Reservoir Computing With Dynamic Reservoir using Cascaded DNA Memristors

Xingyi Liu, *Student Member, IEEE*, and Keshab K. Parhi, *Fellow, IEEE*

***Abstract*—This article proposes molecular and DNA memristors where the *state* is defined by a single output variable. In past molecular and DNA memristors, the state of the memristor was defined based on two output variables. These memristors cannot be cascaded because their input and output sizes are different. We introduce a different definition of state for the molecular and DNA memristors. This change allows cascading of memristors. The proposed memristors are used to build *reservoir computing* (RC) models that can process temporal inputs. An RC system consists of two parts: *reservoir* and *readout layer*. The first part projects the information from the input space into a high-dimensional feature space. We also study the input-state characteristics of the cascaded memristors and show that the cascaded memristors retain the memristive behavior. The cascade connections in a reservoir can change *dynamically*; this allows the synthesis of a dynamic reservoir as opposed to a static one in the prior work. This reduces the number of memristors significantly compared to a static reservoir. The inputs to the readout layer correspond to one molecule per state instead of two; this significantly reduces the number of molecular and DSD reactions for the readout layer. A DNA RC system consisting of DNA memristors and a DNA readout layer is used to detect seizures from intra-cranial electroencephalogram (iEEG). We also demonstrate that a DNA RC system consisting of three cascaded DNA memristors and a DNA readout layer can be used to solve the time-series prediction task. The proposed approach can reduce the number of DNA strand displacement (DSD) reactions by three to five times compared to prior approaches.**

***Index Terms*—Cascaded memristors, DNA memristors, DNA reservoir computing, DNA readout layer, dynamic reservoir, seizure detection, time-series prediction.**

I. Introduction

REASONED from symmetry arguments, the original concept of memristors was described by Chua in 1971 [1], [2]. In 2008, a simple analytical example of a memristor was first introduced [3]. Memristors were defined by the non-linear functional relationship between the magnetic flux linkage and the amount of electric charge that has flowed [1], [2]. They are frequently utilized to build numerous machine learning systems, including in-memory computing systems [4], [5], [6] and reservoir computing (RC) system [7].

Manuscript received 26 April 2023; revised 19 June 2023 and 29 July 2023; accepted 2 September 2023. Date of publication 5 September 2023; date of current version 29 January 2024. This work was supported by the Army Research Office under Grant W911NF-21-1-0265. This paper was recommended by Associate Editor J.-S. Seo. *(Corresponding author: Keshab K. Parhi.)*

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: liux3138@umn.edu; parhi@umn.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TBCAS.2023.3312300.

Digital Object Identifier 10.1109/TBCAS.2023.3312300

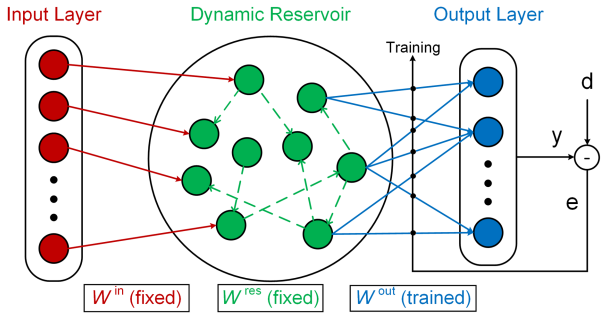

Fig. 1. Typical RC architecture [34]. The dynamic reservoir is made up of neurons that are randomly connected. During the training process, only the weights between the dynamic reservoir and the output layer need to be updated based on the difference (e) between the current output (y) and the desired output (d).

Since the original work by Adleman [8], DNA computing has advanced significantly. Multiple artificial intelligence and machine learning functions can be achieved using DNA [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. In 1982, DNA Hopfield neural network was presented [9]. Feed-forward network and winner-take-all network can be constructed by utilizing a DNA transcriptional switch [15]. A framework that can systematically map arbitrary linear threshold circuits into DNA strand displacement (DSD) cascades was presented in [19], [20]. In our prior work, we have shown that digital filters, support vector machines, and artificial neural networks can be synthesized using DNA strand displacement (DSD) reactions [21], [22], [23], [24]. Other emerging applications based on DNA include: DNA storage [25], [26], and gene editing [27].

Reservoir computing (RC) is a method of computing that evolved from the idea of recurrent neural network (RNN) theory [28], [29], [30]. It maps input signals into higher dimensional computational spaces by using the dynamics of a stationary, nonlinear system (reservoir) [29], [30], [31]. After feeding the input into the reservoir, a simple readout layer is trained to map the *state* of the reservoir to the desired output. Fig. 1 shows a typical RC system. Since training RC systems only involves training the weights between the dynamic reservoir and the output layer while the other weights remain fixed, the training process of RC is much faster than that of the conventional RNNs which requires computing backpropagation through time [32], [33]. This characteristic enables RC to be employed in edge devices with constrained computational resources.

Physical RC systems have been implemented by using a diffusive memristor-based reservoir or a dynamic memristor array [7], [35]. In [36], it has been shown that the perovskite halide-based memristor can be directly driven by emulated neural spikes. There is a growing interest in showcasing the viability of synthesizing DNA memristor-based RC systems. Implementation of molecular and DNA memristors was presented in our previous work [34]. However, the memristors presented in [34] operated in a standalone manner, i.e., the output of a memristor was not input to another memristor. These memristors are suitable for RC with a *static* reservoir. In this article, we further develop the theory of DNA memristors and DNA RC systems. In particular, we study the characteristics of *cascaded* memristors. We demonstrate the design of DNA RC systems using a dynamic reservoir consisting of cascaded memristors. These proposed cascaded molecular and DNA memristors retain the hysteresis behavior. The amount of memory of the memristor is affected by the number of memristors in the cascade.

It is important to point out that the molecular memristors are not intended to replace or compete with electronic memristors. These are intended to perform machine learning functions in synthetic biology applications. In general, the DSD reactions can sense and monitor proteins, perform machine learning operations, and trigger drug delivery, without requiring conversion to electronic format [37]. In this article, we apply memristors to detect seizures from electroencephalogram (EEG) signals using a *dynamic* DNA reservoir. Although the raw EEG signal is an electrical signal, in an actual synthetic biology application, the DNA reservoir may process the concentration of a certain protein or enzyme. DNA memristors may find applications in drug delivery, protein therapy, DNA storage, and gene editing.

This article makes four major contributions. First, we introduce a different definition of *state* for the molecular and DNA memristors. In this article, the state varies from 0 to 1 while in the prior article, it varied from -1 to 1. This change corresponds to encoding using unipolar fractional coding as opposed to bipolar. Second, we study the input-state characteristics of the cascaded memristors. Input-state characteristics of cascaded memristors have not been analyzed before. The cascade connections in a reservoir of an RC system can change dynamically; this allows the synthesis of a dynamic reservoir as opposed to a static one in the prior work. The synthesis of RC systems with dynamic reservoir based on memristors have not been presented before. This reduces the number of memristors significantly compared to a static reservoir. Third, the inputs to the readout layer are now one molecule per state instead of two; this significantly reduces the number of molecular and DSD reactions for the readout layer of the RC system. Fourth, via simulation using Mathematica, we demonstrate the use of a DNA RC system to detect seizures from intra-cranial EEG (iEEG) [38] using cascaded DNA memristors and a DNA readout layer. The connections between DNA memristors are randomly generated. We also demonstrate that a DNA RC system consisting of three cascaded DNA memristors and a DNA readout layer can be used to solve the time-series prediction task [7]. In the Supplementary Information, we demonstrate that the number of DSD reactions can be reduced by a factor of 3 to 4 for the digit recognition task from the MNIST dataset [39], compared to our prior approaches [34] and [40], respectively.

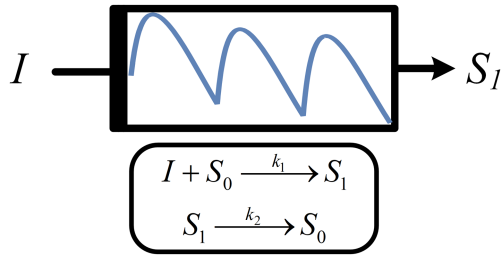

Fig. 2. Molecular memristor. The analog input of the memristor is encoded in the input molecule's (I) concentration. The analog state of the memristor is determined by the concentration of the output molecule (S_1). The parameters k_1 and k_2 represent the rate constants for the two molecular reactions, respectively.

This article is organized as follows. Section II describes the new implementation of molecular and DNA memristors where the state is described by a single molecule. We also study input-state characteristics of cascaded memristors. Section II also describes the implementation of molecular readout layers by combining multiple analog multiplications [37], [40], [41]. Transforming the molecular reactions to DNA is described in Section III. Section IV discusses the molecular and DNA implementations of RC systems and presents experimental results of the proposed architectures for the seizure detection task. Section V shows that the DNA memristor-based RC system can be used to solve a second-order nonlinear dynamical task. Finally, some conclusions and comparisons with previous work are given in Section VI.

II. Molecular Memristor and Reservoir Computing

Several studies have demonstrated the implementation of dynamic reservoir-based RC systems using dynamic memristor devices that exploit short-term memory effects [7], [40], [42], [43], [44]. The reservoir of an RC system should have the ability to project temporal input signals into different reservoir states which can then be further processed by a simple readout layer. Inspired by our previous work [40], we introduce a modified molecular implementation of a memristor and analyze its hysteresis property in this section. Then the dynamic reservoir of the RC system can be constructed by using the proposed molecular memristor [7], [40], [45], [46].

A. Molecular Memristors

A molecular memristor using fractional coding has been introduced in our previous work [40]. In that work, the molecular memristor is implemented using the same two reactions as shown in Fig. 2. The analog input of the memristor is encoded in the input molecule's (I) concentration. The bipolar state of the memristor is given by $S = ([S_1] - [S_0])/([S_1] + [S_0])$ where $[S_1]$ and $[S_0]$ are the concentrations of S_1 and S_0, respectively. Notice the proposed two molecular reactions always convert S_0 and S_1 to the same amount of S_1 and S_0, respectively. The sum of the concentrations of S_1 and S_0 remains constant during the reactions. Assuming C is the sum of S_1 and

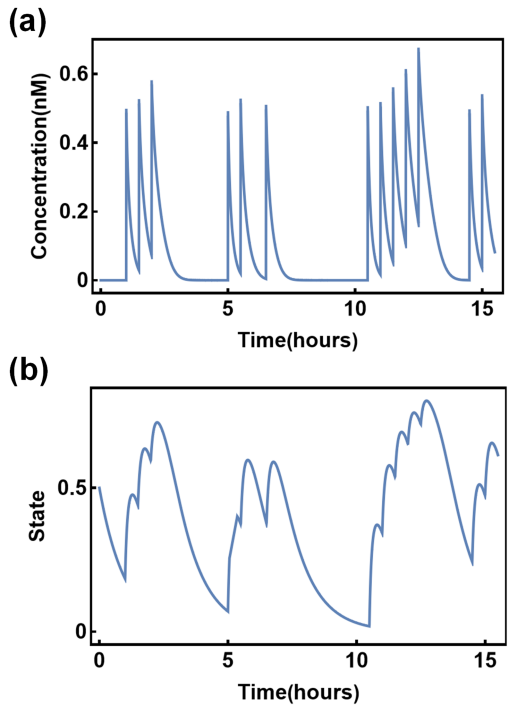

Fig. 3. Behavior of a molecular memristor. **a** The same quantity of the input molecule is added into the system with different time intervals in between. **b** Corresponding memristor state. If multiple short-interval pulses are applied, the memristor state will increase. And the memristor state will return to its initial resting state (0) if there is a long enough time without any input.

S_0's initial concentrations, then the bipolar state can be represented by $S = ([S_1] - [S_0])/([S_1] + [S_0]) = (2[S_1] - C)/C$. This implies that the concentration of S_1 ($[S_1]$) can replace the bipolar state (S) as the molecular memristors' state since S is just a linear transformation of $[S_1]$. In this article, we introduce a different definition of state for the molecular memristor with an analog state as shown in Fig. 2 where the concentration of the output molecule $[S_1]$ determines the *state* of the memristor. Then $S = [S_1]$ where $[S_1]$ is the concentration of S_1. Notice that the state cannot be negative. When the system has some input molecules (I), S_0 will be converted to the state molecule S_1. Without input, the state molecule S_1 will be converted to S_0. The behavior of the molecular reservoir is determined by the two molecular reactions' rate constants (k_1 and k_2).

To show the response of the proposed molecular memristor to a pulse stream input signal, we add the same quantity of the input molecule into the system with different time intervals in between as shown in Fig. 3(a) and monitor the corresponding memristor state which is the concentration of the state molecule S_1 as shown in Fig. 3(b). The initial concentrations of S_1 and S_0 are both set to 0.5 nM and the two molecular reaction rates ($k_1 : k_2$) are set at a ratio of $10:1$. Two properties can be observed: (1) if multiple short-interval pulses are applied, the memristor state will increase, (2) the memristor state will return to its initial resting state (0) if there is a long enough time without any input. These properties demonstrate the molecular memristors' short-term memory effects, where the state is dependent on both the history of external input and an internal state variable [1], [3], [47], [48].

In Fig. 4, we illustrate the responses of the memristor to two different inputs: a sinusoidal input and a triangular input. The initial concentrations of S_1 and S_0 are set to 0 and 1, respectively. The two molecular reaction rates ($k_1 : k_2$) are set at a ratio of $1:1$. The sinusoidal input is given by:

$$I(t) = \frac{1}{2}\sin\left(\frac{1}{75}\pi t\right) + \frac{1}{2}. \tag{1}$$

In Fig. 4(a), the blue and red lines represent the sinusoidal input and the corresponding state, respectively. With a 0.8 h time step between successive samples, we sampled 300 points from two phases of the sine wave and input them to the memristor. The *input-state* curves of the molecular memristor with the sinusoidal pulse stream input form a hysteresis loop as shown in Fig. 4(b). Fig. 4(c) shows the hysteresis loops of the molecular memristors with the same sinusoidal inputs for various rate ratios ranging from $20:1$ to $1:20$ in addition to the original rate ratio $1:1$ between the two molecular reactions in Fig. 2.

Figs. 4(d)–(f) show the responses of the memristor to the triangular input. In Fig. 4(d), the blue and red lines represent the triangular input and the corresponding state, respectively. The *input-state* curves of the molecular memristor with the triangular pulse stream input form a hysteresis loop as shown in Fig. 4(e). Fig. 4(f) shows the hysteresis loops of the molecular memristors with the same triangular inputs for various rate ratios ranging from $20:1$ to $1:20$.

Theoretically, we can prove a memristor is realized by the continuous molecular concentration kinetics of reactions listed in Fig. 2. For these reactions the ordinary differential equations (ODEs) are given by:

$$\begin{aligned}
\frac{d[S_0]}{dt} &= k_2[S_1] - k_1[I][S_0] \\
\frac{d[S_1]}{dt} &= k_1[I][S_0] - k_2[S_1] = -\frac{d[S_0]}{dt} \\
\frac{d[I]}{dt} &= -k_1[I][S_0]
\end{aligned} \tag{2}$$

where $[S_0]$, $[S_1]$ and $[I]$ represent the time-varying concentrations of molecules S_0, S_1 and I, respectively. Parameters k_1 and k_2 represent the rate constant of the two molecular reactions. Notice that since $\frac{d[S_0]}{dt} + \frac{d[S_1]}{dt} = 0$ at any time, then the sum of the concentrations of S_0 and S_1 ($[S_0] + [S_1]$) is constant. Assume C is the sum of S_1 and S_0's initial concentrations.

Then the memristor state (S) can be expressed as:

$$S(t) = [S_1] = C - [S_0],$$

where $S(t)$ represents the time-varying value of the state. Then we can compute $\frac{S(t)}{d[I]/dt}$ by substituting the last equation of (2) as follows:

$$\begin{aligned}
\frac{S(t)}{d[I]/dt} &= \frac{C - [S_0]}{-k_1[I][S_0]} \\
&= \frac{1}{k_1[I]} + \frac{C}{-k_1[I][S_0]}
\end{aligned}$$

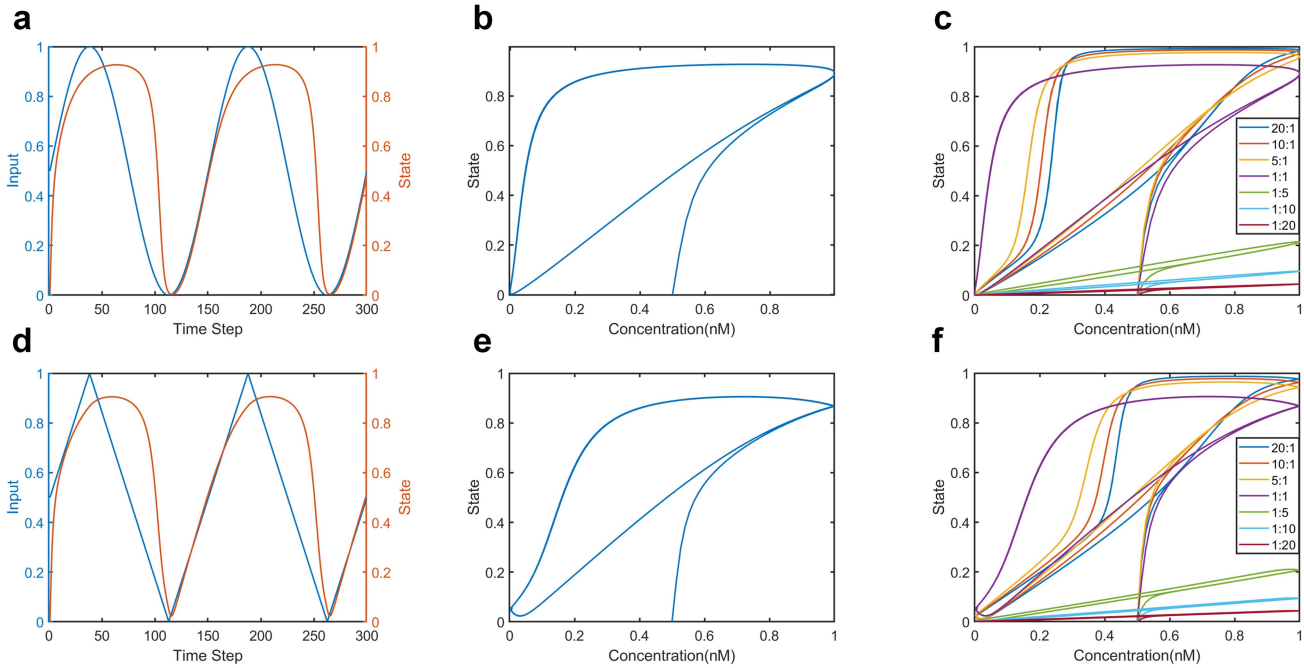

Fig. 4. Behavior of a molecular memristor and its input-output curves. (a) The blue and red lines represent the sinusoidal input and the corresponding state, respectively. (b) The input-state curves of the molecular memristor with the sinusoidal pulse stream input form a hysteresis loop. (c) The hysteresis loops of the molecular memristors with the same sinusoidal inputs for various rate ratios ranging from 20 : 1 to 1 : 20. (d) The blue and red lines represent the triangular input and the corresponding state, respectively. (e) The input-state curves of the molecular memristor with the triangular pulse stream input form a hysteresis loop. (f) The hysteresis loops of the molecular memristors with the same triangular inputs for various rate ratios ranging from 20 : 1 to 1 : 20.

$$= \frac{1}{k_1[I]} + \frac{C}{dI(t)/dt}$$
$$\Rightarrow \frac{S(t)-C}{d[I]/dt} = \frac{1}{k_1[I]}.$$

By definition, the voltage across the charge-controlled memristor can be computed as [1]:

$$v(t) = M(q(t))i(t),$$

where

$$M(q(t)) = \frac{d\Phi/dt}{dq/dt} = \frac{v(t)}{i(t)},$$

where v and i represent voltage and current, respectively. The magnetic flux linkage, Φ, can be regarded as the integral of voltage over time. The amount of electric charge that has flowed is represented by q. Let i be the derivative of q. The memristance of a memristor is characterized by the term M.

The state with an offset $(S - C)$ and the number of input molecules (I) of the proposed molecular memristor are analogous to the voltage (v) and the amount of electric charge (q) of the standard memristor, respectively. Then the memristance of the proposed molecular memristor can be rewritten as:

$$M = \frac{v(t)}{dq(t)/dt} = \frac{S(t)-C}{d[I]/dt}$$
$$= \frac{1}{k_1[I]} = M([I]).$$

Notice that the proposed molecular memristor's memristance is a single-valued function of the input which meets Chua's definition of a memristor [1].

The implementation of molecular memristors in cascade has not been described before. Consider the implementation of two molecular memristors in cascade as shown in Fig. 5(a). The states of the two molecular memristors are represented by the concentrations of the two state molecules (S_{11} and S_{21}), respectively. Notice that the input molecule of the second molecular memristor (I_2) can also be internally generated by the first molecular reaction of the first molecular memristor as shown in Fig. 5(a). To demonstrate the behavior of the two molecular memristors in cascade, we applied the same sinusoidal pulse stream input as shown in (1) to the first molecular memristor. A specified amount of the input molecule (I_1) that is proportional to the external input will be added to the system at each time step. In Fig. 5(b), the blue line represents the sinusoidal pulse stream input. The states of the two molecular memristors are represented by the red solid line and the red dashed line, respectively.

Then we can analyze the mass-action kinetics model of the molecular reactions shown in Fig. 5(a). For the four molecular reactions, the ODEs are given by:

$$\frac{d[S_{10}]}{dt} = k_{12}[S_{11}] - k_{11}[I_1][S_{10}]$$
$$\frac{d[S_{11}]}{dt} = k_{11}[I_1][S_{10}] - k_{12}[S_{11}] = -\frac{d[S_{10}]}{dt}$$

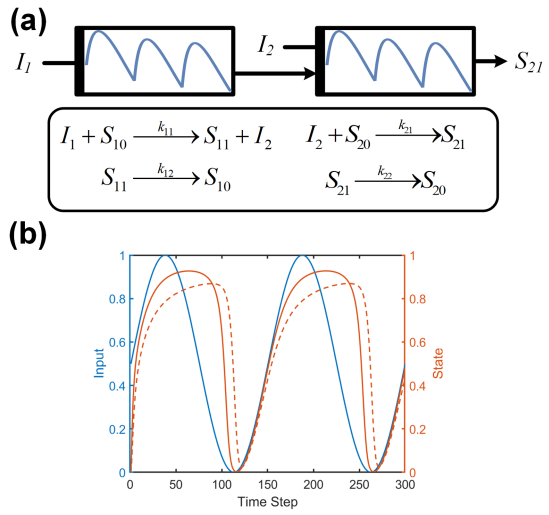

Fig. 5. Implementation of two molecular memristors in cascade with their responses. (a) Two molecular memristors in cascade can be achieved by four molecular reactions. The states of the two molecular memristors are represented by the concentrations of the two state molecules (S_{11} and S_{21}), respectively. (b) Corresponding states of the two molecular memristors with the sinusoidal pulse stream input. A specified amount of the input molecule (I_1) that is proportional to the external input will be added to the system at each time step. The states of the two molecular memristors are represented by the red solid line and the red dashed line, respectively.

$$\begin{aligned}
\frac{d[I_1]}{dt} &= -k_{11}[I_1][S_{10}] \\
\frac{d[S_{20}]}{dt} &= k_{22}[S_{21}] - k_{21}[I_2][S_{20}] \\
\frac{d[S_{21}]}{dt} &= k_{21}[I_2][S_{20}] - k_{22}[S_{21}] = -\frac{d[S_{20}]}{dt} \\
\frac{d[I_2]}{dt} &= k_{11}[I_1][S_{10}] - k_{21}[I_2][S_{20}] \\
&= -\frac{d[I_1]}{dt} - k_{21}[I_2][S_{20}] \qquad (3)
\end{aligned}$$

where $[S_{10}]$, $[S_{11}]$, $[I_1]$, $[S_{20}]$, $[S_{21}]$ and $[I_2]$ represent the time-varying concentrations of molecules S_{10}, S_{11}, I_1, S_{20}, S_{21} and I_2, respectively. The rate constants of the four molecular reactions are k_{11}, k_{12}, k_{21} and k_{22}, respectively. As described in the first three equations of (3), the ODEs of the first molecular memristor are identical to those of a single molecular memristor as shown in (2). Thus, the first molecular memristor meets the definition of a memristor theoretically.

Assume C_2 is the sum of S_{20} and S_{21}'s initial concentrations. Then the time-varying state value of the second molecular memristor ($S_2(t)$) is defined as:

$$S_2(t) = [S_{21}] = C_2 - [S_{20}].$$

From the last equation of (3), we can compute $\frac{S_2(t)}{d([I_1]+[I_2])/dt}$ as follows:

$$\begin{aligned}
\frac{S_2(t)}{d([I_1]+[I_2])/dt} &= \frac{C_2 - [S_20]}{-k_{21}[I_2][S_20]} \\
&= \frac{1}{k_{21}[I_2]} + \frac{C_2}{d([I_1]+[I_2])/dt} \\
\Rightarrow \frac{S_2(t) - C_2}{d([I_1]+[I_2])/dt} &= \frac{1}{k_{21}[I_2]}.
\end{aligned}$$

The state ($S_2 - C_2$) of the second molecular memristor with an offset is equivalent to the voltage (v) of the standard memristor; the sum of the concentrations of the two input molecules (I_1 and I_2) can be considered as the amount of electric charge (q) of the standard memristor. Then the memristance of the second molecular memristor can be rewritten as:

$$\begin{aligned}
M &= \frac{v(t)}{dq(t)/dt} = \frac{S_2(t) - C_2}{d([I_1]+[I_2])/dt} \\
&= \frac{1}{k_{21}[I_2]} = M([I_2]).
\end{aligned}$$

The memristance of the second molecular memristor is also a single-valued function of the input. Thus, the second molecular memristor meets the definition of a memristor theoretically. This analysis can be extended to the molecular memristor with multiple internal inputs from different molecular memristors by considering that the sum of their input molecules is equivalent to the amount of electric charge (q) that has flowed.

B. Molecular Readout Layer

The reservoir state is mapped to the desired output via a readout layer. Inspired by some previous papers [37], [40], [41], multiple analog multiplications can be combined to create a molecular readout layer. The reservoir state ($\mathbf{s}$) and the readout layer (W_{out}) can be expressed as:

$$\mathbf{s} = \begin{bmatrix} S_1 \\ S_2 \\ : \\ S_N \end{bmatrix}, \quad W_{out} = \begin{bmatrix} W_{11} & W_{12} & .. & W_{1\,N} \\ W_{21} & W_{22} & .. & W_{2\,N} \\ & : & & \\ W_{M1} & W_{M2} & .. & W_{MN} \end{bmatrix}$$

Then the output vector is defined as:

$$\mathbf{y} = \begin{bmatrix} Y_1 \\ Y_2 \\ : \\ Y_M \end{bmatrix} = W_{out} \cdot \mathbf{s}.$$

where N represents the dimension of the reservoir state and M represents the number of output classes. All the elements in the reservoir state ($\mathbf{s}$) are nonnegative since they are all determined by the concentration of the corresponding output molecule. However, all the elements in the trained readout layer (W_{out}) cannot be guaranteed to be non-negative. So each element of the readout layer, W_{ij}, can be represented by the difference between two molecules' concentrations, ${W_{ij}}^+$ and ${W_{ij}}^-$, where ${W_{ij}}^+$ and ${W_{ij}}^-$ are defined as:

$${W_{ij}}^+ = \begin{cases} W_{ij} & \text{if } W_{ij} \geq 0 \\ 0 & \text{if } W_{ij} < 0 \end{cases}$$

$$W_{ij}{}^{-} = \begin{cases} 0 & \text{if } W_{ij} \geq 0 \\ |W_{ij}| & \text{if } W_{ij} < 0 \end{cases}$$

Then the readout layer can be implemented by $2M(N+1)$ molecular reactions as follows:

$$\left.\begin{array}{l} S_j + {W_{1j}}^{\pm} \longrightarrow S_j + {W_{1j}}^{\pm} + {Y_1}^{\pm} \\ S_j + {W_{2j}}^{\pm} \longrightarrow S_j + {W_{2j}}^{\pm} + {Y_2}^{\pm} \\ \cdots \\ S_j + {W_{Mj}}^{\pm} \longrightarrow S_j + {W_{Mj}}^{\pm} + {Y_M}^{\pm} \end{array}\right\} \text{for } j = 1, 2, \ldots, N \tag{4}$$

and

$$\left.\begin{array}{l} {Y_i}^{+} \longrightarrow \varnothing \\ {Y_i}^{-} \longrightarrow \varnothing \end{array}\right\} \text{for } i = 1, 2, \ldots, M \tag{5}$$

Each reaction in reactions (4) is a concise representation of two reactions:

$$S_j + {W_{ij}}^{+} \longrightarrow S_j + {W_{ij}}^{+} + {Y_i}^{+}$$
$$S_j + {W_{ij}}^{-} \longrightarrow S_j + {W_{ij}}^{-} + {Y_i}^{-}$$

Reactions (4) and (5) should all have the same reaction rate. All positive rates are accepted. From reactions (4), it may seem that molecules ${Y_i}^{\pm}$ are produced out of nothing since the two reactants are also the products. In fact, the *fuel* molecules that drive the reactions are ignored. It allows us to focus on the computationally relevant molecules [49]. Similar to the representation of the readout layer, each element in the output vector, Y_i, is also represented by the difference between two molecules' concentrations, ${Y_i}^{+}$ and ${Y_i}^{-}$, $Y_i = {Y_i}^{+} - {Y_i}^{-}$. By analyzing the mass-action kinetics model, the ODEs are given by:

$$\begin{aligned} \frac{d{Y_i}^{+}}{dt} &= \sum_{j=1}^{N} {W_{ij}}^{+} S_j - {Y_i}^{+} \\ \frac{d{Y_i}^{-}}{dt} &= \sum_{j=1}^{N} {W_{ij}}^{-} S_j - {Y_i}^{-} \\ \frac{dY_i}{dt} &= \frac{d{Y_i}^{+}}{dt} - \frac{d{Y_i}^{-}}{dt} \\ &= \sum_{j=1}^{N} {W_{ij}}^{+} S_j - \sum_{j=1}^{N} {W_{ij}}^{-} S_j - ({Y_i}^{+} - {Y_i}^{-}) \\ &= W_i \mathbf{s} - Y_i, \end{aligned}$$

where $W_i = \begin{bmatrix} W_{i1} & W_{i2} & .. & W_{iN} \end{bmatrix}$. In the steady-state, we have $Y_i = \sum_{j=1}^{N} W_{ij} S_j = W_i \mathbf{s}$ by setting dY_i/dt to 0. The output vector is then formed by all the Y_i for $i = 1, 2, \ldots, M$. Therefore, $2M(N+1)$ molecular reactions are required to generate the readout layer that transfers the N-dimensional reservoir state to the M-dimensional output vector. The position of the maximum element in the output vector dictates the final decision in classification problems.

III. Mapping Molecular Computing System to DNA

Formal CRNs can be mapped on to DNA strand displacement reactions [23], [24], [34], [50]. The toehold-mediated strand displacement reaction (SDR) was introduced by Yurke et al. in 2000 [51]. It is a powerful enzyme-free tool for DNA computing. A framework that can simulate molecular reactions with one or two reactants by using double-stranded DNA complexes was proposed by Chen et al. [52]. In this article, all the proposed molecular reactions have one or two reactants. We used the CRNSimulator Mathematica package [50] to simulate the DNA RC systems.

A. DNA Memristor

The molecular memristor shown in Fig. 2 can be further mapped into 5 DSD reactions:

$$\begin{array}{l} (a): \quad I + S_0 \overset{k_1}{\Rightarrow} S_1 \\ (b): \quad S_1 \overset{k_2}{\Rightarrow} S_0 \end{array} \Longrightarrow \begin{array}{ll} (a) & \begin{cases} I + L_1 & \underset{q_{\max}}{\overset{q_1}{\rightleftharpoons}} H_1 + B_1 \\ S_0 + H_1 & \overset{q_{\max}}{\Longrightarrow} O_1 \\ O_1 + T_1 & \overset{q_{\max}}{\Longrightarrow} S_1 \end{cases} \\ (b) & \begin{cases} S_1 + G_2 & \overset{q_2}{\Rightarrow} O_2 \\ O_2 + T_2 & \overset{q_{\max}}{\Longrightarrow} S_0 \end{cases} \end{array} \tag{6}$$

where L_i, B_i, G_i and T_i represent different auxiliary complexes. Here $q_{\max}$ stands for the maximum strand displacement rate constant. Rate constants k_1 and k_2 of the two corresponding molecular reactions can be obtained by choosing proper rate constants q_1 and q_2, respectively. Rate constants q_1 and q_2 are less than $q_{\max}$. The details of mapping molecular reactions into DSD reactions are presented in [50].

Since the initial concentrations of L_i, B_i, G_i and T_i are set to a large enough value, $C_{\max}$, these concentrations can be assumed to be effectively unchanged. Then the 5 DSD reactions can be rewritten as follows [50]:

$$\begin{array}{ll} (a) & \begin{cases} I \underset{q_{\max} C_{\max}}{\overset{q_1 C_{\max}}{\rightleftharpoons}} H_1 \\ S_0 + H_1 \overset{q_{\max}}{\Longrightarrow} S_1 \end{cases} \\ (b) & \begin{cases} S_1 \overset{q_2 C_{\max}}{\Longrightarrow} O_2 \\ O_2 \overset{q_{\max} C_{\max}}{\Longrightarrow} S_0 \end{cases} \end{array} \tag{7}$$

For the 4 simplified DSD reactions, the corresponding ODEs can be written as [34]:

$$\begin{aligned} \frac{d[S_0]}{dt} &= q_{\max} C_{\max} [O_2] - q_{\max} [S_0][H_1] \\ \frac{d[S_1]}{dt} &= q_{\max} [S_0][H_1] - q_2 C_{\max} [S_1] \\ \frac{d[I]}{dt} &= q_{\max} C_{\max} [H_1] - q_1 C_{\max} [I] \\ \frac{d[H_1]}{dt} &= q_1 C_{\max} [I] - q_{\max} C_{\max} [H_1] - q_{\max} [S_0][H_1] \\ &= -\frac{d[I]}{dt} - q_{\max} [S_0][H_1] \end{aligned}$$

$$\frac{d[O_2]}{dt} = q_2 C_{\max}[S_1] - q_{\max} C_{\max}[O_2] \tag{8}$$

Since $\frac{d[S_0]}{dt} + \frac{d[S_1]}{dt} + \frac{d[O_2]}{dt} = 0$ at any time, $([S_0] + [S_1] + [O_2])$ should always be constant. From the last two reactions of (7), the rate at which O_2 is consumed ($q_{\max} C_{\max}$) is much higher than the rate at which it is generated ($q_2 C_{\max}$). The concentration of O_2 can be assumed to be close to zero throughout the duration of the experiment. Then we can conclude that $([S_0] + [S_1])$ remains effectively constant at any time. Further, we can assume that I and H_1 achieve instant equilibrium via the first reaction of (7) with $[I]/[H_1] = q_{\max}/q_1$ [50]. By taking the derivative of both sides, we can get $\frac{d[I]}{dt} / \frac{d[H_1]}{dt} = q_{\max}/q_1$. Then the fourth equation of (8) can be rewritten as [34]:

$$\frac{d[H_1]}{dt} + \frac{d[I]}{dt} = -q_{\max}[S_0][H_1]$$

$$\Rightarrow \left(\frac{q_1}{q_{\max}} + 1\right)\frac{d[I]}{dt} = -q_{\max}[S_0]\frac{q_1}{q_{\max}}[I] = -q_1[I][S_0]$$

$$\xRightarrow{q_1 << q_{\max}} \frac{d[I]}{dt} = -q_1[I][S_0]$$

Given that $[S_1] + [S_0] = C$. Then we can rewrite the state as $S(t) = [S_1] = C - [S_0]$. As shown before, the state with an offset value ($S - C$) is analogous to the voltage (v). Similarly, the amount of input molecules (I) is analogous to the amount of electric charge (q). Therefore, the memristance of the proposed DNA memristor can be computed as:

$$M = \frac{v(t)}{dq(t)/dt} = \frac{S(t) - C}{d[I]/dt}$$

$$= \frac{1}{q_1[I]} = M([I]).$$

Notice that the behavior of the proposed DNA memristor is similar to that of the original memristor, as the memristance of the DNA memristor can be expressed in the same form as the proposed molecular memristor. Therefore, the proposed DNA memristor exhibits analogous behavior to the original memristor.

B. DNA Readout Layer

For the molecular readout layer, each molecular reaction shown in reactions (4) and (5) can be mapped into 3 and 2 DSD reactions, respectively, as follows [40]:

$$\begin{aligned}
&(a): S_y + {W_{xy}}^{\pm} \overset{k}{\Rightarrow} S_y + {W_{xy}}^{\pm} + {Y_x}^{\pm} \\
&(b): {Y_x}^{\pm} \overset{k}{\Rightarrow} \varnothing \\
&\Longrightarrow \\
&(a)\begin{cases} S_y + L_j & \underset{q_{\max}}{\overset{q}{\rightleftharpoons}} H_j + B_j \\ {W_{xy}}^{\pm} + H_j & \xRightarrow{q_{\max}} O_j \\ O_j + T_j & \xRightarrow{q_{\max}} S_y + {W_{xy}}^{\pm} + {Y_x}^{\pm} \end{cases} \\
&(b)\begin{cases} {Y_x}^{\pm} + G_j & \overset{q}{\Rightarrow} O_j \\ O_j + T_j & \xRightarrow{q_{\max}} \varnothing \end{cases}
\end{aligned} \tag{9}$$

where L_j and T_j represent auxiliary complexes, and q is chosen to obtain the desired rate (k) of the corresponding molecular reaction. The readout layer with $2M(N+1)$ molecular reactions that transfers the N-dimensional reservoir state to the M-dimensional output vector can be further mapped into $(6NM + 4M)$ DSD reactions.

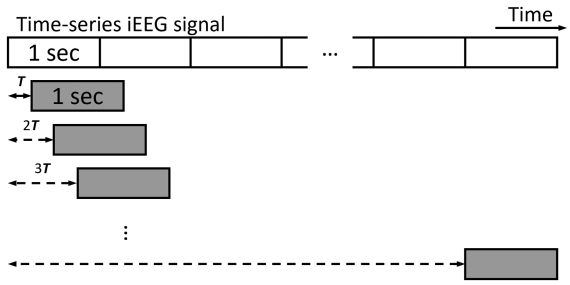

Fig. 6. Create additional segments by sliding a 1-second window along the time axis at every step T where the value of T is determined for each subject to ensure that the number of samples in each class (ictal or interictal) is similar in the training set [54].

TABLE I
UPENN AND MAYO CLINIC'S SEIZURE DETECTION CHALLENGE DATASET INFORMATION

Subj.	# inctal	# interictal	# test	# channel	f_s (Hz)
Dog_1	1771	1669	3181	16	400
Dog_2	1711	1148	2997	16	400
Dog_3	4791	4760	4450	16	400
Dog_4	2561	2790	3013	16	400
Patient_1	691	825	2050	68	400
Patient_2	1501	2990	3894	16	400
Patient_3	3261	3566	1281	55	400
Patient_4	191	190	543	72	400
Patient_5	1073	2610	2986	64	400
Patient_6	2241	2772	2997	30	400
Patient_7	2811	3239	3601	36	400
Patient_8	1433	1710	1922	16	400

IV. SEIZURE DETECTION USING DNA RESERVOIR

In this section, the memristor-based RC system was applied to analyze the iEEG dataset from the UPenn and Mayo Clinic's Seizure Detection Challenge [38], which was organized by Kaggle [53]. The dataset contains iEEG from 4 dogs and 8 patients with epileptic seizures. iEEG signals were recorded from 16 electrodes at a frequency of 400 HZ for dogs and varying numbers of electrodes (ranging from 16 to 72) at two different frequencies (500 Hz or 5000 Hz) for patients. This dataset comprises 48 seizures and 6.5 hours of interictal data that were pre-organized into 1-second clips where each clip represents one second of iEEG data.

The presence of imbalanced class instances poses a challenge in various classification tasks. In the UPenn and Mayo Clinic's dataset, the interictal to ictal ratio per subject averages at $10:1$. To address this imbalance, we adopt an overlapping technique during the training phase to generate additional segments [54]. We slide a 1-second window along the time axis at each step, T, over the ictal time-series EEG signals (refer to Fig. 6) where T is selected for each subject to ensure that the number of samples per class (ictal or interictal) in the training set is approximately the same. For the sake of simplicity, we also resampled all iEEG signals recorded from patients to 400 Hz. Table I presents the

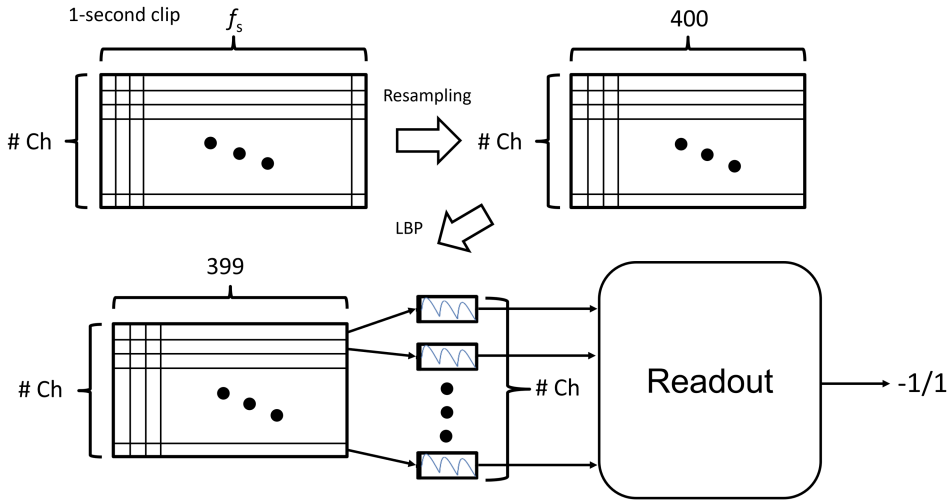

Fig. 7. Process flow of seizure detection task using molecular memristor-based RC system. Each original 1-second clip can be viewed as a matrix, with a number of rows equal to the number of channels and a number of columns equal to the sampling frequencies. Once the resampling process is applied to all clips belonging to the patients, all the matrices have the same size with a number of rows equivalent to the number of channels and a number of columns equal to 400. Then the application of the LBP encoding method resulted in matrices with a size equal to the number of channels multiplied by 399. The information of each channel is fed into one molecular memristor externally. The classification result is produced by the trained readout layer once it receives the state of the reservoir.

number of ictal and interictal 1-second clips after balancing, the number of test 1-second clips, the number of channels, and the sampling frequency of the dataset after resampling.

Then we applied the local binary pattern (LBP) encoding method proposed by [55], [56] to the preprocessed iEEG dataset. The encoding process involved converting iEEG signals into LBP codes. These codes reflect time-domain information and can distinguish between ictal and interictal clips. The computation of LBP codes involves converting consecutive iEEG signal samples into a bit stream, where the assigned LBP code depends on the sign of the temporal difference between adjacent samples. A positive difference corresponds to a code of 1, while a negative difference corresponds to a code of 0. To illustrate, let's consider the encoding of a raw data time series consisting of 7 points ($[2.0, 1.5, 1.2, 2.3, 2.2, 3.2, 3.5]$) using the LBP method. The resulting LBP code has a length of 6 and is represented as $[0, 0, 0, 1, 0, 1]$. Notice that the length of the LBP code is consistently one bit shorter than that of the raw data.

As shown in Fig. 7, each original 1-second clip can be viewed as a matrix, with a number of rows equal to the number of channels and a number of columns equal to the sampling frequencies. Once the resampling process is applied to all clips belonging to the patients, all the matrices have the same size with a number of rows equivalent to the number of channels and a number of columns equal to 400. Then the application of the LBP encoding method resulted in matrices with a size equal to the number of channels multiplied by 399. Subsequently, each channel undergoes encoding as a pulse stream input of 399 time steps, where the 399 binary LBP codes are fed externally into a single molecular memristor. At each time step, a corresponding input molecule (I) unit is added to the system if a pulse is present in the input signal. Thus this reservoir necessitates one molecular memristor for each channel. Each memristor was implemented by 2 molecular reactions or 5 DSD reactions.

We utilized a supervised learning algorithm, namely logistic regression, to train the readout layer in software. For the RC system shown in Fig. 7, the reservoir state, $\mathbf{x}$, is a vector containing $8C$ elements (8 memristors' states over time per memristor and for C memristor in total) where C stands for the number of channels used. Then the state vector was fed into the readout layer. The probability of the molecular reservoir state belonging to the different classes can be computed as follows [57]:

$$h_W(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{W}\cdot\mathbf{x}}},$$

where $\mathbf{W}$ represents the weight matrix which is the same as W^{out} in Fig. 1. And the cost function is defined as:

$$J(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n}[-y^i \log(h_W(\mathbf{x}^i)) - (1-y^i)\log(1-h_W(\mathbf{x}^i))],$$

where n represents the number of training samples and y^i represents the target output for the corresponding state vector $\mathbf{x}^i$. We trained the weight matrix to minimize the cost function using the gradient descent as follows:

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}_j} = \frac{1}{n}\sum_{i=1}^{n}(h_W(\mathbf{x}^i) - y^i)\mathbf{x}_j^i.$$

The training process was computed by using a function called *fmincg()* [58] in MATLAB.

Following this, the trained readout layer can be realized via the molecular reactions mentioned earlier, which can be subsequently mapped into DSD reactions. For each 1-second clip, the output of the readout layer yields a 2-dimensional vector, with one dimension corresponding to the ictal class and the other to the interictal class. The final classification decision is made based on the dimension with the maximum value. To train the readout layer, we employed preprocessed 1-second

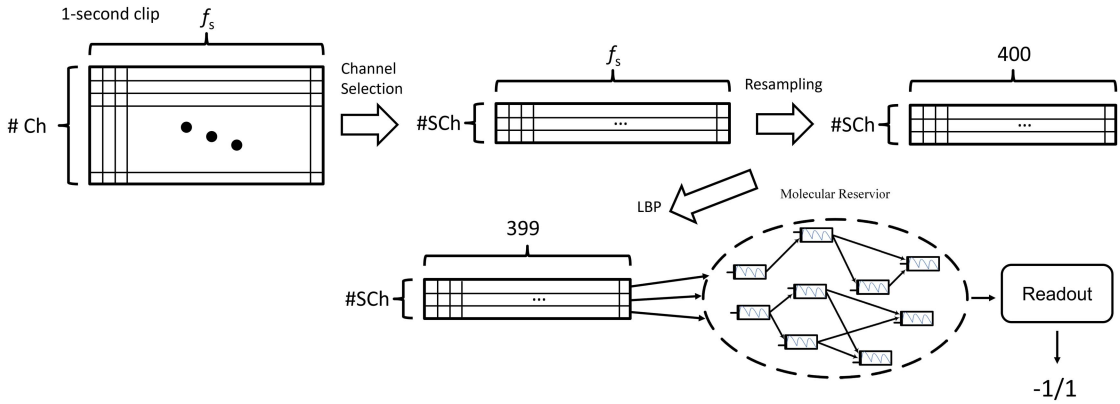

Fig. 8. Process flow of seizure detection task using molecular memristor-based RC system with selected channels. After channel selection, each original 1-second clip matrix of size $C \times f_s$ was reduced to a smaller size of $SC \times f_s$, where C and SC represent the total number of channels and the number of selected channels, respectively. Subsequently, the remaining processes, including resampling and LBP encoding, were identical to those employed before. The information of each selected channel is fed into one molecular memristor externally. The internal directed connections between memristors were randomly generated with probability 0.25. The classification result is produced by the trained readout layer once it receives the state of the reservoir.

TABLE II
SEIZURE DETECTION TEST RESULTS WITH ALL CHANNELS

Subj.	# Memristor	Accuracy (%)	Sensitivity (%)	Specificity (%)	**AUC**
Dog_1	16	85.29	91.19	84.98	**0.95**
Dog_2	16	61.96	69.74	61.55	**0.72**
Dog_3	16	85.60	83.25	85.83	**0.91**
Dog_4	16	58.61	82.40	57.58	**0.80**
Pat_1	68	78.63	28.22	82.99	**0.53**
Pat_2	16	92.55	62.01	94.46	**0.88**
Pat_3	55	55.35	64.84	54.29	**0.66**
Pat_4	72	71.82	20.00	77.08	**0.40**
Pat_5	64	84.19	70.24	85.02	**0.84**
Pat_6	30	96.43	57.73	99.50	**0.93**
Pat_7	36	95.11	87.22	95.99	**0.97**
Pat_8	16	95.01	82.78	96.27	**0.95**
Average		80.05	66.64	81.30	**0.80**

clips from both the ictal and interictal classes for each subject. Subsequently, we utilized preprocessed 1-second clips from a testing dataset to evaluate the classification accuracies of the RC systems.

Throughout the entire simulation duration of 399 time steps, each memristor's state was measured at eight distinct time steps, specifically at the 50th, 100th, 150th, 200th, 250th, 300th, 350th, and 399th time steps. Consequently, the input dimension of the readout layer is $8C$, where C denotes the number of channels. For seizure detection classification, we employed an $8C \times 2$ molecular readout layer. Each molecular memristor can be realized via five DSD reactions, resulting in a total of $5C$ DSD reactions. The molecular readout layer, consisting of $(32C + 4)$ molecular reactions, can be mapped into $(96C + 8)$ DSD reactions. Altogether, the DNA RC system requires $(101C + 8)$ DSD reactions.

As epilepsy is characterized by seizure durations of typically ten seconds to two minutes, seizure detection essentially involves an imbalanced data classification issue. As a result, accuracy may not sufficiently evaluate the detection performance for highly imbalanced datasets. The receiver operating characteristic (ROC) curve is a plot of sensitivity against (1-specificity), and the area under this curve (AUC) can be utilized to evaluate the overall classification performance. AUC is a threshold-free metric. In this article, accuracy, sensitivity, specificity, and AUC are employed to measure classification performance. Table II shows the seizure detection test results for each subject. The DNA memristor-based RC system can achieve an overall detection performance with a test accuracy of 80.05%, test sensitivity of 66.64%, test specificity of 81.30%, and test AUC of 0.80. The RC system in this experiment requires the same number of memristors as the number of channels for each subject, as each memristor processes the information from one channel.

In the previous experiment, it is important to note that there was no interconnection between the memristors, and the data from all channels were utilized. For this experiment, we constructed a molecular memristor-based RC system comprising 16 fixed memristors, utilizing information from the selected channels. As illustrated in Fig. 8, after channel selection, each original 1-second clip matrix of size $C \times f_s$ was reduced to a smaller size of $SC \times f_s$, where C and SC represent the total number of channels and the number of selected channels, respectively. Subsequently, the remaining processes, including resampling and LBP encoding, were identical to those employed in the prior experiment. The information of each selected channel is fed into one molecular memristor externally. The internal directed connections between memristors were randomly generated with probability 0.25. These connections can be represented by a directed graph as shown in Fig. 9, where black circles represent memristors and an arrow from memristor x to memristor y means that the first molecular reaction of memristor x as shown in Fig. 2 will generate the input molecule I_y of the memristor y as well as its state molecule S_x. For example, there are two arrows from memristor 1 to memristor 3 and memristor 9 in Fig. 9. The molecular implementation of memristor 1 can be achieved as follows:

$$I_1 + S_{10} \xRightarrow{k_{11}} S_{11} + I_3 + I_9$$

$$S_{11} \xRightarrow{k_{12}} S_{10}$$

The RC system employed in this experiment consisted of 16 molecular memristors, each state of which was sampled at 8

TABLE III
SEIZURE DETECTION TEST RESULTS WITH SELECTED CHANNELS. TWO SETS OF SELECTED CHANNELS FROM [59] AND [60] WERE USED

	Selected Channels from [60]					Selected Channels from [59]				
Subj.	# Memristor	Accuracy (%)	Sensitivity (%)	Specificity (%)	**AUC**	# Memristor	Accuracy (%)	Sensitivity (%)	Specificity (%)	**AUC**
Dog_1	16	91.26	87.42	91.46	**0.95**	16	84.44	92.45	84.02	**0.96**
Dog_2	16	54.02	86.84	52.26	**0.81**	16	24.08	90.13	20.53	**0.69**
Dog_3	16	82.13	83.50	82.00	**0.89**	16	82.49	85.25	82.22	**0.91**
Dog_4	16	56.29	88.80	54.88	**0.68**	16	46.96	74.40	45.78	**0.56**
Pat_1	16	63.46	42.33	65.29	**0.56**	16	70.54	56.44	71.75	**0.67**
Pat_2	16	86.08	37.55	89.11	**0.72**	16	91.37	53.71	93.72	**0.83**
Pat_3	16	15.38	100.00	5.98	**0.87**	16	32.40	93.75	25.59	**0.81**
Pat_4	16	67.22	28.00	71.20	**0.44**	16	57.27	34.00	59.63	**0.43**
Pat_5	16	89.85	52.98	92.05	**0.79**	16	80.25	38.09	82.75	**0.61**
Pat_6	16	94.19	63.64	96.61	**0.92**	16	89.09	48.18	92.33	**0.78**
Pat_7	16	85.31	65.00	87.57	**0.87**	16	86.56	76.94	87.62	**0.91**
Pat_8	16	88.24	32.22	94.03	**0.79**	16	86.32	38.33	91.27	**0.80**
Average		72.79	64.02	73.54	**0.77**		69.31	65.14	69.77	**0.75**

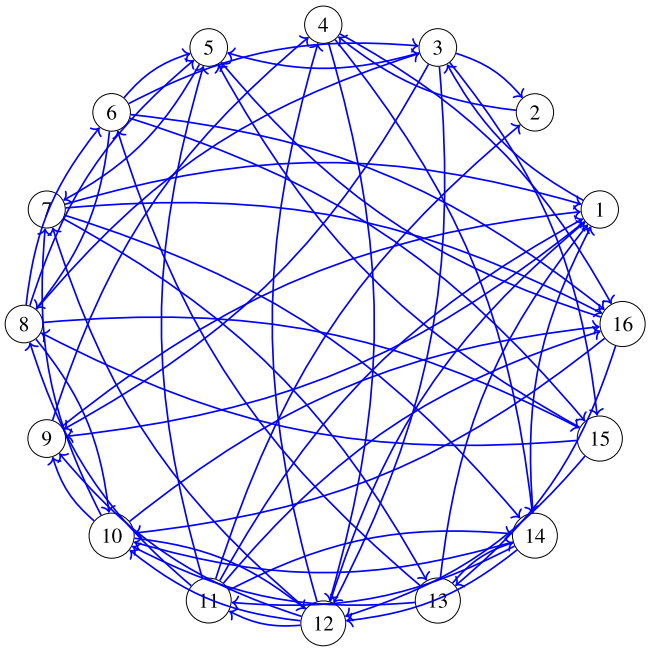

Fig. 9. Randomly generated internal directed connections between 16 memristors. 58 directed connections were generated. Black circles represent memristors and an arrow from memristor x to memristor y means that the first molecular reaction of memristor x as shown in Fig. 2 will generate the input molecule I_y of the memristor y as well as its state molecule S_x.

distinct time steps, as previously described. A molecular readout layer of size 128×2 was utilized. The 16 memristors were realized through 32 molecular reactions or 80 DSD reactions, while the molecular readout layer required 516 molecular reactions, equivalent to 1544 DSD reactions. We utilized two sets of selected channels derived from [59] and [60]. Table III shows the corresponding seizure detection test performance. Comparing Tables II and III, using selected channels can achieve similar average AUC performance as that of using all channels. We note that the DNA memristors cannot successfully detect seizures for all subjects. For example, with all channels, the AUC values for patients Pat_1, Pat_3, and Pat_4 are below 0.7 (see Table II). Using selected channels, the AUC values for dog Dog_4 and patients Pat_1, and Pat_4 are below 0.7 (see Table III).

V. SOLVING A SECOND-ORDER NONLINEAR DYNAMICAL TASK USING DNA RESERVOIR

The section employed a memristor-based RC system to solve a second-order dynamical task that involves input in the time domain [7], [34]. Nonlinear dynamical systems have numerous applications in various engineering fields [61]. Given their close affinity with electrical systems, second-order nonlinear dynamical systems have been the subject of extensive scrutiny.

Given an input $I(t)$ and past outputs, we generated the current output $y(t)$ based on a nonlinear input-output equation [62]:

$$y(t) = 0.4y(t-1) + 0.4y(t-1)y(t-2) + 0.6I^3(t) + 0.1.$$

This experiment, as described in [7], involves predicting the current output $y(t)$ based on the current input $I(t)$ and the product of two past outputs, $y(t-1)$ and $y(t-2)$. This introduces a two-step time-lag to the system, making the task difficult for conventional networks to solve. However, it is well-suited for RC systems with memory effects. The DNA memristor-based RC system was trained to learn the implicit and hidden nonlinear relationship between the input and output.

We generated two input signals for training and testing the DNA memristor-based RC system, which are illustrated in Fig. 10(a) and (b), respectively. Each input signal consisted of 300 time steps with the value of each time step being sampled from a continuous uniform distribution ranging from 0 to 0.5. As shown in Fig. 11, the RC system's reservoir, consisting of 3 molecular memristors in cascade, was implemented using 6 molecular reactions. These reactions could be further mapped to 15 DSD reactions. External analog input was encoded into the concentration of the input molecule I_1. The input molecules I_2 and I_3 are internally generated by the previous memristor as shown in Fig. 11. The state vector of the reservoir was computed by cascading the concentrations of the three state molecules (S_{11}, S_{21}, and S_{31}), and the readout layer was a 4×1 feed-forward layer used to convert the three states of the memristors and one bias to a single output $y(t)$. The molecular readout layer, consisting of 10 molecular reactions, could be mapped to 28 DSD reactions. During the training process, the first transient period (50 initial data points) was excluded, and only the last 250 points in the training dataset were used to train the readout function using linear regression.

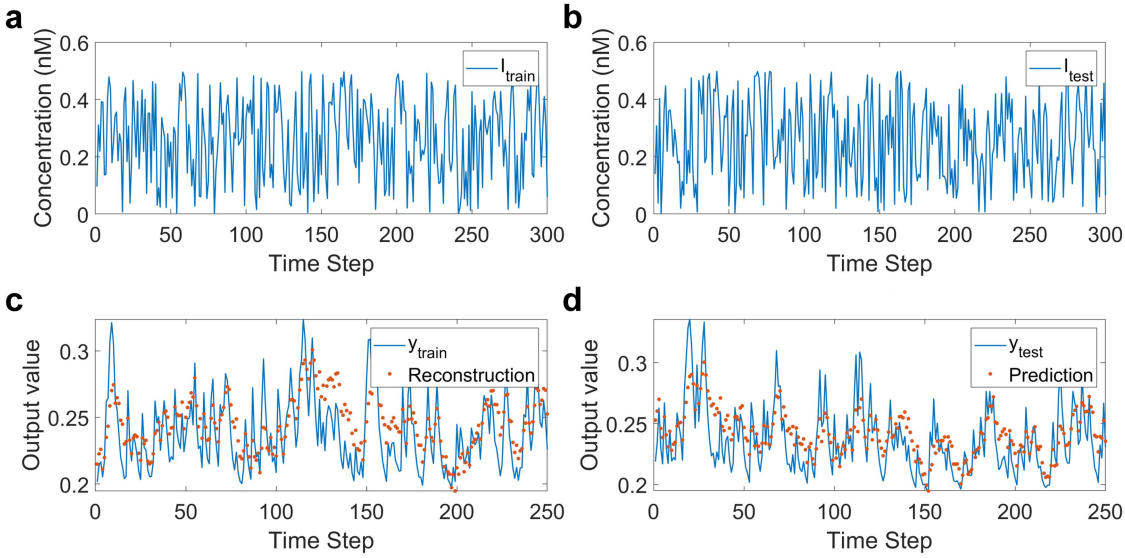

Fig. 10. Time-Series Prediction of Second-Order Nonlinear Dynamical System. **a** Uniform randomly generated training input signal $I_{train}(t)$ **b** Uniform randomly generated testing input signal $I_{test}(t)$ **c** The theoretical output (blue solid line) and the experimentally reconstructed output (red dots) from the DNA memristor reservoir computing system are compared for the last 250 time steps during training. **d** The theoretical output (blue solid line) and the experimentally reconstructed output (red dots) from the DNA memristor reservoir computing system are compared for the last 250 time steps during testing.

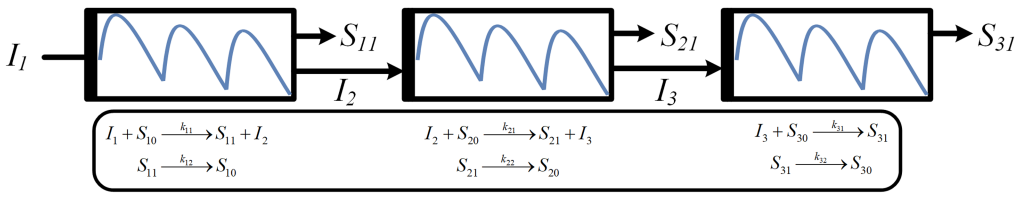

Fig. 11. Three molecular memristors in cascade which are used as the reservoir of the memristor-based RC system. These memristors can be achieved by 6 molecular reactions. External analog input is encoded into the concentration of the input molecule I_1. The state vector of the reservoir is computed by cascading the concentrations of the three state molecules (S_{11}, S_{21} and S_{31}).

Given the reservoir state vector, $\mathbf{x}$, consisting of 3 states of the memristors and one bias, the cost function can be defined as:

$$J(\mathbf{W}) = \frac{1}{2n}\sum_{i=1}^{n}(\mathbf{W}^{\intercal}\mathbf{x}^{(i)} - y^{(i)})^2,$$

where n represents the number of samples, $y^{(i)}$ represents the target output for the corresponding input $x^{(i)}$ and $\mathbf{W}$ represents the weight vector containing 4 readout-layer weights W_j $(j = 1, \cdots 4)$.

We trained the readout layer to minimize the cost function using the gradient descent as follows:

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}_j} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{W}^{\intercal}x^{(i)} - y^{(i)})\mathbf{x}_j^{(i)}.$$

The training process was computed in MATLAB. The normalized mean square error (NMSE) is defined as:

$$NMSE = \frac{\sum_t \sum_i (\tilde{y}_i(k) - y_i(k))^2}{\sum_k \sum_i y_i^2(k)},$$

where $\tilde{y}$ is the predicted signal and y is the target signal. The error is unitless due to the normalization.

The results of the DNA memristor-based RC system are presented in Fig. 10(c), where the blue solid line represents the target output $y(t)$ and the red dots represent the predicted outputs after training. The training process resulted in a normalized mean squared error (NMSE) of 2.7222×10^{-5}. The system was then tested using a different input signal than the one used during training, as shown in Fig. 10(b). The predicted output for this new input signal is shown in Fig. 10(d), and it indicates that the DNA memristor-based RC system can accurately predict the target output for this new input signal. The NMSE for this testing phase is 2.4008×10^{-5}, which is comparable to the NMSE achieved in our previous work with a DNA memristor-based RC system consisting of 10 memristors [34]. Notably, the NMSE of the DNA memristor-based RC system is an order of magnitude better than that of the DNA oscillator-based RC system [40]. Table IV summarizes the comparison of the number of molecular and DSD reactions and the NMSEs between this work and our previous works [34], [40]. Note that the number of DSD reactions in the proposed approach is about 4 and 9

TABLE IV
COMPARISON OF SECOND-ORDER NONLINEAR DYNAMICAL TASK RESULTS USING THE PROPOSED MOLECULAR AND DNA RC SYSTEM WITH EXISTING WORK

Method		Proposed	Liu *et al.* [34]	Liu *et al.* [40]
# Memristors/Oscillators		3	10	10
Size of Readout Layer		4×1	11×1	11×1
# Molecular Reactions	Memristors/Oscillators	6	20	100
	Readout Layer	10	44	24
# DSD Reactions	Memristors/Oscillators	15	50	300
	Readout Layer	28	132	70
Total Molecular Reactions		16	64	124
Total DSD Reactions		43	182	370
Training NMSE		2.7222×10^{-5}	1.3472×10^{-5}	2.3595×10^{-4}
Testing NMSE		2.4008×10^{-5}	1.2019×10^{-5}	3.1455×10^{-4}

times less compared to the prior approaches [34] and [40], respectively.

VI. CONCLUSION

In this article, we propose the molecular and DNA memristor with an analog input and an analog output. The new proposed molecular and DNA memristors have short-term memory effects and are proven to be analogous to the original memristor mathematically. This article also shows that the memristor-based RC system can be synthesized using molecular reactions as well as DNA. Then we demonstrated that even a small RC system with only 16 DNA memristors can be used to solve a seizure detection task. Another RC system with 3 DNA memristors was also used to solve a second-order nonlinear dynamical task and it can successfully predict the target output without knowing the original nonlinear input-output equation. For the second task, the proposed DNA memristor-based RC system can achieve similar performance to that of the original DNA memristor-based RC system from our previous work [34] with much fewer DSD reactions. For the time-series prediction, the number of reactions can be reduced by a factor ranging from 4 to 8. In the Supplementary Information, we show that for classifying digits from the MNIST database, the proposed approach can reduce the number of DSD reactions by a factor of 3 to 4, compared to prior approaches [34] and [40], respectively. Future work will be directed toward analyzing the relationship between the number of memristors used in the DNA memristor-based RC system and its performance.

While practical validation is currently lacking, the proposed method has been shown to be theoretically feasible through simulation. This theoretical advance is significant as it paves the way for future practical use, either *in-vitro* or *in-vivo*, once the necessary technology becomes available. Further optimization of DNA memristor-based RC systems and practical validation of the proposed framework are needed in future work. It should be noted that although the simulation of the chemical kinetics of molecular and DNA systems is highly accurate, the accuracy of experiments in a test tube may differ. DNA computing involves using DNA molecules as information carriers and computational elements. However, due to various factors such as differences in DNA synthesis, purification, and handling, there can be inherent variations between individual DNA molecules or devices. This device-to-device variation can lead to inconsistencies in computation results, affecting the reliability of DNA computing systems. In DNA computing, computation typically involves multiple cycles of biochemical reactions, including DNA hybridization, enzymatic reactions, and DNA amplification. However, these reactions are subject to inherent variability, resulting in cycle-to-cycle variation. This variation can introduce errors and uncertainties, leading to unreliable computation outcomes. It's important to note that while device-to-device and cycle-to-cycle variation pose challenges to the reliability of DNA computing, research efforts are focused on addressing these issues [63]. Therefore, it is important to continue working toward practical validation. Additionally, future work will focus on exploring the use of these memristors for other computations, such as dot products and neural networks.

REFERENCES

[1] L. Chua, "Memristor-the missing circuit element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, Sep. 1971.
[2] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proc. IEEE*, vol. 64, no. 2, pp. 209–223, Feb. 1976.
[3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.
[4] G. W. Burr et al., "Neuromorphic computing using non-volatile memory," *Adv. Phys.: X*, vol. 2, no. 1, pp. 89–124, 2017.
[5] I. Boybat et al., "Neuromorphic computing with multi-memristive synapses," *Nature Commun.*, vol. 9, no. 1, pp. 1–12, 2018.
[6] A. Sebastian, M. L. Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnol.*, vol. 15, no. 7, pp. 529–544, 2020.
[7] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, "Reservoir computing using dynamic memristors for temporal information processing," *Nature Commun.*, vol. 8, no. 1, pp. 1–10, 2017.
[8] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 5187, pp. 1021–1024, 1994.
[9] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, no. 8, pp. 2554–2558, 1982.
[10] E. Mjolsness, D. H. Sharp, and J. Reinitz, "A connectionist model of development," *J. Theor. Biol.*, vol. 152, no. 4, pp. 429–453, 1991.
[11] A. Hjelmfelt, E. D. Weinberger, and J. Ross, "Chemical implementation of neural networks and turing machines," *Proc. Nat. Acad. Sci.*, vol. 88, no. 24, pp. 10983–10987, 1991.
[12] D. Blount, P. Banda, C. Teuscher, and D. Stefanovic, "Feedforward chemical neural network: An in silico chemical system that learns xor," *Artif. Life*, vol. 23, no. 3, pp. 295–317, 2017.

[13] T. Mestl, C. Lemay, and L. Glass, "Chaos in high-dimensional neural and gene networks," *Phys. D: Nonlinear Phenomena*, vol. 98, no. 1, pp. 33–52, 1996.
[14] N. E. Buchler, U. Gerland, and T. Hwa, "On schemes of combinatorial transcription logic," *Proc. Nat. Acad. Sci.*, vol. 100, no. 9, pp. 5136–5141, 2003.
[15] J. Kim, J. Hopfield, and E. Winfree, "Neural network computation by in vitro transcriptional circuits," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2005, pp. 681–688.
[16] H.-W. Lim et al., "In vitro molecular pattern classification via DNA-based weighted-sum operation," *Biosystems*, vol. 100, no. 1, pp. 1–7, 2010.
[17] R. Lopez, R. Wang, and G. Seelig, "A molecular multi-gene classifier for disease diagnostics," *Nature Chem.*, vol. 10, no. 7, pp. 746–754, 2018.
[18] W. Poole et al., "Chemical Boltzmann machines," in *Proc. Int. Conf. DNA-Based Comput.*, 2017, pp. 210–231.
[19] L. Qian, E. Winfree, and J. Bruck, "Neural network computation with DNA strand displacement cascades," *Nature*, vol. 475, no. 7356, pp. 368–372, 2011.
[20] K. M. Cherry and L. Qian, "Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks," *Nature*, vol. 559, no. 7714, pp. 370–376, 2018.
[21] H. Jiang, S. A. Salehi, M. D. Riedel, and K. K. Parhi, "Discrete-time signal processing with DNA," *ACS Synthetic Biol.*, vol. 2, no. 5, pp. 245–254, 2013.
[22] S. A. Salehi, K. K. Parhi, and M. D. Riedel, "Chemical reaction networks for computing polynomials," *ACS Synthetic Biol.*, vol. 6, no. 1, pp. 76–83, 2016.
[23] S. A. Salehi, X. Liu, M. D. Riedel, and K. K. Parhi, "Computing mathematical functions using DNA via fractional coding," *Sci. Rep.*, vol. 8, 2018, Art. no. 8312.
[24] X. Liu and K. K. Parhi, "Molecular and DNA artificial neural networks via fractional coding," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 3, pp. 490–503, Jun. 2020.
[25] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in dna," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012.
[26] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free dna-based data storage," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 5011.
[27] M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna, and E. Charpentier, "A programmable dual-RNA–guided DNA endonuclease in adaptive bacterial immunity," *Science*, vol. 337, no. 6096, pp. 816–821, 2012.
[28] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German Nat. Res. Center Inf. Technol. Tech. Rep.*, vol. 148, no. 34, 2001, Art. no. 13.
[29] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.
[30] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural Netw.*, vol. 20, no. 3, pp. 391–403, 2007.
[31] L. Appeltant et al., "Information processing using a single dynamical node as complex system," *Nature Commun.*, vol. 2, no. 1, pp. 1–6, 2011.
[32] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.
[33] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 659–686.
[34] X. Liu and K. K. Parhi, "DNA memristors and their application to reservoir computing," *ACS Synthetic Biol.*, vol. 11, no. 6, pp. 2202–2213, 2022.
[35] R. Midya et al., "Reservoir computing using diffusive memristors," *Adv. Intell. Syst.*, vol. 1, no. 7, 2019, Art. no. 1900084.
[36] X. Zhu, Q. Wang, and W. D. Lu, "Memristor networks for real-time neural activity analysis," *Nature Commun.*, vol. 11, no. 1, 2020, Art. no. 2439.
[37] S. A. Salehi, H. Jiang, M. D. Riedel, and K. K. Parhi, "Molecular sensing and computing systems," *IEEE Trans. Molecular, Biol. Multi-Scale Commun.*, vol. 1, no. 3, pp. 249–264, Sep. 2015.
[38] "Upenn and mayo clinic's seizure detection challenge," 2014. Accessed: Jan. 27, 2023. [Online]. Available: https://www.kaggle.com/competitions/seizure-detection/data,
[39] Y. LeCun, "The mnist database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[40] X. Liu and K. K. Parhi, "Reservoir computing using DNA oscillators," *ACS Synthetic Biol.*, vol. 11, no. 2, pp. 780–787, 2022.
[41] S. Hayat and T. Hinze, "Toward integration of in vivo molecular computing devices: Successes and challenges," *Hum. Front. Sci. Prog. J.*, vol. 2, no. 5, pp. 239–243, 2008.
[42] C. Du, W. Ma, T. Chang, P. Sheridan, and W. D. Lu, "Biorealistic implementation of synaptic functions with oxide memristors through internal ionic dynamics," *Adv. Funct. Mater.*, vol. 25, no. 27, pp. 4290–4299, 2015.
[43] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J. K. Gimzewski, and M. Aono, "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature Mater.*, vol. 10, no. 8, pp. 591–595, 2011.
[44] T. Chang, S.-H. Jo, and W. Lu, "Short-term memory to long-term memory transition in a nanoscale memristor," *ACS Nano*, vol. 5, no. 9, pp. 7669–7676, 2011.
[45] A. Goudarzi, M. R. Lakin, and D. Stefanovic, "DNA reservoir computing: A novel molecular computing approach," in *Proc. Int. Workshop DNA-Based Comput.*, 2013, pp. 76–89.
[46] W. Yahiro, N. Aubert-Kato, and M. Hagiya, "A reservoir computing approach for molecular computing," in *Proc. Artif. Life Conf.*, 2018, pp. 31–38.
[47] Y. V. Pershin and M. D. Ventra, "Neuromorphic, digital, and quantum computation with memory circuit elements," *Proc. IEEE*, vol. 100, no. 6, pp. 2071–2080, Jun. 2012.
[48] R. Waser and M. Aono, "Nanoionics-based resistive switching memories," in *Nanoscience and Technology: A Collection of Reviews From Nature Journals*. Singapore: World Scientific, 2010, pp. 158–165.
[49] M. Vasić, D. Soloveichik, and S. Khurshid, "CRN++: Molecular programming language," *Natural Comput.*, vol. 19, no. 2, pp. 391–407, 2020.
[50] D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal substrate for chemical kinetics," *Proc. Nat. Acad. Sci.*, vol. 107, no. 12, pp. 5393–5398, 2010.
[51] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann, "A DNA-fuelled molecular machine made of DNA," *Nature*, vol. 406, no. 6796, pp. 605–608, 2000.
[52] Y.-J. Chen et al., "Programmable chemical controllers made from DNA," *Nature Nanotechnol.*, vol. 8, no. 10, pp. 755–762, 2013.
[53] S. N. Baldassano et al., "Crowdsourcing seizure detection: Algorithm development and validation on human implanted device recordings," *Brain*, vol. 140, no. 6, pp. 1680–1691, 2017.
[54] N. D. Truong et al., "Integer convolutional neural network for seizure detection," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 4, pp. 849–857, Dec. 2018.
[55] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. 12th Int. Conf. Pattern Recognit.*, 1994, vol. 1, pp. 582–585.
[56] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996.
[57] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
[58] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 3011–3015, 2010.
[59] L. Ge and K. K. Parhi, "Applicability of hyperdimensional computing to seizure detection," *IEEE Open J. Circuits Syst.*, vol. 3, pp. 59–71, 2022.
[60] Z. Zhang and K. K. Parhi, "Seizure detection using regression tree based feature selection and polynomial SVM classification," in *Proc. IEEE 37th Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2015, pp. 6578–6581.
[61] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[62] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: Unifying the algorithms and accelerating convergence," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 697–709, May 2000.
[63] S.-Y. Shin, I.-H. Lee, D. Kim, and B.-T. Zhang, "Multiobjective evolutionary optimization of dna sequences for reliable dna computing," *IEEE Trans. Evol. Comput.*, vol. 9, no. 2, pp. 143–158, Apr. 2005.

Xingyi Liu (Student Member, IEEE) received the B.S. and M.S.E.E degrees in 2016 and 2018, respectively, from the University of Minnesota-Twin Cities, Minneapolis, MN, USA, where he is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering. His research interests include molecular signal processing, DNA computing and deep neural network using stochastic logic.

Keshab K. Parhi (Fellow, IEEE) received the B.Tech. degree from the Indian Institute of Technolgy, Kharagpur, India, in 1982, the M.S.E.E. degree from the University of Pennsylvania, Philadelphia, PA, USA, in 1984, and the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 1988. He has been with the University of Minnesota, Minneapolis, MN, USA, since 1988, where he is currently the Erwin A. Kelen Chair and a Distinguished McKnight University Professor with the Department of Electrical and Computer Engineering. He has authored or coauthored more than 700 papers, is the inventor or co-inventor of 34 patents, has authored the textbook *VLSI Digital Signal Processing Systems: Design and Implementation* (New York, NY, USA: Wiley, 1999), and coedited the reference book *Digital Signal Processing for Multimedia Systems* (Boca Raton, FL, USA: CRC Press, 1999). His current research interests include the VLSI architecture design of signal processing, and machine learning systems, neuromorphic computing, data-driven neuroscience, biomarkers for neuro-psychiatric disorders, hardware security, and DNA computing. Dr. Parhi has served on the Editorial Boards of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART I AND PART II, the IEEE TRANSACTIONS ON VLSI SYSTEMS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and the *IEEE Signal Processing Magazine*, and was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART I from 2004 to 2005. He currently serves on the Editorial Board of the *Journal of Signal Processing Systems* (Springer). He was the Technical Program Co-Chair of the 1995 IEEE VLSI Signal Processing Workshop and the 1996 Application Specific Systems, Architectures, and Processors conference, and as the General Chair of the 2002 IEEE Workshop on Signal Processing Systems. He was the Distinguished Lecturer of the IEEE Circuits and Systems Society during 1996–1998 and 2019–2020. He was a Board of Governors Elected Member of the IEEE Circuits and Systems Society from 2005 to 2007. He was the recipient of numerous awards including the 2003 IEEE Kiyo Tomiyasu Technical Field Award, the 2017 Mac Van Valkenburg Award, the 2012 Charles A. Desoer Technical Achievement Award and the 1999 Golden Jubilee Medal, from the IEEE Circuits and Systems society, the 2013 Distinguished Alumnus Award from IIT Kharagpur, the 2013 Graduate/Professional Teaching Award from the University of Minnesota, the 2004 F. E. Terman award from the American Society of Engineering Education, and the 2001 IEEE W. R. G. Baker Prize Paper Award. He is a Fellow of ACM, AAAS, NAI, and AIMBE.