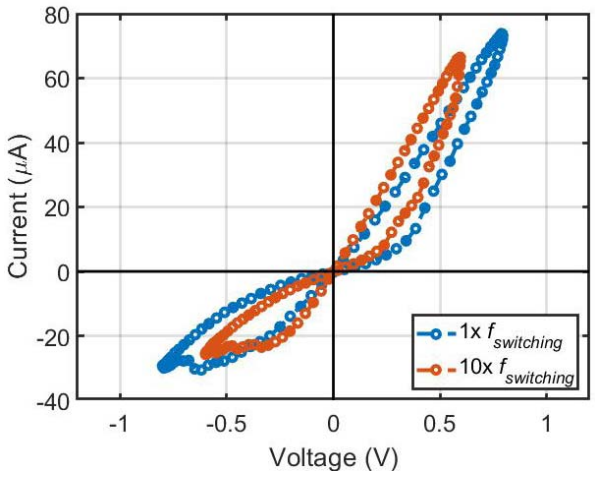


Fig. 6. Bipolar switching behavior of ReRAM cell.

the system performance and the inference accuracy, only the binary weight, represented by HRS and LRS of the ReRAM cell, with a large resistance ratio ($\frac{HRS}{LRS} \approx 50$) is applied in our hardware implementation [36].

The bipolar switching behavior of the discrete ReRAM cell is depicted in Fig. 6. By gradually increasing the potential across the ReRAM cell, the current jumps abruptly at a positive voltage of 0.4V, switching the ReRAM cell from HRS to LRS. On the other hand, the ReRAM cell switches from LRS to HRS at a negative voltage of −0.4V, and fully resets at a negative voltage of −0.8V. By directly processing the information extracted from weighted elements through the crossbar with a linear current amplifier, such in-memory computing capability can be achieved.

C. Linearity of Sum-of-Product Computation

Recently, the voltage sensing amplifier with a fixed resistor [37] and the spike sensing neuron [38] are the most widely used sensing methodology to read out the valuable information from the crossbar. However, the accuracy of the sensing network is degraded by the fixed resistor and the output headroom of the voltage sensing amplifier; additionally, the nonlinear behavior occurs in the spike sensing neuron due to the intrinsic delay within the membrane capacitance.

Within each hidden reservoir layer of our DESN, a linear current amplifier with the inlaid current-to-voltage converter is implemented, as shown in Fig. 7. During the operation, the transistor M_1 accumulates the total current from the input, I_{in}, and the reference current from the current source, I_{r1}, such that, $I_{M1} = I_{in} + I_{r1}$. An operational amplifier with the transistor M_3 create a negative feedback, allowing the voltage of V_{r+} keeps tracking the variation of input voltage, V_{r-}, and dynamically regulate the driving voltage of transistor M_3. With a 1:1 design ratio between M_1 and M_2, the current through the transistor M_3 can be denoted as $I_{M3} = I_{M2} - I_{r2} = I_{in}$. The output current mirror duplicates the buffered current, I_{M3}, to the transistor M_4 and consistently converts into a voltage signal through the loading transistor M_L.

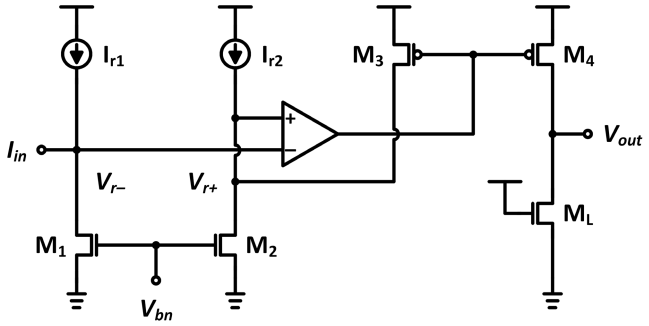


Fig. 7. Design scheme of linear current amplifier with inlaid current-to-voltage converter.

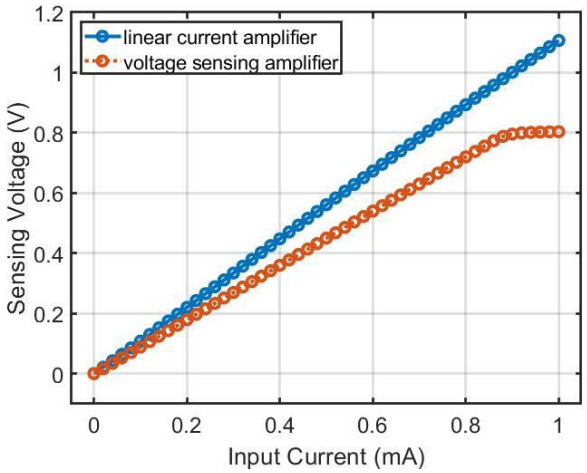


Fig. 8. Linearity of the linear current amplifier with inlaid current-to-voltage converter compared to conventional voltage sensing methodology.

Such linear current amplifier isolates the sum-of-product computation in the memristive crossbar and the current-to-voltage conversion in the current amplifier, and thus, computation result in the crossbar cannot be distorted, and the linearity as well as the stability during the current-to-voltage conversion can be preserved. In general, the optimal goal of implementing the linear current amplifier is to minimize the output voltage variation under various input current. To demonstrate such functionality, the input current, collected from the bit-line of the crossbar with a range of 0 to 1mA, was applied. As plotted in Fig. 8, it can be observed that the linear correlation between the input current and the output voltage can be obtained. It is reasonable to conclude that the implemented linear current amplifier is capable of providing a stable and accurate current-to-voltage conversion compared to the conventional voltage sensing amplifier.

D. Nonlinear Behavior of Network Transition

The nonlinear transition is typically introduced between synapses in neural network designs, allowing such networks to learn and model a complex arbitrary function between inputs and outputs. As both sigmoid and hyperbolic tangent functions are suffered from the vanishing gradient problem [39], the rectified linear unit (ReLU) has become the most widely used activation function in recent neural network designs [40]. However, the ReLU function does not contain the timing coefficient to model the time domain computation required by RNNs. Originating from a biological perspective, the Mackey-Glass (MG) equation [41] defines a feedback system in which dynamics depend on both current and previous states. The explicit representation of MG equation can be written as

$$X_{out} = \frac{\alpha \cdot X_{in}}{1 + \tau^n \cdot X_{in}^n}, \tag{10}$$

where α is the arbitrary design parameter that define the scaling factor of the equation, n and τ are the nonlinear and timing coefficients, respectively.

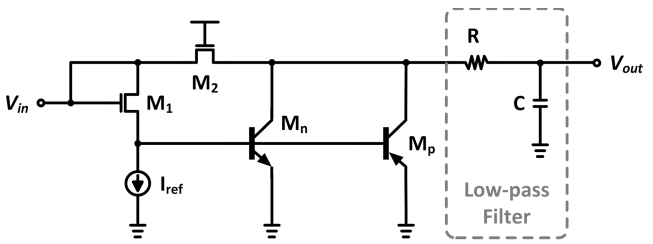


Fig. 9. Analogue circuit model of MG nonlinear activation function.

Fig. 9 illustrates the analogue circuit model of the MG equation. In general, the nonlinear characteristic of the MG equation can be formed by controlling the switching conditional of a n-type switch, M_n, and a p-type switch, M_p. During the operation, M_p fully turns on to reduce charges from the low-pass filter under the condition of $V_{in} < V_{th,p}$, where $V_{th,p}$ is the threshold voltage of M_p; as such, the voltage across the low-pass filter remains at zero. If $V_{th,p} < V_{in} < V_{th,n}$, where $V_{th,n}$ is the threshold voltage of M_n, charges from the input source accumulate in the low-pass filter, and thus, the voltage across the low-pass filter follows the input voltage. Under the condition of $V_{in} > V_{th,n}$, M_n fully turns on to reduce charges from the low-pass filter again, such that the voltage across the low-pass filter remains at zero. To accurately model the explicit representation of the MG equation, the transistor M_2 is implemented to serve as the scaling parameter in the circuit point of view. The non-linearity of the signal can be turned by the aspect ratio of M_n and M_p, while the timing coefficient can be adjusted by the reference current source, I_{ref}.

To demonstrate the nonlinear characteristic of the designed circuitry, a sequential bias voltage was applied as the input with the range of -1.8V to 1.8V. By gradually increasing the bias voltage, the nonlinear characteristic of the MG equation was recorded together with the corresponding numerical fit, as plotted in Fig. 10. It can be observed that the designed analogue circuit model of the MG equation fits the ideal MG equation with the scaling parameter, the nonlinear and the timing coefficients of $\alpha = 1$, $n = 0.4$ and $\tau = 6$, respectively.

E. Power Analysis

Our introduced neural network design is implemented through the standard 180nm CMOS process. In the experiment of power analysis, total of 128 neurons were implemented

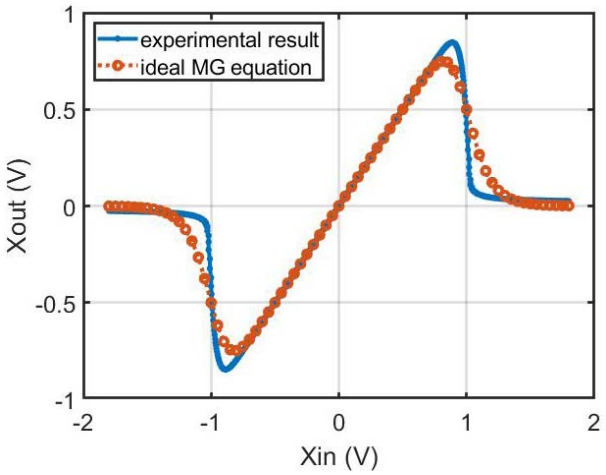


Fig. 10. Experimental result on analogue circuit model of MG nonlinear activation function together with the corresponding numerical fit.

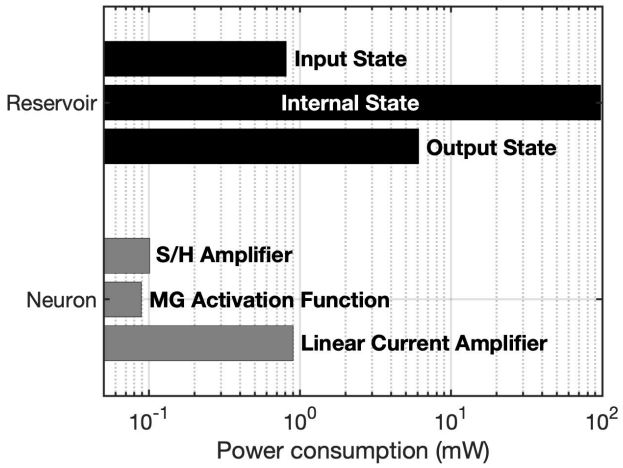


Fig. 11. Average power distribution of a single reservoir unit and a single neuron.

for the major crossbar, while 8 neurons were built for the output crossbar on a single hidden reservoir layer. The power consumption of the hybrid memristor-CMOS co-design was then simulated through the Cadence Virtuoso platform with the sampling frequency at 1MHz. The power distribution of the implemented reservoir layer is illustrated in Fig. 11. The total power of a single reservoir layer reaches 104.51mW, where the input state of the network consumes 0.81mW of total power, the internal state of the network absorbs 97.6mW of total power, and the rest are occupied by the output state of the network. Within each neuron, the sample/hold amplifier requires 9% of the reservoir's power, the analogue circuit model of MG nonlinear activation function absorbs 8% of the reservoir's power, and the rest are occupied by the linear current amplifier. The design specification of the implemented reservoir layer with state-of-the-art ESN implementations are summarized in Table I.

In recent years, several application specific integrated circuit (ASIC) implementation methodologies have introduced to accelerate the operation and reduce the energy cost for

TABLE I

COMPARISON OF INTRODUCED MEMRISTIVE-BASED RESERVOIR LAYER WITH STATE-OF-THE-ART ESN DESIGNS

	[43]	[44]	This Work
Algorithm	ESN	ESN	DESN
Learning	off-chip	off-chip	off-chip
Architecture	in-memory	in-memory	in-memory
CMOS Process	45nm	130nm	180nm
# of Neurons	30	1000	128
Supply Voltage	0.55V	1.2V	1.8V
Power Consumption	125.36mW	N/A	104.51mW

the MIMO detection through the pure CMOS implementation [42]. However, the accuracy of the detection is toughly affected by the performance of power amplifier, low noise amplifier, and signal demodulator. Our neural network-based symbol detector, on the other hand, significantly reduces the power overhead and noise interference caused by conventional RF components, but still, achieving a low BER performance.

V. APPLICATION EVALUATION

A. Experimental Setup

The preferment of our DESN are evaluated through a benchmark on a nonlinear system prediction task as well as the application on the MIMO-OFDM symbol detection. To demonstrate the robustness and reliability of our system, experimental results from our DESN are compared to the baseline model of shallow ESN and state-of-the-art neural network designs. During the evaluation, the number of neurons for each reservoir layer was kept at 128 on both experiments. For the OFDM symbol detection task on a 5G MIMO system, the number of transmitting and receiving antennas was set to be 4 and the number of sub-carriers in the OFDM system was set to be 1024.

B. Nonlinear System Prediction

The experiment was initially carried out through the tenth-order nonlinear auto-regressive moving average system (NARMA-10) benchmark [50], which can be governed by

$$o(t) = \beta \cdot o(t-1) + \gamma \cdot o(t-1) \cdot \sum_{k=0}^{9} o(t-k) + \delta \cdot d(t-9) \cdot d(t) + \epsilon \quad (11)$$

where $d(t)$ is the random input signal at time t; $o(t-1)$ is the output at the previous time step; β, γ, δ, and ϵ are random design parameters that would be replaced with a new random values taken from a $\pm 50\%$ interval around the respective original constants for every 2000 steps. In this experiment, the initial condition of design parameters were set to be $\beta = 0.3$, $\gamma = 0.05$, $\delta = 1.5$, and $\epsilon = 0.1$. Total of 10 thousand sampling points were generated for training and inference. 100 samples were used for the initialization, while 5900 samples were used for the training. The predicted output was automatically generated once the training operation is completed, and compared to target output. The prediction

TABLE II
COMPARISON OF INFERENCE ERROR ON NARMA-10 BENCHMARK TO STATE-OF-THE-ART NEURAL NETWORK DESIGNS

	Structure	Inference Error
[45]	Time-delay Reservoir	0.464
[46]	Time-delay Reservoir	0.17
[47]	Deep ESN	0.1573
[48]	Deep ESN	0.0832
[49]	Time-delay Reservoir	0.0683
This Work (baseline)	Shallow ESN	0.0578
This Work (DESN)	Deep ESN	0.045

error was examined using the normalized mean square error (NMSE), which can be express as

$$NMSE = \left(\frac{|y_i - \hat{y}_i|}{|\hat{y}_i|}\right)^2, \tag{12}$$

where y_i and $\hat{y}_i$ represent the target output and predicted output, respectively. Inference error on our DESN is compared to our baseline shallow ESN model and state-of-the-art neural network designs, as summarized in Table II. Compared to the baseline model, it can be observed that our DESN exhibits 22.15% accuracy improvement in the nonlinear system prediction task. Moreover, the NMSE on our DESN yields 2.33X $\sim$ 10.31X reduction on inference error compared to state-of-the-art neural network designs.

C. Symbol Detection

To further demonstrate the performance, our DESN is used as the symbol detector in the receiving chain of a 5G network. The analog waveform of received MIMO-OFDM signals are directly fed into our DESN. Through the learning operation, readout weights of our DESN are adjusted to generate the desired output, which is the transmitted MIMO-OFDM signals. In this experiment, the MIMO-OFDM signal used for the training is generated according to the 5G NR specification that follows the standard 3GPP TS 38.212 version 15.2.0 [51], where the channel is generated according to the Winner II channel model [52]. The modulation method is configured as 16-quadrature amplitude modulation (16-QAM). Specifically, pilots of communications system, which are utilized for channel estimation, are evenly used as in the training set, offering a compatible way to replace the state-of-the-art receiving process to the neural network-based ones.

The inference BER of our DESN is shown in Fig. 12 compared to the state-of-the-art model-based and neural network-based techniques. The LMMSE is a classic model-based approach using the linear processing method for symbol detection. Such method requires the knowledge of the noise variance of the channel. However, the LMMSE method relies on accurate channel information, which is challenging to be obtained in the low signal-to-noise ratio (SNR) regime. Comparing to the reported average BER of 11.02×10^{-2} of the LMMSE approach, the BER from our DESN-based approach is 5.76×10^{-2}, which is 47.73% more accurate. The inference BER from our DESN is also compared to the multilayer perception (MLP) model with three hidden layers and 1024 neurons per layer. Due to the limited training data, the MLP-based approach has an average inference BER

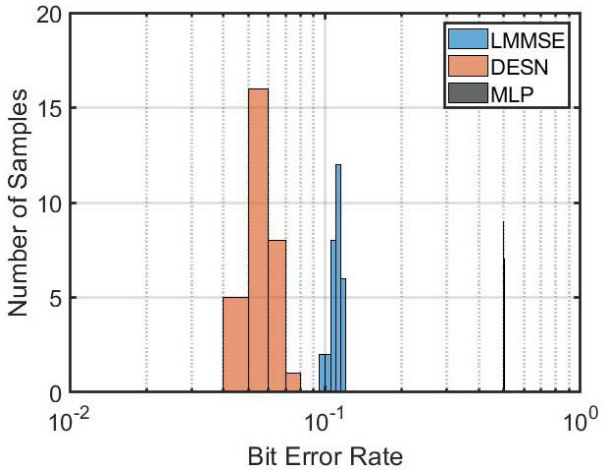


Fig. 12. Inference bit error rate with respect to various symbol detectors.

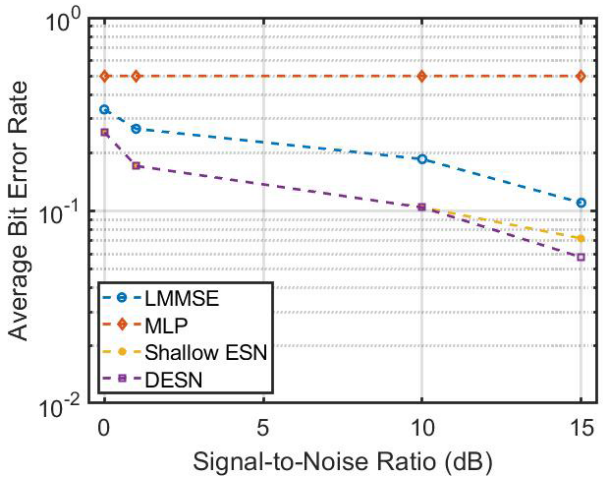


Fig. 13. Inference bit error rate versus the signal-to-noise ratio with respect to various symbol detectors.

of 50.12×10^{-2}. As such, it is convincing that our DESN outperforms state-of-the-art symbol detection techniques.

The average BER under various SNR scenarios for the MIMO-OFDM system is plotted in Fig. 13. It can be observed that our DESN beats the classic model-based LMMSE symbol detection for all SNR regimes. Furthermore, our DESN performs very close to the shallow ESN when the SNR is below 10dB, and demonstrates a lower BER starting at 10dB and beyond. Most importantly, our baseline shallow ESN and DESN do not require the statistical channel information, where the model-based LMMSE symbol detector does. Fig. 15 illustrates the inference BER with respect to various model of ESN. Compared to the shallow ESN, which contains only one reservoir layer, our DESN demonstrate a lower inference BER. Intuitively, such improvement can be interpreted as latter reservoir layers further increase the detection based on the processed observation from any previous reservoir layers. As shown in Fig. 14, our DESN with MG activation function has similar inference BER compared to the one with hyperbolic tangent function; however, more samples are distributed in the low BER regime with our DESN.

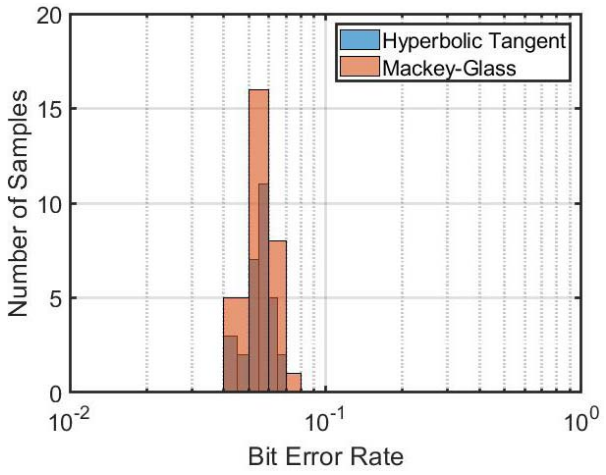


Fig. 14. Inference bit error rate with respect to various activation functions.

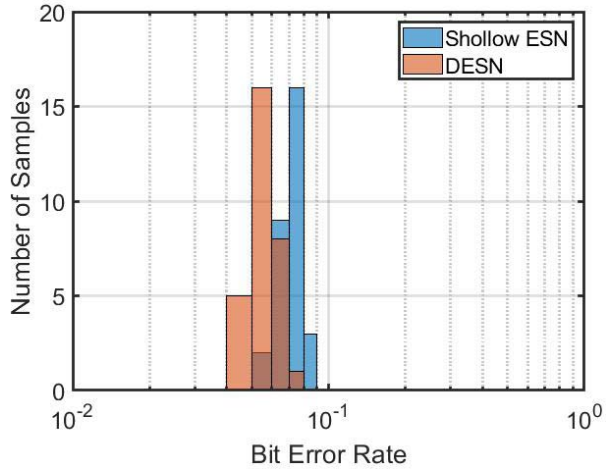


Fig. 15. Inference bit error rate with respect to various module of ESN.

VI. Conclusion

In this paper, we demonstrate a DESN with embedded memristive synapses, facilitating the in-memory computing capability. By fashioning multiple reservoir layers in a hierarchical processing structure, our DESN enables the learning behavior on both temporal and spatial domains. Experimental results on the hybrid memristor-CMOS co-design offers the necessary nonlinear computation required by each reservoir layer with merely 104.51mW of power consumption. Experimental results on nonlinear system prediction task achieve an error rate as low as 0.045, exhibiting 22.15% improvement compared to the shallow ESN and 2.33X $\sim$ 10.31X improvement compared to state-of-the-art ESN designs. Moreover, through the symbol detection task on a 5G MIMO-OFDM system, our DESN demonstrates an average BER of 5.76×10^{-2}, which is 47.73% more precise compared to state-of-the-art techniques in the literate for 5G networks.

References

[1] S.-Y. Lien, S.-L. Shieh, Y. Huang, B. Su, Y.-L. Hsu, and H.-Y. Wei, "5G new radio: Waveform, frame structure, multiple access, and initial access," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 64–71, 2017.

[2] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[3] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. Auerbach Publications, 2016.

[4] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, Aug. 2009.

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

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015, *arXiv:1506.06579*. [Online]. Available: http://arxiv.org/abs/1506.06579

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note," German Nat. Res. Center Inf. Technol., Bonn, Germany, GMD Rep. 148, 2001. vol. 148, no. 34, p. 13.

[10] W. Maass, P. Joshi, and E. D. Sontag, "Computational aspects of feedback in neural circuits," *PLoS Comput. Biol.*, vol. 3, no. 1, p. e165, 2007.

[11] L. Chua, "Memristor—The missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.

[12] T. Hasegawa *et al.*, "Learning abilities achieved by a single solid-state atomic switch," *Adv. Mater.*, vol. 22, no. 16, pp. 1831–1834, Apr. 2010.

[13] U. Ramacher and C. von der Malsburg, *On the Construction of Artificial Brains*. Berlin, Germany: Springer, 2010.

[14] H. Zhang, G. Chen, B. C. Ooi, K.-L. Tan, and M. Zhang, "In-memory big data management and processing: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 7, pp. 1920–1948, Jul. 2015.

[15] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, 4th Quart., 2015.

[16] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[17] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[18] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[19] J. Xu, X. Tao, and P. Zhang, "Analytical SER performance bound of M-QAM MIMO system with ZF-SIC receiver," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 5103–5107.

[20] S. Sarkar, "An advanced detection technique in MIMO-PSK wireless communication systems using MMSE-SIC detection over a Rayleigh fading channel," *CSI Trans. ICT*, vol. 5, no. 1, pp. 9–15, Mar. 2017, doi: 10.1007/s40012-016-0137-5.

[21] H. S. Kim, W. Zhu, J. Bhatia, K. Mohammed, A. Shah, and B. Daneshrad, "A practical, hardware friendly MMSE detector for MIMO-OFDM-Based systems," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, Dec. 2008, Art. no. 267460, doi: 10.1155/2008/267460.

[22] T.-H. Kim and I.-C. Park, "High-throughput and area-efficient MIMO symbol detection based on modified Dijkstra's search," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1756–1766, Jul. 2010.

[23] B. Gestner, W. Zhang, X. Ma, and D. V. Anderson, "Lattice reduction for MIMO detection: From theoretical analysis to hardware realization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 4, pp. 813–826, Apr. 2011.

[24] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[25] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.

[26] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," 2019, *arXiv:1906.04610*. [Online]. Available: http://arxiv.org/abs/1906.04610

[27] S. S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4694–4708, Oct. 2018.

[28] R. Shafin *et al.*, "Realizing green symbol detection via reservoir computing: An energy-efficiency perspective," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[29] Z. Zhou, L. Liu, and H.-H. Chang, "Learning for detection: MIMO-OFDM symbol detection through downlink pilots," *IEEE Trans. Wireless Commun.*, to be published.

[30] X. Liu, M. Chen, C. Yin, and W. Saad, "Analysis of memory capacity for deep echo state networks," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 443–448.

[31] C. Gallicchio, A. Micheli, and L. Pedrelli, "Design of deep echo state networks," *Neural Netw.*, vol. 108, pp. 33–47, Dec. 2018.

[32] K. Hamedani, Z. Zhou, K. Bai, and L. Liu, "The novel applications of deep reservoir computing in cyber-security and wireless communication," in *Intelligent System and Computing*. London, U.K.: IntechOpen, 2019.

[33] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 177–191, Jan. 2020.

[34] T. W. Molter and M. A. Nugent, "The generalized metastable switch memristor model," in *Proc. 15th Int. Workshop Cellular Nanosc. Netw. Appl. (CNNA)*, 2016, pp. 1–2.

[35] B. J. Choi *et al.*, "Electrical performance and scalability of pt dispersed SiO2 nanometallic resistance switch," *Nano Lett.*, vol. 13, no. 7, pp. 3213–3217, Jul. 2013.

[36] K. Bai, Q. An, L. Liu, and Y. Yi, "A training-efficient hybrid-structured deep neural network with reconfigurable memristive synapses," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 1, pp. 62–75, Jan. 2020.

[37] M. Hu, H. Li, Q. Wu, and G. S. Rose, "Hardware realization of BSB recall function using memristor crossbar arrays," in *Proc. 49th Annu. Design Autom. Conf.*, 2012, pp. 498–503.

[38] C. Liu *et al.*, "A spiking neuromorphic design with resistive crossbar," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.

[39] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018, *arXiv:1811.03378*. [Online]. Available: http://arxiv.org/abs/1811.03378

[40] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[41] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, Jul. 1977.

[42] G. Peng, L. Liu, S. Zhou, S. Yin, and S. Wei, "A 1.58 Gbps/W 0.40 Gbps/mm^2 ASIC implementation of MMSE detection for 128×8 64 -QAM massive MIMO in 65 nm CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 5, pp. 1717–1730, May 2018.

[43] D. Kudithipudi, Q. Saleh, C. Merkel, J. Thesing, and B. Wysocki, "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing," *Frontiers Neurosci.*, vol. 9, p. 502, Feb. 2016.

[44] A. M. Hassan, H. H. Li, and Y. Chen, "Hardware implementation of echo state networks using memristor double crossbar arrays," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2171–2177.

[45] A. Goudarzi, M. R. Lakin, and D. Stefanovic, "Reservoir computing approach to robust computation using unreliable nanoscale networks," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.* Berlin, Germany: Springer, 2014, pp. 164–176.

[46] S. Ortín and L. Pesquera, "Reservoir computing with an ensemble of time-delay reservoirs," *Cognit. Comput.*, vol. 9, no. 3, pp. 327–336, Jun. 2017.

[47] X. Sun, T. Li, Q. Li, Y. Huang, and Y. Li, "Deep belief echo-state network and its application to time series prediction," *Knowl.-Based Syst.*, vol. 130, pp. 17–29, Aug. 2017.

[48] X. Sun, T. Li, Y. Li, Q. Li, Y. Huang, and J. Liu, "Recurrent neural system with minimum complexity: A deep learning perspective," *Neurocomputing*, vol. 275, pp. 1333–1349, Jan. 2018.

[49] K. Bai and Y. Yi, "DFR: An energy-efficient analog delay feedback reservoir computing system for brain-inspired computing," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 14, no. 4, pp. 1–22, Dec. 2018.

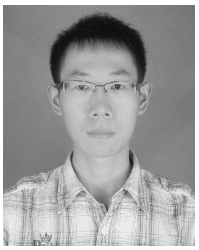
[50] P. Whitle, *Hypothesis Testing in time Series Analysis*, vol. 4. Stockholm, Sweden: Almqvist & Wiksells, 1951.

[51] *NR; Physical Channels and Modulation*, document 3GPP TS 38.211, Sep. 2019.

[52] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, "Winner ii channel models," in *Radio Technologies and Concepts for IMT-Advanced*. Hoboken, NJ, USA: Wiley, 2009, pp. 39–92.

Kangjun Bai (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from San Francisco State University (SFSU), San Francisco, USA, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, USA. His research interests include but are not limited to analog/mixed-signal integrated circuit design, neuromorphic computing, machine intelligence, and emerging deep learning systems.

Yang Yi (Senior Member, IEEE) is an Associate Professor with the Bradley Department of Electrical Engineering and Computer Engineering, Virginia Tech. Her research interests include very large scale integrated (VLSI) circuits and systems, computer-aided design (CAD), neuromorphic architecture for brain-inspired computing systems, and low-power circuits design with advanced nano-technologies for high-speed wireless systems.



Zhou Zhou (Student Member, IEEE) received the B.S. degree in communications engineering from the University of Electronic Science and Technology of China (UESTC) in 2011. Since 2018, he has been with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, as a Research Assistant. His current research interests are in the broad area of neural networks, machine intelligence, and wireless communications.

Shashank Jere (Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2014, and the M.S. degree in electrical engineering from the University of California at Los Angeles, Los Angeles, CA, USA, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Virginia Tech. From 2016 to 2019, he worked as a Platform and Product Development Engineer at Qualcomm Technologies Inc., San Diego, CA, USA. His current research interests are in the broad area of wireless communications, neural networks, and machine learning.

Lingjia Liu (Senior Member, IEEE) is an Associate Professor with the Bradley Department of Electrical Engineering and Computer Engineering, Virginia Tech. He is also the Associate Director of Wireless@VT. Prior to joining VT, he was an Associate Professor with the EECS Department, University of Kansas (KU). He spent more than four years working in the Mitsubishi Electric Research Laboratory (MERL) and the Standards and Mobility Innovation Laboratory, Samsung Research America (SRA), where he received the Global Samsung Best Paper Award in 2008 and 2010. He was leading Samsung's efforts on multiuser MIMO, CoMP, and HetNets in LTE/LTE-advanced standards. His general research interests mainly lie in emerging technologies for beyond 5G cellular networks, including machine learning for wireless networks, massive MIMO, massive MTC communications, and mmWave communications.