Detection Through Deep Neural Networks: A Reservoir Computing Approach for MIMO-OFDM Symbol Detection

Kangjun Bai, Lingjia Liu, Zhou Zhou, and Yang Yi
Virginia Tech, Blacksburg, VA
{kangjun,llj,zhou89,yangyi8}@vt.edu

ABSTRACT

The Reservoir Computing, a neural computing framework suited for temporal information processing, utilizes a dynamic reservoir layer for high-dimensional encoding, enhancing the separability of the network. In this paper, we exploit a Deep Learning (DL)-based detection strategy for Multiple-input, Multiple-output Orthogonal Frequency-Division Multiplexing (MIMO-OFDM) symbol detection. To be specific, we introduce a Deep Echo State Network (DESN), a unique hierarchical processing structure with multiple time intervals, to enhance the memory capacity and accelerate the detection efficiency. The resulting hardware prototype with the hybrid memristor-CMOS co-design provides the in-memory computing and parallel processing capabilities, significantly reducing the hardware and power overhead. With the standard 180nm CMOS process and memristive synapses, the introduced DESN consumes merely 105mW of power consumption, exhibiting 16.7% power reduction compared to shallow ESN designs even with more dynamic layers and associated neurons. Furthermore, numerical evaluations demonstrate advantages of the DESN over state-of-the-art detection techniques in the literate for MIMO-OFDM systems even with a very limited training set, yielding a 47.8% improvement against conventional symbol detection techniques.

KEYWORDS

deep learning, reservoir computing, echo state network, memristor crossbar, MIMO-OFDM, symbol detection

1 INTRODUCTION

Due to the "memory wall" phenomenon [1], deploying the general-purpose computing system for the spatial-temporal information processing has become inefficient in terms of the hardware implementation cost and the power consumption. Artificial Neural Networks (ANNs), a brain-inspired system which is intended to replicate the way that we humans learn, offer an alternative solution to accelerate the computational efficiency with witnessed remarkable progress [2]. ANNs are generally represented by a network of neuron-like processing units interconnected through synapse-like weighted elements, having advantages of finding similar patterns.

ICCAD '20, November 2–5, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8026-3/20/11…$15.00
https://doi.org/10.1145/3400302.3415722

The reservoir computing, a unified computing framework divided from the recurrent neural network (RNN), allows effective processing and learning of temporal information [3]. The major characteristic of the reservoir computing is that the connectivity of input weights and internal weights remains fixed at all times, and thus, training is not required. Similar as classical RNNs, the internal state of the network evolves dynamically across time to process sequential patterns over arbitrary time intervals. Since the training operation only involves in the readout stage, the computational cost of learning can be significantly reduced. Recent theoretical analyses demonstrate that the reservoir computing model can provide excellent performance in speech recognition [4] and wireless communication [5] tasks.

In modern wireless communication networks, an accurate characterization of the underlying wireless channel is typically needed at the receiver to detect the transmitted symbol [6]. However, the nonlinear distortion caused by practical Radio Frequency (RF) components in the transceiver chain and the interference introduced by wireless multi-access strategies can significantly impact the performance of channel estimation. ANNs, on the other hand, offer an alternative technique for transmitted symbol detection by inverse processing signals propagated through wireless channels. Based on the framework of supervised learning, neural networks can learn to reconstruct corrupted symbols from the aforementioned distortion, interference, and noise at the receiver. Furthermore, due to the nonlinear feature of communication signals, RNNs are expected to be the most suitable architecture in wireless systems.

In this work, we exploit a deep learning (DL)-based symbol detection technique for MIMO-OFDM systems by using a Deep Echo State Network (DESN) with memristive synapses. Major contributions of our work are summarized as followings:

- By concatenating multiple dynamic reservoir layers in a hierarchical processing structure, the introduced DESN offers a high-dimensional random encoding over multiple time intervals for sequential inputs, enhancing the separability and memory capacity of the network;
- The DL-based detection strategy significantly reduces the complexity of the receiver, accelerating the detection efficiency and increasing the robustness;
- The hybrid memristor-CMOS co-design enables in-memory computing and parallel processing capabilities, yielding 16.7% power reduction over shallow ESN designs even with more dynamic reservoir layers and associated neurons;
- Numerical evaluations on the high-speed transmitted symbol detection demonstrates advantages of the DESN over state-of-the-art techniques in the literate for MIMO-OFDM systems even with a very limited training data.

2 WIRELESS COMMUNICATION NETWORKS

In the contemporary society, communications between people or devices are interconnected by wireless networks. Unquestionable, the new level of performance and efficiency of 5G/beyond-5G communication networks will empower new user experiences with high throughput, ultra-reliable connection, ultra-low latency, and massive capacity [7]. However, conducting transmitted symbol detection, particularly for Multiple-input, Multiple-output Orthogonal Frequency-division Multiplexing (MIMO-OFDM) systems under heterogeneous environments and channel conditions, is one of the major challenges in 5G/beyond-5G wireless networks. In a MIMO-OFDM system, the received signal 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 [8]. Most importantly, more information through a modulation symbol can be conveyed as the size of the alphabet table increases.

In the conventional approach for the receiving operation, a channel estimation is firstly conducted, followed by detecting corresponding symbols within the coherence time based on its estimated channel. However, an accurate channel estimation results in a colossal resource consuming. Furthermore, there is a clear trade-off between the performance of the channel estimation and resources used for data transmission; in particular, a more accurate channel estimation often requires more resources to be allocated on this process, making less available resources for data transmission. To this end, it is crucial to study the symbol detection method with limited or even without available channel knowledge, especially for 5G/beyond-5G wireless networks.

Conventional symbol detection approaches for MIMO-OFDM systems, including the Maximum Likelihood (ML) [9] and the Minimum Mean Squared Error (MMSE) [10], often rely on modeling the feature of transmission channels and solving the formulated problem based on a particular model. The data-driven approach powered by the DL offers a solution to detection approaches without relying on such model-based assumptions. In particular, by formulating the symbol detection task as a classification problem, the DL-based symbol detector can be used to perform a parameter tuning based on the estimated Channel State Information (CSI) and received symbols [11, 12]. The complexity analysis of such DL-based symbol detector is carried out with considerations of RF impairments and noise interference, demonstrating a lower Bit Error Rate (BER) with less required resources [13].

3 DESIGN METHODOLOGY

3.1 Deep Echo State Network

Benefited by the supervised learning framework, the introduced DESN, as demonstrated in Fig. 1, can learn to reconstruct corrupted symbols from RF impairments, signal distortion, and noise interference. In general, the DESN contains three major computing layers, namely, the input layer, a hierarchy of stacked reservoir layers with intermediate input/output (I/Os), and the output layer. Crucially, the introduced DESN embeds the historical information into a dynamic state representation in each hidden reservoir layer, exploiting the temporal information in a feed-forward structure to enable the spatial-temporal processing characteristic.

During the computation, the analogue domain of real and imaginary components, representing a set of complex time-domain symbols of binary digits, are applied as the input signal, which can be defined as $u(t) = x^{(1)}(t) \in N_U$, where N_U is the input dimension. The association between the input layer and the first hidden reservoir layer is communicated through input weighted elements, $W_{in}^{(1)} \in [N_U \times N_R]$, where N_R is the number of neurons in each hidden reservoir layer. Through the hierarchical structure, each hidden reservoir layer adopts the intermediate output, generated from its previous layer, to update its internal state and compute the corresponding output for its following layer. By denoting the total number of hidden reservoir layers as N_L, the internal state of the l-th hidden reservoir layer can be written as

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

where $f()$ is a nonlinear activation function; $x^{(l)}(t) = y^{(l-1)}(t)$ represents the local input at the present time step, which equals to the local output from the previous layer; $W_{in} \in [N_U \times N_R]$, $W_{res} \in [N_R \times N_R]$, and $W_{fb} \in [N_Y \times N_R]$ denote input weights, internal weights within the reservoir, and feedback weights from the local output to the reservoir, respectively; $s^{(l)}(t-1)$ and $y^{(l)}(t-1)$ indicate the internal state of the l-th hidden reservoir layer and its output at the previous time step. The present output state of the l-th hidden reservoir layer can be then expressed as

$$y^{(l)}(t) = s^{(l)}(t) \cdot W_{out}^{(l)}, \quad (2)$$

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

By stacking multiple hidden reservoir layers into a hierarchical processing structure, the state computation and the learning operation are carried out through the pipeline. To reduce the design complexity, the teacher forcing for each hidden reservoir layer is the same. Correspondingly, the final output, $y^{(N_L)}(t)$, generated from the last hidden reservoir layer estimates the desired OFDM symbol through output weighted elements, $W_{out}^{(N_L)}$.

3.2 Learning Rule

In the training operation, weighted elements for W_{res} are randomly generated according to the echo state property [14], while only W_{out} is trained by minimizing the *L2* norm distance. For each computing cycle, the trajectory is computed by feeding the training input, $\{x(t)\}_{t=0}^{T}$, with the target symbol, $\{y(t)\}_{t=0}^{T}$, where T denotes the length of training patterns. The set of internal states, $\{s(t)\}_{t=0}^{T}$ is then obtained by Eq. (1). Output weights are then updated by minimizing the *L2* norm distance between the target output and the predicted output, which can be expressed as

$$\min_{W_{out}} \sum_{t=0}^{T} \|y(t) - s(t) \cdot W_{out})\|_2^2 . \quad (3)$$

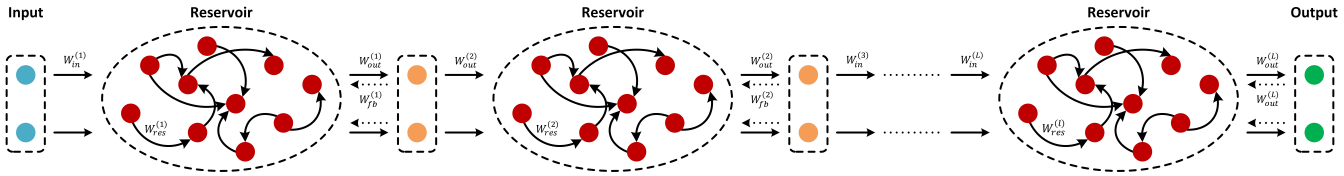

Figure 1: Architecture of Deep Echo State Network (DESN).

Algorithm 1: DL-based MIMO-OFDM Symbol Detection Strategy using DESN

Data: $x(t), y(t)$
Result: $\bar{y}(t)$
initialization
for $t \leftarrow 0$ **to** T **do**
  for $l \leftarrow 0$ **to** $l-1$ **do**
    Generate the state matrix according to Eq. (1):
    $s^{(l)}(t) = f(x^{(l)}(t) \cdot W_{in}^{(l)} + s^{(l)}(t-1) \cdot W_{res}^{(l)}$
  end
end
return $s(t)$
Calculate the output matrix according to Eq. (2):
$y(t) = s(t) \cdot W_{out}$
Determine the loss between outputs:
$loss = \|y(t) - s(t) \cdot W_{out})\|_2^2$
Minimize the $L2$ norm distance according to Eq. (3)
$loss\_min = \min \sum_0^T loss$
Update output weights according to Eq. (4):
$W_{out} = \bar{Y} \cdot \bar{S}^+$

The closed-form solution can be then determined as

$$W_{out} = \bar{Y} \cdot \bar{S}^+, \tag{4}$$

where $\bar{Y} = [\bar{y}_0, \cdots, \bar{y}_T]$, and $\bar{S}^+$ is the pseudo-inverse of matrix $\bar{S} = [\bar{s}_0^T, \cdots, \bar{s}_T^T]$. Since the symbol detection task can be formulated as a classification problem using the DL-based technique, the prediction operation based on the historical information is not required, and thus, the feedback computation with W_{fb} is eliminated in this design to further reduce the computation and power overhead. The general learning operation of the introduced DESN can be summarized in Algorithm 1.

3.3 Memory Capacity

Due to the recurrent nature of the reservoir layer, the reservoir state, $s(t)$, reflects traces of the historical information, known as the dynamic short-term memory. The memory capacity, represented the amount of variance that the delayed input can be recovered from optimally trained output units, is limited in a shallow ESN. To be specific, as the data density and the complexity of the application scale up, the learning capability of shallow ESN reduces. Based on the short-term memory definition [15], the determination coefficient with a single set of I/O can be expressed as

$$d(k, W_{out}) = \frac{cov^2(x(t-k), y(t))}{\sigma^2(x(t))\sigma^2(y(t))}, \tag{5}$$

where $cov()$ denotes the co-variance, $\sigma()$ represent the variance, and k is the delay coefficient. The short-term memory capacity of a hierarchical network can be then written as

$$MC = \sum_1^{\infty} \max_{W_{out}} d(k, W_{out}). \tag{6}$$

The general definition of short-term memory for stacked ESN with multiple I/Os can be obtained by extending the concept to each I/O pair. The introduced DESN with stacked hierarchy of dynamic reservoir layers achieves multiple temporal representation for input sequences, allowing such system to capture more features between input and output patterns, and most importantly, enhancing the richness of reservoir states and the memory capacity.

4 HARDWARE PROTOTYPING

4.1 Reservoir Layer with Memristive Synapses

A generic model of the reservoir layer is deployed on double-column memristive crossbars, as depicted in Fig. 2. As discussed in the previous section, the internal state of the reservoir layer at the present time step can be written as in Eq. (1), while the output state is expressed as in Eq. (2). In the mathematical point of view, such operation can be achieved by the sum-of-product computation. By mapping the sequential input to voltage and weighted elements to conductance, such sum-of-product computation can be implemented by a memristive crossbar. The introduced memristive-based reservoir layer contains two crossbars, in which 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 two groups of memristive cells, representing the fully-connected W_{in} and the sparsely-connected W_{res}. Since each weighted element can be either positive or negative, a double-column crossbar is implemented to represent a single weighted element; for instance, the conductance of a memristor cell, G_{ij}, representing the weighted element of W_{ij}, is expressed as $G_{ij} = G_{ij}^+ - G_{ij}^-$.

During the operation, the input, $x(t)$, represented by an analogue voltage, is applied to the horizontal word-line 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 memristor cell, which can be explicated as

$$I_j = I_j^+ - I_j^- = \sum_{i=1}^{n} V_i \cdot G_{ij}^+ - \sum_{i=1}^{n} V_i \cdot G_{ij}^-, \tag{7}$$

where V_i is the i-th input vector; n is the input dimension; I_j^+ and I_j^- represent the positive and negative bit-line current, respectively;

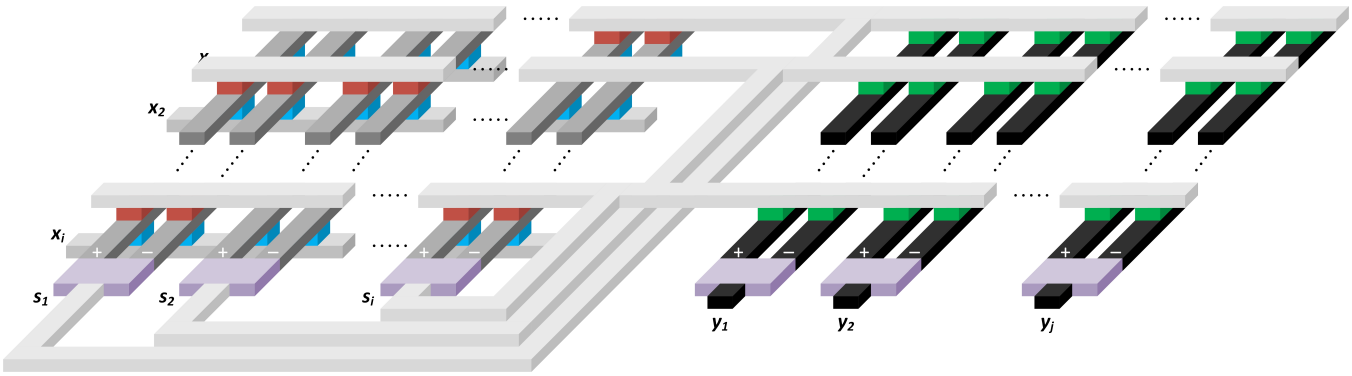

Figure 2: Deploying the reservoir layer on double-column memristive crossbars. W_{in} (highlighted in blue) and W_{out} (highlighted in green) are fully-connected, while W_{res} (highlighted in red) is sparsely-connected. "+" and "-" denote positive and negative weighted elements, respectively.

G_{ij}^{+} and G_{ij}^{-} denote the positive and negative conductance, respectively, of a menristor cell located between the i-th word-line and the j-th bit-line. Similarly, the corresponding current signal within the reservoir layer can be computed by adopting the feedback signal from the previous internal state, and thus, the total bit-line current generated from the reservoir layer (major crossbar) at the present time step can be defined as

$$\begin{aligned} s^{+} &= \sum_{i=1}^{N_U} V_i(t) \cdot G_{ij}^{+} + \sum_{i=N_U+1}^{N_R} V_i(t-1) \cdot G_{ij}^{+}, \\ s^{-} &= \sum_{i=1}^{N_U} V_i(t) \cdot G_{ij}^{-} + \sum_{i=N_U+1}^{N_R} V_i(t-1) \cdot G_{ij}^{-}. \end{aligned} \tag{8}$$

It can be observed that a subtraction operation is required for a double-column crossbar design; to support such feature, a bilateral linear current amplifier with the inlaid current-to-voltage converter is implemented. Two separated states of the network, s^{+} and s^{-}, in the format of analogue current, are accumulated in the bilateral linear current amplifier. s_{sum}, representing the difference between s^{+} and s^{-} in the format of analogue voltage, is generated from the bilateral linear current amplifier, which will be then projected onto a higher dimensional space through the Mackey-Glass (MG) activation function [16].

In neural network designs, the transition between synapses are often carried out through a nonlinear activation function, projecting inputs onto a higher dimensional space for prediction or classification. Recent research has found that both sigmoid and hyperbolic tangent functions, typical activation functions used in RNNs, are suffered by the vanishing gradient problem [17]. Originating from a biological perspective, the MG equation [18] defines a feedback system in which dynamics depend on both present and previous states, becoming the suitable candidate for RNN designs.

Similar as the reservoir layer (major crossbar), the output layer is also implemented by a double-column crossbar. As such, the output from the reservoir layer can be computed by multiplying the transferred state of the network and output weighted elements, which can be written as

$$y_j = y_j^{+} - y_j^{-} = \sum_{i=N_R+1}^{N_Y} s_{sum} \cdot (G_{ij}^{+} - G_{ij}^{-}). \tag{9}$$

With the hierarchy of stacked dynamic reservoir layers, the output generated from the present layer will be then used as the input for its following layer.

4.2 Modeling of Memristor

Due to the high access latency from/to memory units, the conventional computing architecture can no longer offer timely response [19]. Benefited from the memristive crossbar, the introduced DESN closely models the recurrent computation of the reservoir layer with in-memory computing and parallel processing capabilities, reducing the access latency between processing elements and data storage. In the hardware prototyping, each element in the crossbar is composed of the discrete Resistive Random-Access Memory (ReRAM)-based memristor cell [20]. The resistance range of the memristor cell is set to be 20kΩ to 1MΩ, where the according conductance $G \in [1\mu S, 50\mu S]$. Weighted elements are then updated through the offline training strategy.

It has been well known that the sneak path leakage is one of the major challenges in a memristor crossbar, representing an inevitably current flow through any unselected memristor cells, and thus, degrading the accuracy of the network [21]. Furthermore, memristors are known to have large device-to-device and cycle-to-cycle variations as the system is scaled up [22]. In this experiment, for properly predicting the system performance and inference accuracy, only the binary weight, represented by the high-resistance-state (HRS) and the low-resistance-state (LRS) of the memristor cell with a large resistance ratio ($\frac{HRS}{LRS} \approx 50$), is implemented.

4.3 Linearity of Sum-of-Product Computation

With the consideration of precise modeling, both positive and negative weighted elements are used to form a double-column crossbar; to accurately read out the information, a bilateral linear current amplifier is implemented, as shown in Fig. 3. During the operation,

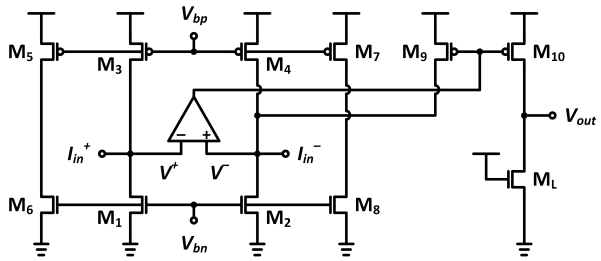

Figure 3: Design scheme of bilateral linear current amplifier with inlaid current-to-voltage converter.

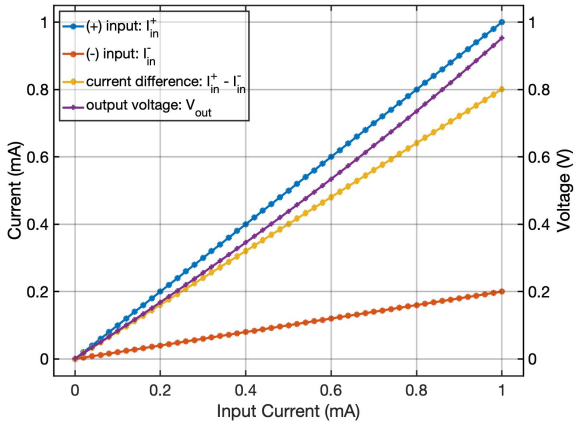

Figure 4: Characteristics of bilateral linear current amplifier with inlaid current-to-voltage conversion.

the transistor M_1 accumulates the total current from the positive input, I_{in}^+, and the reference current generated through the transistor M_3, as such, $I_{M1} = I_{in}^+ + I_{M3}$. This current is then duplicated across the associated current mirror, M_5 - M_8, and thus, $I_{M8} = I_{M6} = I_{M1}$. As the negative input current, I_{in}^-, injects into the transistor M_2, the associated current mirror forces M_2 to duplicate the current at M_1, as such, $I_{M2} = I_{M1}$. By balancing the current of I_{M1} and I_{M2}, the current through the transistor M_9 can be expressed as $I_{M9} = I_{in}^+ - I_{in}^-$, such that $I_{M2} = I_{M9} + I_{in}^- + I_{M4} = I_{in}^+ + I_{M4} = I_{M1}$. The high-gain operational amplifier keeps tracking the variation of V^+ and V^-, and dynamically regulate the driving voltage of M_9. The output current mirror, M_9 - M_{10}, duplicates the current difference to the output and consistently converts the current into a voltage through the loading transistor M_L.

In general, the optimal goal of implementing the bilateral linear current amplifier is to minimize the output voltage variation under various input current. To demonstrate such functionality, input currents, I_{in}^+ and I_{in}^- collected from the bit-line of the crossbar, was applied. As plotted in Fig. 4, it can be observed that the linear correlation between input currents and the output voltage can be obtained. It is reasonable to conclude that the introduced bilateral linear current amplifier is capable of providing a stable and accurate current subtraction and current-to-voltage conversion.

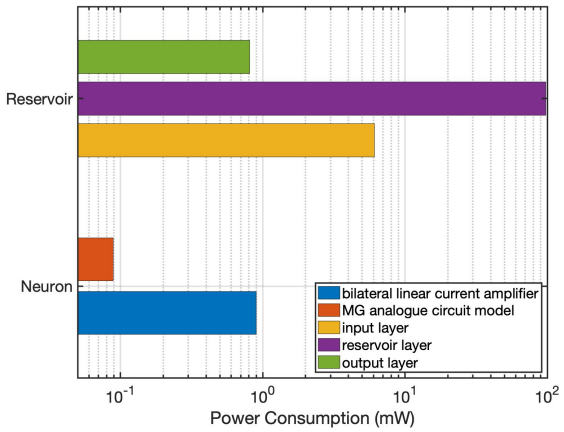

Figure 5: Average power distribution of a single dynamic reservoir layer and a single neuron.

Table 1: Comparison of introduced Memristive-based Reservoir Layer with State-of-the-art ESN Designs.

	[23]	[24]	This Work
Architecture	in-memory	in-memory	in-memory
Implementation	FPGA	MATLAB	CMOS
CMOS Process	45nm	N/A	180nm
# of Layers	1	1	2
# of Neurons	30	1000	128
Memory	memristor	memristor	memristor
Activation Function	tanh	tanh	MG
Supply Voltage	0.55V	N/A	1.8V
Power Consumption	125.36mW	N/A	104.51mW

4.4 Performance Metric

In the hardware prototype, analogue circuits were implemented with the standard 180nm CMOS process, and electronic synapses were built with discrete memristor cells. In this experiment, total of 128 neurons were implemented for the major crossbar, and 2 neurons were used for the input and output layers. The circuit model was simulated in both time-domain and frequency-domain through the Cadence Virtuoso platform to demonstrate its capability against noise. In a transient response, the signal-to-noise ratio (SNR) of the input was set to be 20dB, 10dB, and 5dB; compared to the scenario without noise, the average output error rate is found to be 2.49%, 4.09%, and 9%, respectively. In a frequency response, the circuit model is more robust against noise when the operating frequency is higher than 50kHz.

The power distribution of a single dynamic reservoir layer and neuron is illustrated in Fig. 5. The total power of the reservoir layer reaches 104.51mW, in which the input state of the network consumes 0.81mW of the total power, the internal state of the network absorbs 97.6mW of the total power, and the rest are occupied by the output state of the network. For each neuron, the MG analogue circuit model [25] absorbs 16% of the total power, and the rest are

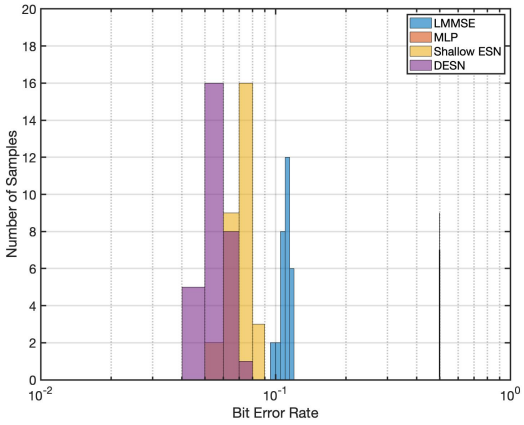

Figure 6: Testing bit error rate with respect to various symbol detection strategies.

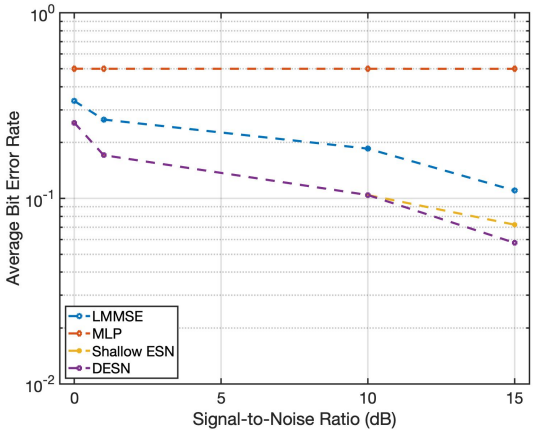

Figure 7: Testing bit error rate versus signal-to-noise ratio with respect to various symbol detection strategies.

occupied by the bilateral linear current amplifier. The resulting hybrid CMOS-memristor co-design of the dynamic reservoir layer is then compared to state-of-the-art ESN designs, exhibiting 16.7% power reduction over shallow ESN designs even with more dynamic layers and associated neurons, as summarized in Table 1.

5 MIMO-OFDM SYMBOL DETECTION

5.1 Experimental Setup

To demonstrate the performance, the introduced DESN is used as the symbol detector in the receiving chain of MIMO-OFDM systems. The analog waveform of received MIMO-OFDM signals are directly fed into the DESN. Through the learning operation, readout weights of the DESN are optimized 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 new radio (NR) specification that follows the standard 3GPP TS 38.212 version 15.2.0 [26], where the channel is generated according to the Winner II channel model [27]. The modulation method is configured as 16-Quadrature Amplitude Modulation (16-QAM). To be specific, pilots of the communication 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 DL-based strategies. Details of the system specification are set as followings: the number of transmitting and receiving antennas is set to be 4, the number of sub-carriers in the OFDM system is set to be 1024, and the number of neurons for each reservoir layer is set to be 128.

5.2 Result Discussion

The testing BER of the introduced DESN is shown in Fig. 6 and Fig. 7 with the comparison to the classic detection approach and state-of-the-art DL-based strategies. The Linear Minimum Mean Squared Error (LMMSE) is a classic model-based approach using the linear processing method for symbol detection. Such approach requires the knowledge of the noise variance of the channel. However, the LMMSE approach relies on accurate channel information, which is challenging to be obtained in the low SNR regime. Comparing to the reported average BER of 11.02×10^{-2} in the LMMSE approach, the average BER of the introduced DESN-based detection approach is 5.76×10^{-2}, which is 47.73% more accurate. The testing BER of the introduced DESN is also compared to the Multilayer Perception (MLP) model with three hidden layers and 1024 associated neurons. Due to the limited training set, the MLP approach has an average BER of 50.12×10^{-2}. Thereby, it is convincing that the introduced DESN outperforms state-of-the-art symbol detection strategies for all SNR regimes.

Furthermore, the introduced DESN with a hierarchy of stacked dynamic reservoir layers demonstrates a lower average BER compared to the shallow ESN design, which contains only one reservoir layer. Intuitively, such improvement can be interpreted as latter reservoir layers further increase the detection based on the processed observation from the previous reservoir layer.

6 CONCLUSIONS

In this paper, we exploit a DL-based symbol detection strategy for MIMO-OFDM systems. By concatenating multiple dynamic reservoir layers in a hierarchical processing structure, the introduced DESN enhances the separability and memory capacity of the network, capturing more features between input and output patterns. The resulting hybrid memristor-CMOS co-design enables in-memory computing and parallel processing capabilities, accelerating the computation efficiency and reducing the power overhead. Through the symbol detection task on MIMO-OFDM systems, the introduced DESN demonstrates an average BER of 5.76×10^{-2}, yielding a 47.8% improvement against classic symbol detection techniques even with a very limited training set.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. National Science Foundation (NSF) under grants CCF-1750450, ECCS-1811497, and CCF-1937487.

REFERENCES

[1] W. A. Wulf and S. A. McKee, "Hitting the memory wall: implications of the obvious," *ACM SIGARCH computer architecture news*, vol. 23, no. 1, pp. 20–24, 1995.

[2] J. M. Zurada, *Introduction to artificial neural systems*. West St. Paul, 1992, vol. 8.

[3] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[4] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A digital liquid state machine with biologically inspired learning and its application to speech recognition," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 11, pp. 2635–2649, 2015.

[5] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific reports*, vol. 2, p. 287, 2012.

[6] M. Speth, S. A. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using ofdm. i," *IEEE Transactions on communications*, vol. 47, no. 11, pp. 1668–1677, 1999.

[7] 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 communications magazine*, vol. 55, no. 6, pp. 64–71, 2017.

[8] S. Yang and L. Hanzo, "Fifty years of mimo detection: The road to large-scale mimos," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1941–1988, 2015.

[9] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE transactions on information theory*, vol. 48, no. 8, pp. 2201–2214, 2002.

[10] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "Near-maximum-likelihood detection of mimo systems using mmse-based lattice-reduction," in *2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577)*, vol. 2. IEEE, 2004, pp. 798–802.

[11] N. Samuel, T. Diskin, and A. Wiesel, "Deep mimo detection," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2017, pp. 1–5.

[12] S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets mimo-ofdm," *IEEE transactions on neural networks and learning systems*, no. 99, pp. 1–15, 2017.

[13] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[14] H. Jaeger, "Echo state network," *scholarpedia*, vol. 2, no. 9, p. 2330, 2007.

[15] ——, *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001, vol. 5.

[16] K. Bai, Q. An, L. Liu, and Y. Yi, "A training-efficient hybrid-structured deep neural network with reconfigurable memristive synapses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 1, pp. 62–75, 2019.

[17] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

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

[19] H. Zhang, G. Chen, B. C. Ooi, K.-L. Tan, and M. Zhang, "In-memory big data management and processing: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1920–1948, 2015.

[20] T. W. Molter and M. A. Nugent, "The generalized metastable switch memristor model," in *CNNA 2016; 15th International Workshop on Cellular Nanoscale Networks and their Applications*. VDE, 2016, pp. 1–2.

[21] M. Hu, H. Li, Q. Wu, and G. S. Rose, "Hardware realization of bsb recall function using memristor crossbar arrays," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 498–503.

[22] B. J. Choi, A. C. Torrezan, K. J. Norris, F. Miao, J. P. Strachan, M.-X. Zhang, D. A. Ohlberg, N. P. Kobayashi, J. J. Yang, and R. S. Williams, "Electrical performance and scalability of pt dispersed sio2 nanometallic resistance switch," *Nano letters*, vol. 13, no. 7, pp. 3213–3217, 2013.

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

[24] A. M. Hassan, H. H. Li, and Y. Chen, "Hardware implementation of echo state networks using memristor double crossbar arrays," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2017, pp. 2171–2177.

[25] K. Bai, Y. Yi, Z. Zhou, S. Jere, and L. Liu, "Moving toward intelligence: Detecting symbols on 5g systems through deep echo state network," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2020.

[26] 3GPP TS 38.211, "NR; Physical channels and modulation," September 2019.

[27] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, "Winner ii channel models," *Radio Technologies and Concepts for IMT-Advanced*, pp. 39–92, 2009.