scientific reports

Check for updates

OPEN

Simulation platform for pattern recognition based on reservoir computing with memristor networks

Gouhei Tanaka[1,2,3] & Ryosho Nakane[2]

Memristive systems and devices are potentially available for implementing reservoir computing (RC) systems applied to pattern recognition. However, the computational ability of memristive RC systems depends on intertwined factors such as system architectures and physical properties of memristive elements, which complicates identifying the key factor for system performance. Here we develop a simulation platform for RC with memristor device networks, which enables testing different system designs for performance improvement. Numerical simulations show that the memristor-network-based RC systems can yield high computational performance comparable to that of state-of-the-art methods in three time series classification tasks. We demonstrate that the excellent and robust computation under device-to-device variability can be achieved by appropriately setting network structures, nonlinearity of memristors, and pre/post-processing, which increases the potential for reliable computation with unreliable component devices. Our results contribute to an establishment of a design guide for memristive reservoirs toward the realization of energy-efficient machine learning hardware.

Machine learning (ML) has been becoming a vital technology in many industries for promoting artificial intelligence (AI) during the last decade. For further penetration and expansion of practical applications based on ML methods, it is often demanded to enhance their computational efficiency by reducing computational time and resources while maintaining desired performance. Reservoir computing (RC) is one of the ML frameworks that can meet such a demand[1–4]. An RC system is normally composed of an unadaptable dynamic *reservoir* for transforming input time series data (or sequential data) into a high-dimensional feature space and a trainable *readout* for performing a pattern analysis with a simple learning algorithm. The reservoir needs to be well designed for achieving high computational performance in exchange for the simplicity and speediness of the learning process in the readout. For reservoirs constructed with recurrent neural networks[5–7], there are some practical design guides which are mainly helpful in software computation[8,9].

In the exploration of hardware implementation of the adaptability-free reservoirs, various *physical reservoirs* have been developed based on electronics, photonics, spintronics, mechanics, material engineering, robotics, and biology[10]. A unified viewpoint can be obtained by categorizing the reservoir architectures into the network type[11–14], the single nonlinear node plus a delayed-feedback type[15–18], and the excitable continuous medium type[19–21]. However, it is still challenging to derive a design guide for each type of physical reservoir. This is partly because of a difficulty in comprehensively understanding how the computational performance of physical RC systems depends on possible influential factors, such as the system architecture, the physical characteristics of system components, and the signal processing method.

In this study, we tackle the above-mentioned issue by focusing on memristive reservoirs. Memristive systems and devices (or memristors)[22,23] are suitable for constructing physical reservoirs, because their dynamics can show inherent nonlinearity and their resistance, called memristance, can be time-varying based on the history of an applied voltage signal. The nonlinearity and the input history-dependent reaction of memristors are favorable in solving linearly inseparable problems with time series data[3]. Previous studies have demonstrated the potential of memristive reservoirs for temporal pattern recognition, which can be mainly divided into two types: memristor

[1]International Research Center for Neurointelligence, The University of Tokyo, Tokyo 113-0033, Japan. [2]Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan. [3]Department of Mathematical Informatics, Graduate School of Information Technology and Science, The University of Tokyo, Tokyo 113-8656, Japan. ✉email: gtanaka@g.ecc.u-tokyo.ac.jp

1

networks[14,24-26] and memristor arrays[27-30]. Memristor networks use complex dynamic behavior of interacting memristors as a reservoir state, whereas memristor arrays leverage a set of nonlinear responses of independent memristors with optional delayed feedback loops. Our target in this study is the network-type memristive reservoirs which are more difficult to design and control compared with the array-type ones. We aim to develop a systematic approach for examining the effects of system components, such as the network architecture and the nonlinear characteristics of memristors, on the computational performance of the memristor-network-based RC systems toward establishing a practical design guide and facilitating their hardware implementation.

We mathematically formulate a general system of memristor networks and develop a simulation platform for performing temporal pattern classification in the RC framework. In temporal classification tasks, the dataset is given as a set of multiple time series data and the corresponding class labels. Our purpose is to construct a pattern classifier having a high generalization ability by using the memristor-network-based RC systems, which can well predict the true class label even for unknown time series data after learning. With our simulation platform, we can examine the effects of various system factors on the computational performance of the memristive RC systems. Numerical experiments are conducted to evaluate the classification performance of the RC systems composed of simple memristor models under different system conditions for three temporal pattern classification tasks: the waveform classification[31], the electrocardiogram (ECG) classification[32], and the spoken digit recognition[33]. In the waveform classification task with sine and triangular waves, a perfect classification is achieved at the best conditions. In the ECG classification task with normal and abnormal patterns, the classification accuracy reaches a maximum of 86%. In the spoken digit recognition task with the TI-46 Word corpus, the best classification accuracy is 97.3%. These classification accuracies are comparable to those obtained by state-of-the-art ML methods. The results suggest that the RC systems with memristor networks are very promising as a building block of next-generation ML and AI hardware.

Results

RC systems with memristor networks.
A physical RC system with a memristor network is illustrated in Fig. 1a, which consists of a preprocessing part, an input part, a reservoir part, and a readout part. First, a given time series data is converted to a voltage signal in a preprocessing step, such as appropriate scaling and masking, depending on the type of data. Then, this voltage signal is fed into the voltage source of the memristor-network-based reservoir. A dynamic response of the reservoir to the input signal is obtained as time evolutions of electric currents flowing through the individual memristors. In the readout part, the current signals are converted to a matrix through a collection of reservoir states and an optional postprocessing step, and then, used to produce a system output. Only the output weight matrix W^{out} is trainable, which is optimized by linear regression so as to minimize the error between the system output and the target output. We limit the pre/post-processing methods to matrix operations, in order to shed light on the nonlinear transformation effect of the memristive reservoir.

We consider a general memristor network consisting of N_m memristors, N_n circuit nodes, and N_i voltage sources (see Fig. 1a with N_n=8, N_m=11, and $N_i = 1$). By regarding the circuit nodes as vertices and the memristor branches as edges, the connectivity of the memristors can be represented as a directional graph, described by an incidence matrix $E_m \in \mathbb{R}^{N_n \times N_m}$. The connectivity of voltage sources can be similarly described by an incidence matrix $E_i \in \mathbb{R}^{N_n \times N_i}$ (see "Methods" section).

In this study, we assume that the individual memristors in the reservoir are described by the linear drift model[34]. When an electric voltage is applied to a metal-oxide memristor, the oxygen vacancies (i.e. dopants) drift in the device as charge carriers, and shift the boundary between a doped region with a low resistance and an undoped region with a high resistance. This is simply represented by the linear drift model composed of a series of a low resistance state (LRS) with resistance R_{ON} and a high resistance state (HRS) with resistance R_{OFF} ($\gg R_{ON}$) as illustrated in Fig. 1b (see "Methods" section for details). With a variation of the length of the LRS, the total memristance changes in time. The polarity of the memristor, related to its directionality, is determined depending on whether the drift of the dopants expands or contracts the LRS[35]. For a sinusoidal input voltage, the linear drift model can exhibit a pinched hysteresis loop in the I-V curve as shown in Fig. 1c. A variation in the ON/OFF resistance ratio, $r = R_{OFF}/R_{ON}$, changes the nonlinearity of the I-V characteristics. When micro/nano-scale memristor devices are fabricated, a device-to-device variation in their electrical properties is inevitable[27,36-38]. Therefore, R_{ON} and R_{OFF} are assumed to follow a normal distribution with mean $\bar{R}_{ON}$ and $\bar{R}_{OFF}$, respectively (see "Methods" section). Figure 1d shows a normal distribution of R_{ON} with mean $\bar{R}_{ON} = 100\,\Omega$ and standard deviation $\sigma \bar{R}_{ON}$ for different values of σ controlling the degree of variability.

Figure 1e illustrates examples of four different types of network structures tested in our numerical simulations, including a ring with unidirectional polarity (Ring-UP), a ring with random polarity (Ring-RP), a random network with unidirectional polarity (Rand-UP), and a random network with random polarity (Rand-RP). These networks are considered in order to clarify the effect of randomness in network connectivity and polarity. For instance, the difference between Ring-UP and Ring-RP lies in the polarities of the memristors indicated by the arrows. To avoid a disconnected graph, we construct networks of the Rand-UP and Rand-RP types by adding long-range memristor branches to those of the Ring-UP and Rand-RP types, respectively.

We explicitly formulated circuit equations of a general memristor network based on the new modified nodal analysis method[39] as follows (see "Methods" section for derivation):

$$E_m W(E_m^\top \mathbf{\Phi}_n(t)) E_m^\top \frac{d\mathbf{\Phi}_n(t)}{dt} + E_i \mathbf{j}_i(t) = \mathbf{0}, \tag{1}$$
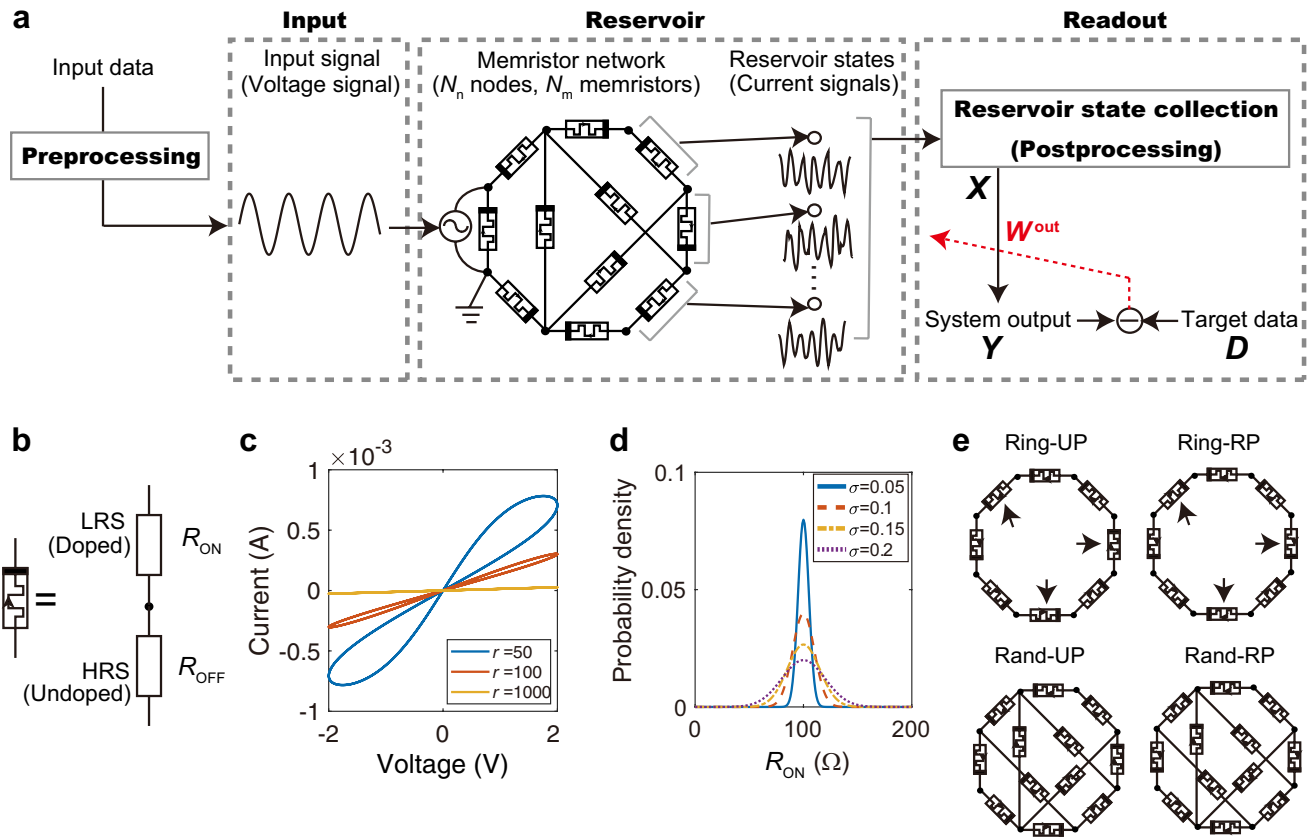
Figure 1. Memristor-network-based RC system. (**a**) System architecture composed of preprocessing, input, reservoir, and readout parts. An input time series data is preprocessed and then fed into the memristor network through the voltage source. The memristor network consists of N_m memristors, N_n circuit nodes, and N_i voltage sources. The current signals are measured as a reservoir state and processed in the readout part. The output weight matrix W^{out} is optimized by linear regression in the training process. (**b**) The linear drift model of a memristor[34], which is equivalent to a series of a low resistance state (LRS) with resistance R_{ON} and a high resistance state (HRS) with resistance $R_{OFF}(\gg R_{ON})$. The ratio between the lengths of the LRS and HRS changes in time depending on an applied voltage. (**c**) Current–voltage (I–V) characteristics of a single memristor model in response to a sinusoidal voltage signal for different values of $r = R_{OFF}/R_{ON}$. (**d**) Normal distributions of R_{ON} with mean $\bar{R}_{ON} = 100\,\Omega$ and standard deviation $\sigma\bar{R}_{ON}$ for different values of σ controlling the degree of the device-to-device variation. (**e**) Examples of four types of network structures, including a ring with unidirectional polarity (Ring-UP), a ring with random polarity (Ring-RP), a random network with unidirectional polarity (Rand-UP), and a random network with random polarity (Rand-RP). The memristors indicated by the arrows in the Ring-UP and Ring-RP types have opposite polarities. The networks of the Rand-UP and Rand-RP types are generated by randomly adding non-local memristor branches to those of the Ring-UP and Ring-RP types, respectively.

$$\frac{\mathrm{d}\Phi_n(t)}{\mathrm{d}t} - v_n(t) = 0, \tag{2}$$

$$E_i^\top v_n(t) - v_i(t) = 0, \tag{3}$$

where t represents continuous time, $\Phi_n(t) \in \mathbb{R}^{N_n}$ is a vector of nodal magnetic fluxes, $W(\cdot) \in \mathbb{R}^{N_m \times N_m}$ is a diagonal matrix of memductances (memory conductances) of the memristors as a function of fluxes, $j_i(t) \in \mathbb{R}^{N_i}$ is a vector of currents at the voltage sources, $v_n(t) \in \mathbb{R}^{N_n}$ is a vector of nodal voltages, $v_i(t) \in \mathbb{R}^{N_i}$ is a vector of voltages at the voltage sources. This set of equations with respect to the state variables ($v_n(t)$, $\Phi_n(t)$, and j_i) are categorized into a differential-algebraic system of equations (DAEs), where differential and algebraic equations are mixed. In general, DAEs are more difficult to solve than ordinary differential equations (ODEs), because the Jacobian matrix (i.e. the first-order derivative) used for numerical integration becomes singular[40]. By setting E_m, E_i, v_i, and W depending on the reservoir network structure and the memristor models, concrete system equations can be obtained from Eqs. (1)–(3) (see "Methods" section). In our simulation platform on MATLAB[41], these equations are derived by symbolic computation and solved by using the DAE solver. The dynamic state of the reservoir is represented by time courses of vector $j_m(t) \in \mathbb{R}^{N_m}$ of electric currents flowing through the memristors. The reservoir states are used to produce the system output in the readout processing (see "Methods" section).
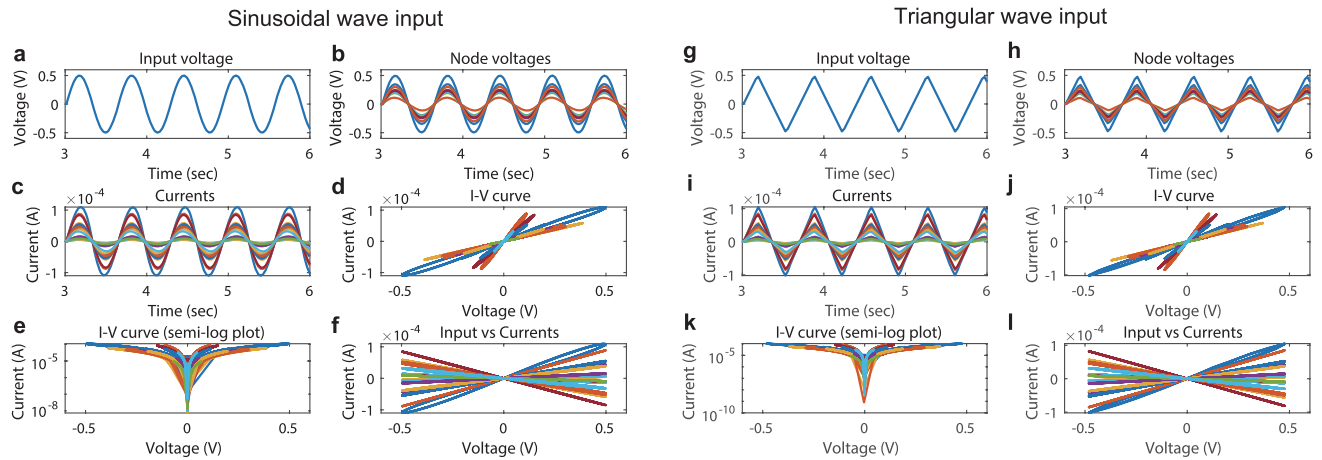
Figure 2. Dynamical behavior of a memristor-network-based reservoir driven by waveform signals. The network structure is the Rand-UP type, the average ON/OFF resistance ratio is $\bar{r} = 50$, and the degree of variability is $\sigma = 0.2$. (**a**) A sine wave input. (**b**) The nodal voltages. (**c**) The currents on the memristor branches. (**d**) The I–V curves. (**e**) The same as (**d**), but the absolute current on a semi-log plot. (**f**) The currents plotted against the nodal voltages. (**g**)–(**l**) The same as (**a**)–(**f**), but for a triangular wave input.

Waveform classification.

The waveform classification task has often been used as a benchmark task for evaluating computational performance of physical RC systems[29,31,42–44]. We consider a two-class classification problem with 100 sine and 100 triangular waves having the same amplitude and different frequencies. The frequency of each waveform was randomly generated from a uniform distribution in a certain range (see "Methods" section). Some data were used for training and the other data were for testing. These waveform data were converted to voltage signals and then fed into the memristor-network-based reservoir. The parameter conditions used for numerical experiments are listed in Table 1.

Figure 2 demonstrates the responses of a memristor network of the Rand-UP type to sine and triangular input voltage signals when $\bar{r} = \bar{R}_{\mathrm{OFF}}/\bar{R}_{\mathrm{ON}} = 50$ and $\sigma = 0.2$. Figure 2a–c show the sine wave input, the nodal voltages, and the currents on the memristor branches, respectively. Due to the random connectivity of the memristors and the device-to-device variation, the individual memristors exhibit different nonlinear I–V characteristics as shown in Fig. 2d,e. Figure 2f shows the relationship between the input voltage signal and the branch currents, indicating the input-output transformation realized by the reservoir. The corresponding figures for a triangular wave input are shown in Fig. 2g–l (see Supplementary Figs. 1–5 for other system conditions). The N_{m} current signals were converted to sequences of length $L_{\mathrm{out}} = 100$ by sampling and then used to form a state collection matrix X for the readout processing (see "Methods" section).

Figure 3 shows the results of the waveform classification task. We tested the four network structures shown in Fig. 1b: Ring-UP with (magenta triangles), Ring-RP (green diamonds), Rand-UP (blue squares), and Rand-RP (red circles). We evaluated the classification accuracies for 10 random network realizations based on the 10-fold cross validation. Figure 3a–d plot the average value over the 100 trials together with the error bar indicating the standard error. Figure 3a shows the classification accuracies when the number of training data, N_{train}, is varied. It is remarkable that the networks of the Rand-UP and Rand-RP types achieve the 100% accuracy independently of N_{train} due to the rich variety in the reservoir states (see Fig. 2; Supplementary Fig. 3). The performances of the Ring-UP and Ring-RP types are inferior to those of the Rand-UP and Rand-RP types, but much higher than the baseline accuracy obtained by a linear classifier. This indicates the benefit of the nonlinear signal transformation by the physical reservoir. Figure 3b shows that the classification performance is not sensitive to the change in the variability parameter σ, suggesting a variability-tolerant property of the memristive RC system. Figure 3c demonstrates that the classification accuracy monotonically increases with the number of signals, N_{x} ($\leq N_{\mathrm{m}}$), used for the readout. In other words, a higher-dimensional reservoir state yields a better accuracy. The accuracy reaches the perfect level when $N_{\mathrm{x}} \geq 14$. Figure 3d shows that the average ON/OFF resistance ratio $\bar{r}$ related to the nonlinearity of the I–V characteristics is a key factor influencing the classification performance, which is maximized at the intermediate values around $\bar{r} = 5 \times 10^2$–10^3. Figure 3e,f show the confusion matrices when $\bar{r} = 50$ and $\bar{r} = 10^4$, respectively. The results imply that the misclassifications are linked to some failures in the training for $\bar{r} = 50$ whereas they are caused by over-training for $\bar{r} = 10^4$.

ECG classification.

An electrocardiogram (ECG) is an electric signal associated with heartbeats, which is often used for health checks and cardiac disease detections. In a normal state, the ECG signal shows a repetition of typical waveforms called the QRS complex[45] due to depolarization and repolarization of the membrane potentials of the cardiac muscle cells. Irregular ECG signals often correspond to abnormal cardiac behavior caused by diseases such as ischemia and myocardial infarction. An ECG-based heartbeat classification task is aimed at separating abnormal ECG signals from normal ones[32,46]. We used the ECG200 dataset formatted in the UCR Time Series Classification Archive[47], which contains a total of 200 samples of ECG segment data, including 100 samples for training and the other 100 samples for testing.
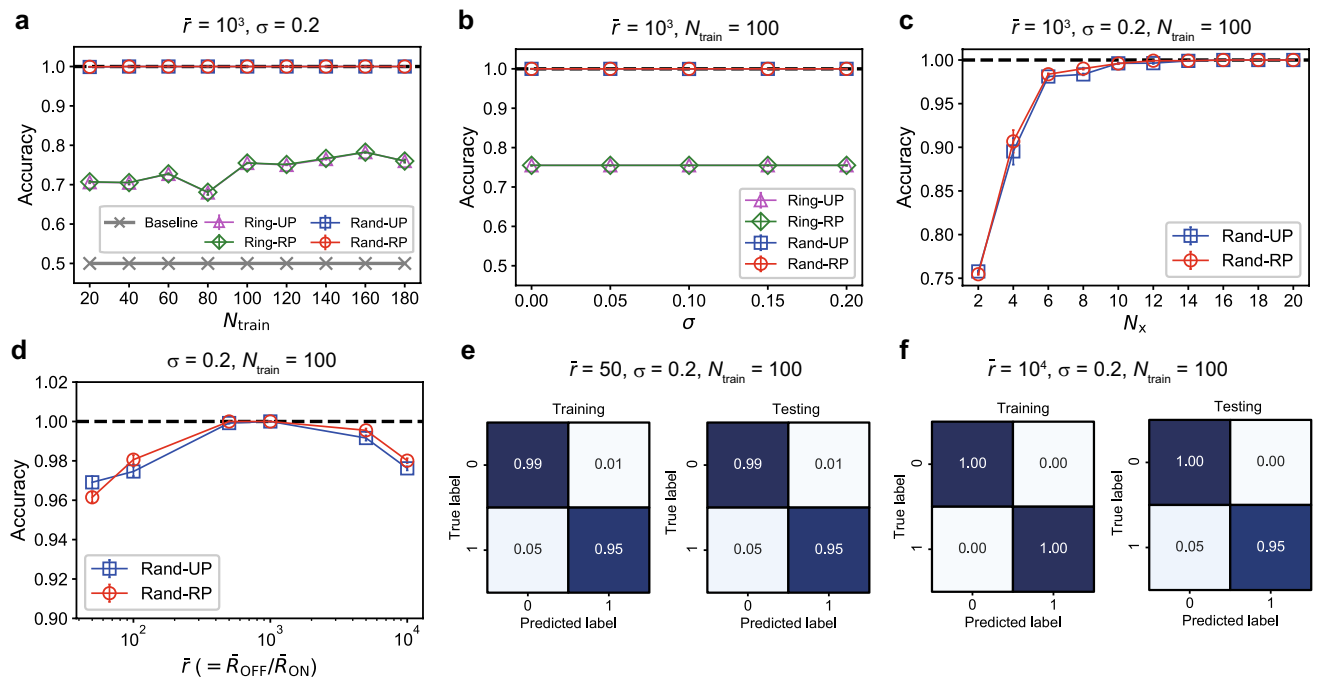
Figure 3. Results of the waveform classification task. (**a**) The classification accuracies for the four different network types when N_{train} is varied. (**b**) The classification accuracies plotted against the variability parameter σ. (**c**) The classification accuracies when N_x signals out of $N_m (= 20)$ current signals were used for the readout. (**d**) The classification accuracies plotted against the average ON/OFF resistance ratio $\bar{r}$. (**e**) The confusion matrices for $\bar{r} = 50$. (**f**) The confusion matrices for $\bar{r} = 10^4$.

Figure 4a shows a preprocessing step for each ECG data of length $L_{\text{data}} (= 96)$. The one-dimensional vector representing the original data was transformed into a 2D masked data by using a random mask of size $S_{\text{mask}} (= 50)$, each element of which is -1 or 1. This randomization process corresponds to a multiplication of input data by random input weights in echo state networks[3,5]. The masked data was separated into L_{data} column-wise sequences, each of which was converted to a voltage signal and then fed into the memristor-network-based reservoir in the order of the column index sequentially. After the injection of each sequence, we reset the reservoir state. Figure 4b demonstrates a dynamic response of the reservoir to an input time series. The $N_m (= 20)$ current signals were transformed into the sequences of length $L_{\text{out}} (= 50)$ for constructing a state collection matrix X in two possible ways as shown in Fig. 4c. In Case (i), the transposed matrices of size $N_m \times L_{\text{out}}$ were stacked vertically for all the inputs to yield a state collection matrix $X \in \mathbb{R}^{N_m L_{\text{data}} \times L_{\text{out}}}$. In Case (ii), the N_m sequences were concatenated into a one-dimensional sequence and those for all the inputs were further concatenated to form a state collection matrix (vector) $X \in \mathbb{R}^{N_m L_{\text{data}} L_{\text{out}} \times 1}$. The parameter conditions for numerical experiments are listed in Table 1.

Figure 5 shows the results of the ECG classification task. Each plot corresponds to the average accuracy over 10 random network realizations and the error bar indicates the standard error. Figure 5a plots the classification accuracies for the four types of reservoir structures combined with the postprocessing in Case (i) (monochrome marks) and Case (ii) (colored marks) when $\bar{r}$ is varied. The baseline accuracy (gray crosses) obtained by a linear classifier is 64%, meaning that all the testing data were classified as normal patterns. We can see that the performance with the postprocessing in Case (ii) are much better than those in Case (i). If the postprocessing method in Case (ii) is employed, the classification performance is kept at a high level against the variation of $\bar{r}$, indicating the robustness against the device-to-device variation as shown in Fig. 5b. The effect of the number of signals used for the readout processing, $N_x (\leq N_m)$, on the performance is shown in Fig. 5c. By increasing N_x, the performance is significantly improved in Case (i) but only slightly in Case (ii) (see Supplementary Fig. 6). We observe that the maximum input voltage V_{max} significantly influences the classification performance in Case (ii) as shown in Fig. 5d. The classification accuracy reaches 86% when the network structure is the Rand-RP type and $V_{\text{max}} = 0.02$ V. This performance is comparable to that of the other ML methods, ranging between 77 and 92%[49]. We note that the number of trained weights is 1920 in Case (i) and 96,000 in Case (ii). The difference in the number of trainable weights is a potential cause of the gap in the classification performance. These results indicate that the classification performance can largely depend on the postprocessing method even when the same reservoir states are used. It is a future issue to fully understand the influence of the postprocessing on the performance, which is linked with a tradeoff between classification accuracy and computational time for learning.

Spoken digit recognition. The isolated spoken digit recognition has been widely used as a benchmark task for testing the classification ability of RC systems[10,33,43]. The sound dataset from the NIST TI-46 Word corpus[50] contains 500 sound waveform data corresponding to 10 utterances of 10 digits ("zero" to "nine"), spo-
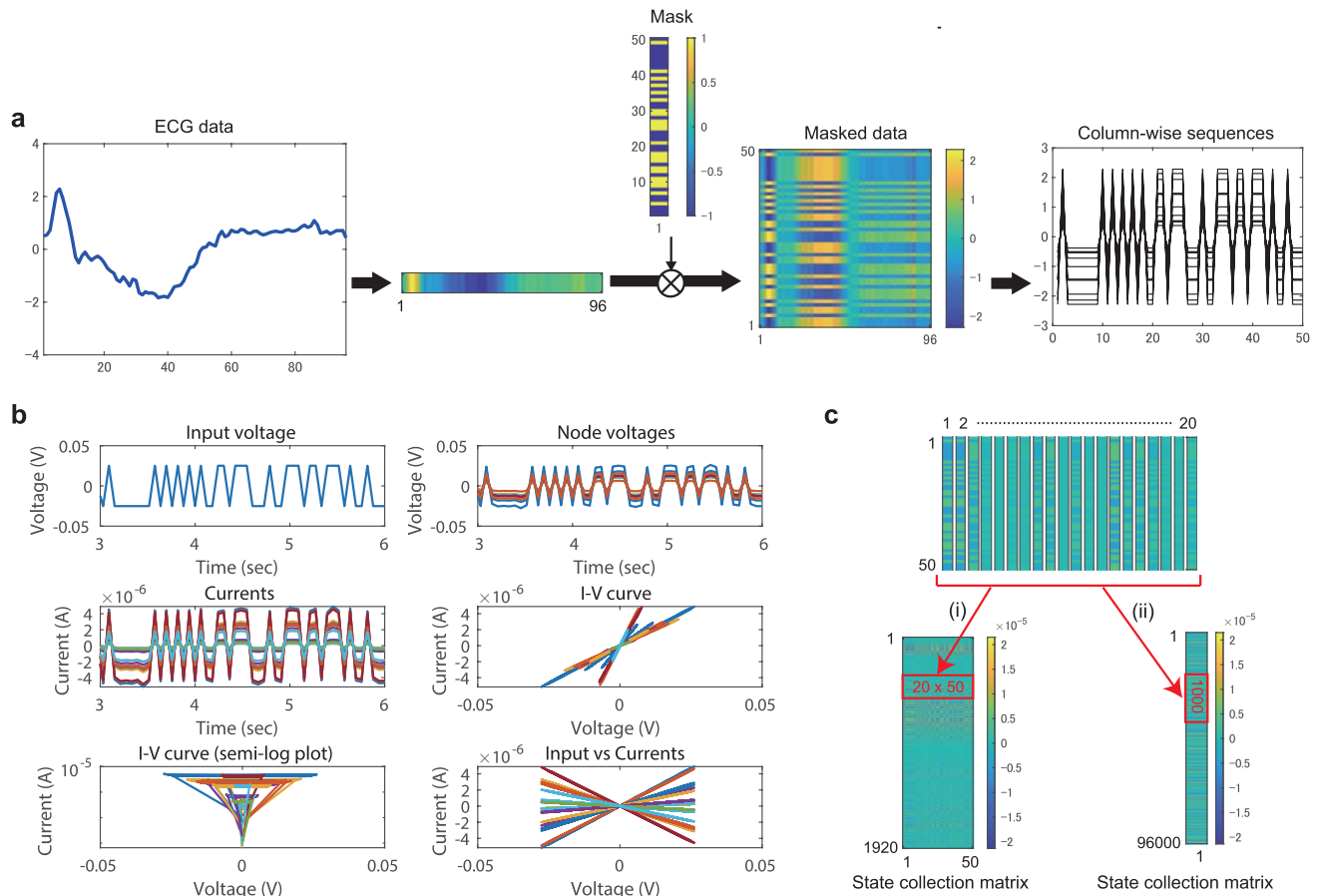
Figure 4. Signal processing for the ECG classification task. (**a**) The preprocessing method. An original ECG data is transformed with a binary mask of $+1$ and -1 into a masked data. Each of the L_{data} (=96) column-wise sequences is fed into the memristive reservoir. (**b**) A dynamic response of the memristive reservoir of the Rand-UP type when $\bar{r} = 50$ and $\sigma = 0.2$. (**c**) The postprocessing method. The N_m (= 20) sequences of length L_{out} (= 50) are obtained from the reservoir for each input sequence. In Case (i), the reservoir outputs of size $L_{out} \times N_m$ are rotated and stacked for all the reservoir inputs to form a state collection matrix. In Case (ii), the reservoir outputs are transformed into a one-dimensional sequence and concatenated for all the reservoir inputs to form a state collection matrix.

ken by 5 different speakers (ID: 1,2,5,6,7)[51]. The aim of this 10-class classification problem is to correctly predict the true digit from a sound signal.

Each sound data was converted to a cochleagram by using the Lyon's passive ear model after an elimination of silence periods[48] as shown in Fig. 6a (see "Methods" section for details). The data length L_{data} ranges from 48 to 102. The cochleagram was transformed into a masked data with a random binary mask of size $S_{mask} \times N_f$. Each of the N_f column-wise sequences is given to the reservoir after an appropriate scaling. An example of the dynamic behavior of the reservoir is shown in Fig. 6b. The N_m (=20) currents from the reservoir are transformed into the sequences of length L_{out} (=100) by sampling. They are concatenated into a one-dimensional vector of size $N_m L_{out}$ as shown in Fig. 6c. These column vectors are collected for all the inputs to construct a state collection matrix $X \in \mathbb{R}^{N_m L_{out} \times L_{data}}$. The parameter conditions for numerical experiments are listed in Table 1.

Figure 7 shows the results of the spoken digit recognition task. Each plot is the average classification accuracy over 10 random network realizations, evaluated based on the 10-fold cross validation, and the error bar indicates the standard error. Figure 7a demonstrates the classification accuracies when the number of training data, N_{train}, is varied. The results obtained by the memristive RC systems with $S_{mask} = 100$ are indicated by the colored marks for the four different structure types, i.e. Ring-UP, Ring-RP, Rand-UP, and Rand-RP. For comparison, the performance of linear classifiers (LCs) applied to the masked data are plotted for different mask size $S_{mask} = 10, 25, 50, 100$. The classification accuracies for all the tested methods are improved by increasing N_{train}. The performance of the LCs increases with the mask size, but peaks out at around $S_{mask} = 100$ (see Supplementary Fig. 7). The memristive RC systems produce better accuracies than the LC in the case of $S_{mask} = 100$, which highlights the benefit of the nonlinear transformation by the memristive reservoir[52]. The network structures of the Rand-UP and Rand-RP types yield better accuracies than those of the Ring-UP and Ring-RP types, owing to the diversity in the reservoir dynamics. Figure 7b shows that a smaller average ON/OFF ratio $\bar{r}$, corresponding to a stronger nonlinearity of I-V characteristics on average, leads to better classification performance for the Rand-UP and Rand-RP types when $N_{train} = 450$. The best accuracy reaches 97.3% when the ON/OFF ratio is
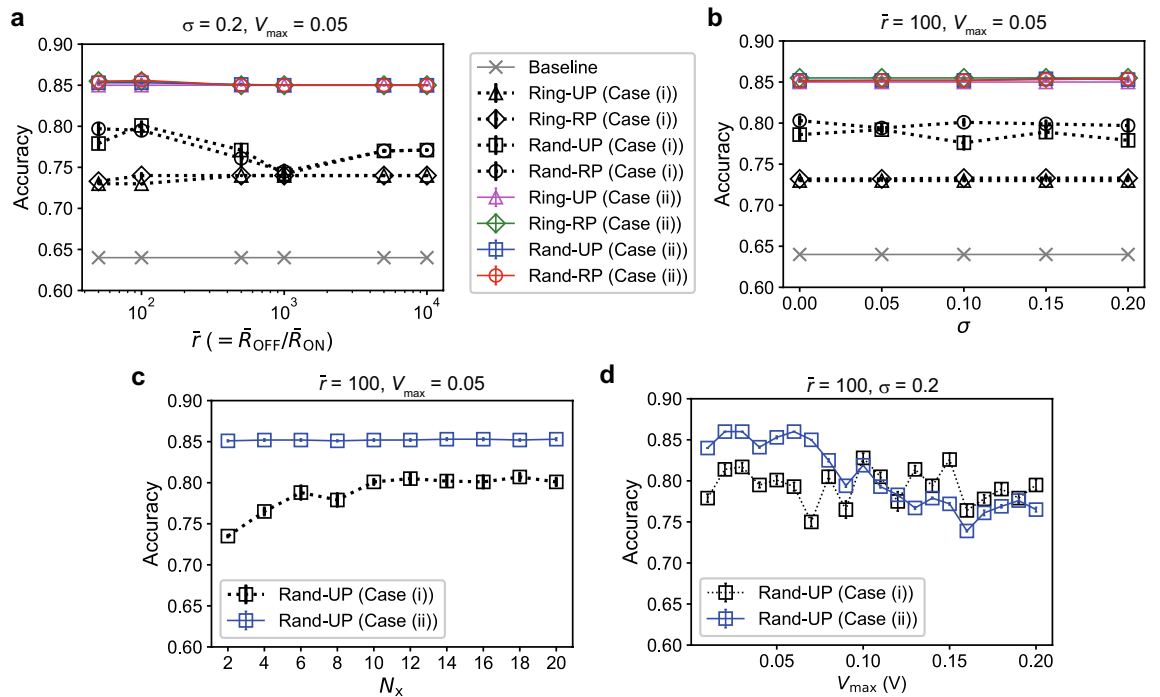
Figure 5. Results of the ECG classification task. (**a**) The classification accuracies for a variation of the average ON/OFF ratio $\bar{r} = \bar{R}_{OFF}/\bar{R}_{ON}$. (**b**) The classification accuracies plotted against the variability parameter σ. The meanings of the marks are the same as those in (**a**). (**c**) The classification accuracies when N_x signals out of $N_m (= 20)$ current signals were used for the readout. (**d**) The dependence of the accuracy on the maximum input voltage V_{max}.

$\bar{r} = 50$ and the network type is Rand-RP, which is comparable to that achieved by other physical RC systems[10]. For a specific separation between the training and testing datasets among the 10 cases for the cross validation, we obtained 99.8% accuracy on average over 10 different memristor networks. This is the first report that the memristor networks can yield such high performance in this task. Figure 7c indicates that the classification performance is kept at a high level irrespective of a change in the variability level σ of the memristors. Figure 7d demonstrates the results when N_x signals out of N_m signals were used for the readout process, indicating that a higher dimensional reservoir state contributes more to increasing the classification accuracy and $N_x \sim 10$ is enough for obtaining the maximum accuracy. Our analysis on the confusion matrix shown in Fig. 7e reveals that a majority of misclassifications occur when the sound data of digit 2 is incorrectly classified as digit 1 or 8. By overcoming this issue through modifications in the signal processing parts, the performance could be further improved.

Discussion

In this study, we have provided the explicit mathematical formula of the general memristor networks and developed the simulation platform for performing temporal pattern classifications with the memristor-network-based RC systems. The platform enables a considerable search of various system conditions and an identification of the key system component for improving classification performance. The results on the three classification tasks indicate that the randomness of the network connectivity (i.e. Rand-UP and Rand-RP) is favorable for generating diverse nonlinear responses to the input signal and achieving the excellent classification performance compared to the other two (i.e. Ring-UP and Ring-RP). The results have also lead to a new finding that the ON/OFF resistance ratio, controlling the nonlinearity of memristors, can have a large impact on the classification accuracy. Although the best ON/OFF ratios are different depending on the task (see Figs. 3d, 5a, and 7b), all of them ($\bar{r} \sim 50$ to 500) are within the feasible range reported for real memristor devices[53].

The memristor network model that we formulated is quite general, and therefore our simulation platform is easily extendable. In this study, we have used at most 20 memristors in the reservoir to test many different system designs while saving the simulation time. Tackling a more complex pattern recognition task with a larger number of memristors is one of the future works. For this purpose, it is an option to inject multiple different input signals into multiple nodes in the memristive reservoir. Any network connectivity of the memristors, including the four types investigated in this study, can be examined conveniently by setting the incidence matrices. We have assumed the ideal linear drift model for the individual memristors. It can be replaced with any other model by deriving or approximating its memductance as a function of the magnetic flux[39]. In the tested cases, the device-to-device variability following a normal distribution does not cause a degradation in the classification accuracy, implying a reliable computation with heterogeneous units. As far as we checked, the network topology seems to play a decisive role in the input transformation by the reservoir rather than the device-to-device variability. If the shapes and the ranges of the distributions of R_{ON} and R_{OFF} are estimated through measurements with specific
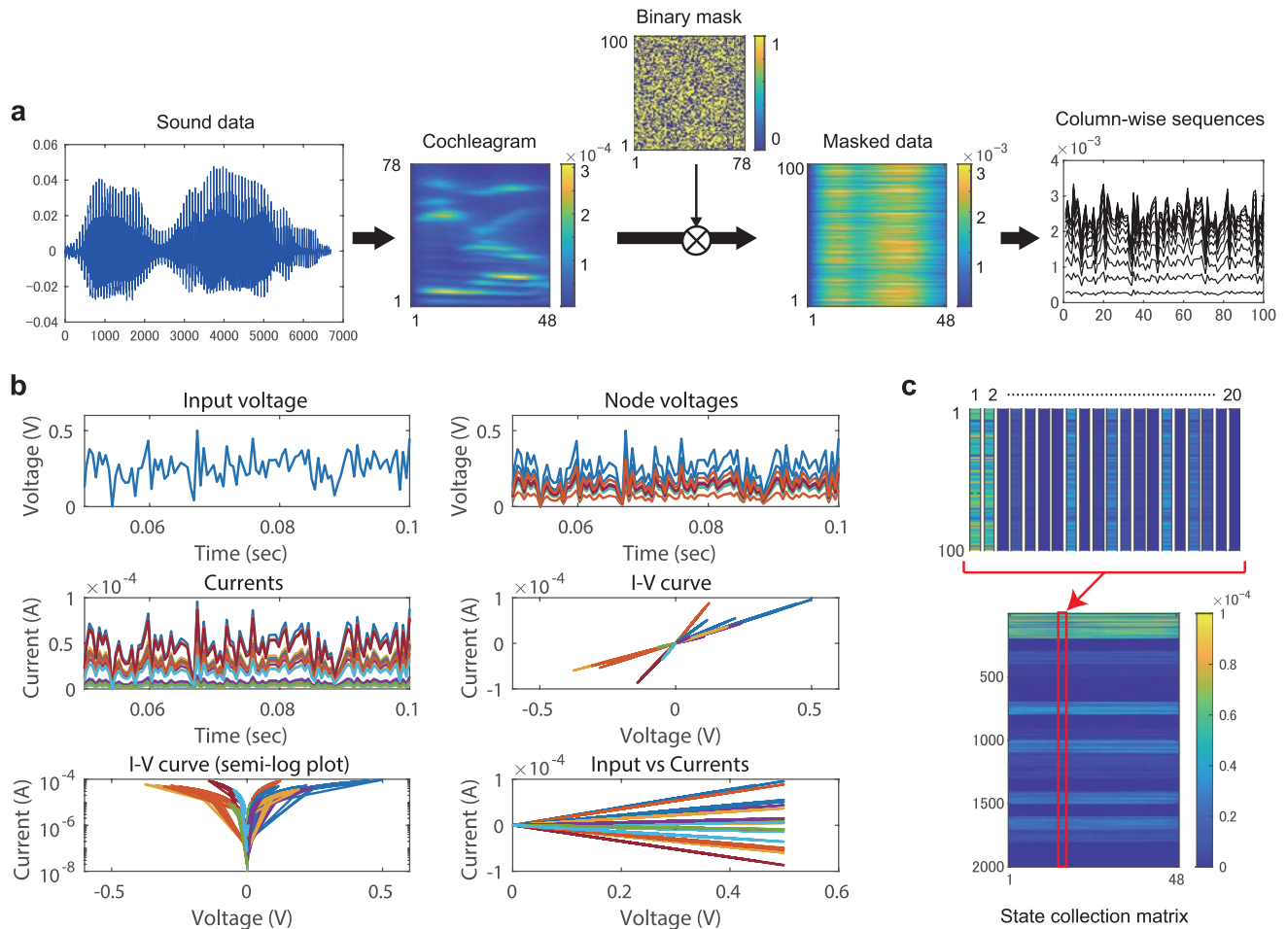
Figure 6. Signal processing for the spoken digit recognition task. (**a**) The preprocessing method. A sound data is transformed into a cochleagram of size $N_f (= 78) \times L_{data} (= 48)$ by using the Lyon's passive ear model[48] and then multiplied by a random binary mask of size $S_{mask} (= 100) \times N_f$. The masked data is separated into L_{data} column-wise sequences. (**b**) An example of the dynamic response of the memristive reservoir of the Rand-UP type when $\bar{r} = 50$ and $\sigma = 0.2$. (**c**) The postprocessing method. The N_m current signals are transformed into the sequences of length $L_{out} (= 100)$ for each input, which are concatenated into one-dimensional vector of size $N_m L_{out}$. The vectors for all the L_{data} inputs are concatenated to make the state collection matrix.

devices, they can be easily incorporated into the memristor network model and the simulation platform. We have leveraged appropriate pre/post-processing methods for each task to get high classification accuracies comparable to those obtained by the other ML methods. It would be possible to further improve the classification accuracy by adding more advanced operations in the readout part, but our signal conversion methods limited to matrix operations are reasonable for maintaining the merit of low learning cost.

The memristive reservoir of the network type is an attractive option for hardware implementation, because the number of possible network structures producing different dynamic responses to input signals can be drastically increased by scaling up the network size. This structural diversity is the major advantage of the network-type reservoir over the array-type and single-node-type reservoirs[10]. Since only a part of system components are controllable in real memristive devices and materials, it is a significant issue to find better system settings under practical constraints[54]. From a scientific perspective, our final goal is to comprehensively understand the relationship between dynamical properties and information processing capacity of memristor networks. The dynamical properties can be investigated through spectral analysis[21] and nonlinear time series analysis[55]. The features related to information processing can be evaluated by computing relevant measures such as memory capacity[56], kernel quality[57], and other capacity scores[58]. Our simulation platform can contribute to both these purposes. A target for the future is to integrate numerical and experimental approaches for establishing a design principle of memristive reservoirs, thereby accelerating the development of energy-efficient RC-based ML hardware.

Methods

Memristor networks and incidence matrices.
The physical reservoir in this study is a general memristor network consisting of N_m memristors, N_n circuit nodes, and N_i input voltage sources (see Fig. 1a). By regarding the circuit nodes as vertices v_n for $n = 1, \ldots, N_n$ and the memristor branches as edges e_m for $m = 1, \ldots, N_m$, a network structure of memristors can be represented as a directional graph with an incidence
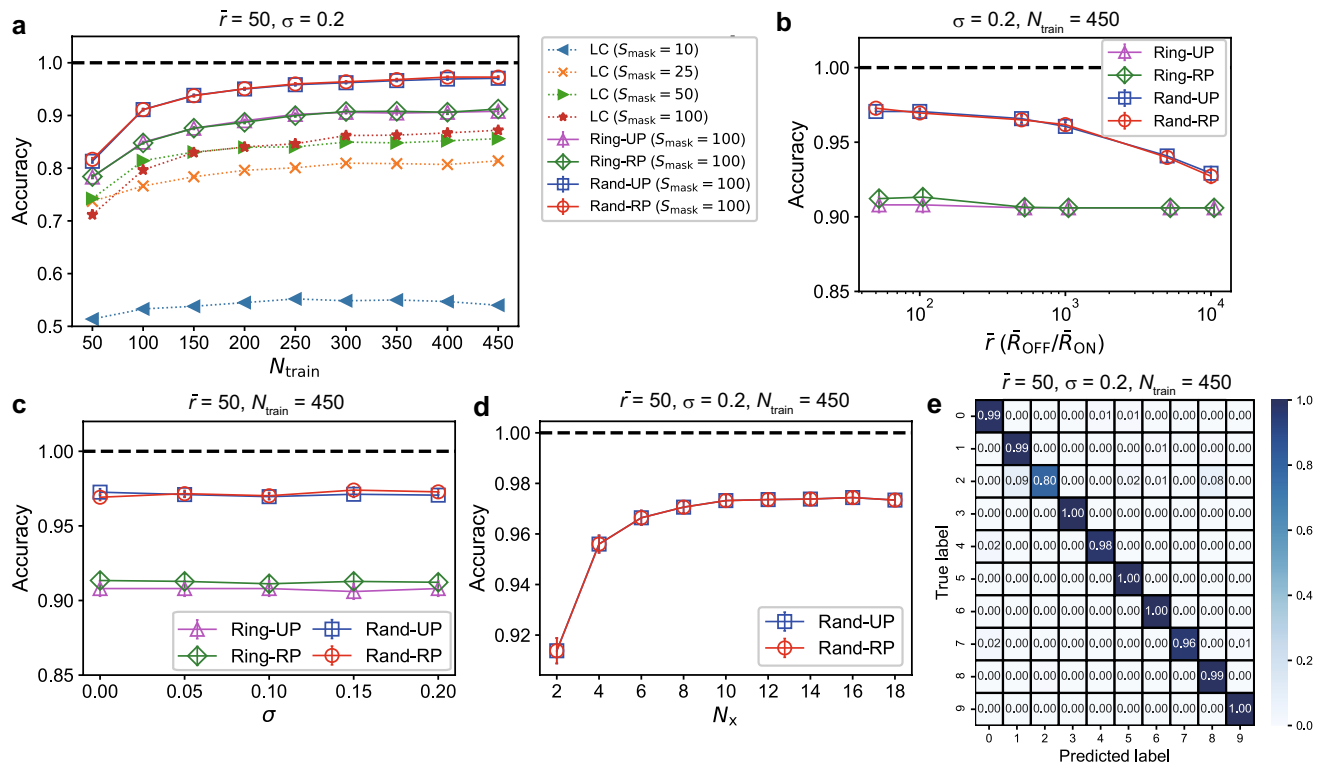
Figure 7. Results of the spoken digit recognition task. (**a**) The classification accuracies when the number of training data N_{train} is varied. The results with the linear classifier (LC) include the four cases with $S_{\text{mask}} = 10, 25, 50, 100$. The results with the memristive RC systems include the four cases with Ring-UP, Ring-RP, Rand-UP, and Rand-RP, where $S_{\text{mask}} = 100$. (**b**) The classification accuracies plotted against the average ON/OFF ratio $\bar{r}$. (**c**) The classification accuracies plotted against the variability parameter σ. (**d**) The classification accuracies when N_{x} signals out of $N_{\text{m}}(= 20)$ current signals were used for the readout. (**e**) The confusion matrix for testing data.

Parameter	Symbol	Values		
Number of memristors	N_{m}	20		
Number of circuit nodes	N_{n}	10 (Ring-UP/RP), 20 (Rand-UP/RP)		
Number of voltage sources	N_{i}	1		
Device length	D	10^{-8} m[34]		
Average ion mobility	μ_v	10^{-14} m^{2}s^{-1}V^{-1}[34]		
Mean resistance (Doped)	$\bar{R}_{\text{ON}}$	100 Ω[34]		
Mean resistance (Undoped)	$\bar{R}_{\text{OFF}}$	5 k–1M Ω		
Device variation degree	σ	0–0.2		
		Waveform	**ECG**	**Spoken digit**
Number of classes	N_{c}	2	2	10
Length of time series data	L_{data}	100	96	48–102
Maximum voltage level	V_{max}	0.5 V	0.05 V	0.5 V
Mask size	S_{mask}	–	50	100
Length of inputs	L_{in}	100	50	100
Length of outputs	L_{out}	100	50	100
Time length for relaxation	Δt_{relax}	3 s	3 s	0.05 s
Time length for main signal	Δt_{main}	3 s	3 s	0.05 s

Table 1. List of system parameters and their values.

matrix $E_m \in \mathbb{R}^{N_n \times N_m}$. If a memristor branch e_m $(m = 1, \ldots, N_m)$ connects a starting node v_k and an ending node v_l, then the incidence matrix is defined as follows:

$$E_m(n, m) = \begin{cases} -1 & (n = k) \\ 1 & (n = l) \\ 0 & (n \neq k, l) \end{cases}. \tag{4}$$

In a similar way, an incidence matrix $E_i \in \mathbb{R}^{N_n \times N_i}$ for the positions of the voltage sources can be defined. If a voltage source i connects a starting node k and an ending node l, then

$$E_i(n, i) = \begin{cases} -1 & (n = k) \\ 1 & (n = l) \\ 0 & (n \neq k, l) \end{cases}. \tag{5}$$

Single memristor model. The linear drift memristor model[34] was adopted for our simulations. It is assumed that a memristor device consists of a doped region (or LRS) with low resistance R_{ON} and an undoped region (or HRS) with high resistance R_{OFF} (see Fig. 1c). The memristance M at time t is written as follows:

$$M(w(t)) = \left(R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right) \right), \tag{6}$$

where D and $w(t)$ represent the lengths of the memristor device and the doped region, respectively. When a voltage signal $v(t)$ is applied to the memristor, the current $j(t)$ flowing through the memristor and the time evolution of the internal variable $w(t)$ are described as follows:

$$j(t) = v(t)/M(w(t)), \tag{7}$$

$$\frac{dw(t)}{dt} = \mu_v \frac{R_{ON}}{D} j(t), \tag{8}$$

where μ_v represents the average ion mobility. Since the memristance can be written as $M(q) = d\Phi(q)/dq$ using the charge q and the magnetic flux Φ, the memductance $W(q) = dq(\Phi)/d\Phi$ is represented as follows[59]:

$$W(\Phi) = (M(w(0))^2 - 2a\Phi)^{-1/2}, \tag{9}$$

where $M(w(0))$ is the memristance at the initial condition. The constant parameter a is obtained as follows:

$$a = \frac{\mu_v R_{ON}(R_{OFF} - R_{ON})}{D^2} = \mu_v R_{ON}(r - 1)/D^2. \tag{10}$$

where r is the ON/OFF resistance ratio.

Variability of memristors. The device-to-device variation can be represented as parameter distributions in the single memristor model. In our study, the resistance R_{ON} of the LRS was generated from a normal distribution with mean $\bar{R}_{ON}$ and standard deviation $\sigma \bar{R}_{on}$. The resistance R_{OFF} of the HRS was generated from a normal distribution with mean $\bar{R}_{OFF}(= \bar{r}\bar{R}_{ON})$ and standard deviation $2\sigma \bar{R}_{OFF}$. It was assumed that R_{OFF} has larger variability than R_{on}[38]. The average ON/OFF resistance ratio $\bar{r}$ controls the nonlinearity of the I-V characteristic (see Fig. 1d). The variability parameter σ controls the degree of device-to-device variation (see Fig. 1e). The initial condition $w(0)$ was generated from a normal distribution with mean $D/10$ and standard deviation $\sigma D/10$.

Formulation of memristor networks. A general memristor network can be formulated as Eqs. (1)–(3) by using the new modified nodal analysis[39]. First, the Kirchhoff's current law gives a set of N_m equations as follows:

$$E_m \mathbf{j}_m(t) + E_i \mathbf{j}_i(t) = \mathbf{0}, \tag{11}$$

where $\mathbf{j}_m(t) \in \mathbb{R}^{N_m}$ is the vector of currents flowing on the N_m memristors and $\mathbf{j}_i(t) \in \mathbb{R}^{N_i}$ is the vector of currents flowing on the N_i voltage sources at time t. The current vector can be expressed as follows:

$$\mathbf{j}_m = \frac{d}{dt} \mathbf{q}_m(E_m^\top \mathbf{\Phi}_n(t)), \tag{12}$$

where $\mathbf{q}_m \in \mathbb{R}^{N_m}$ and $\mathbf{\Phi}_n \in \mathbb{R}^{N_n}$ denote the vectors of the charges and fluxes, respectively. By substituting Eq. (12) into Eq. (11), Eq. (1) is derived. Using the first-order linearization, Eq. (1) can be rewritten as follows:

$$E_m W_m(E_m^\top \mathbf{\Phi}_n(t)) E_m^\top \frac{d\mathbf{\Phi}_n(t)}{dt} + E_i \mathbf{j}_i(t) = \mathbf{0}, \tag{13}$$

where $W_m \in \mathbb{R}^{N_m \times N_m}$ denotes the diagonal matrix whose elements are the memductances of the memristors. For the linear drift model, the diagonal elements are given by Eq. (9). Second, the Faraday's law gives Eq. (2) which is a set of N_n differential equations. Third, the Kirchhoff's voltage law gives Eq. (3) which is a set of N_i algebraic equations. As a result, the network of the linear drift memristor models is formulated as a set of differential-algebraic

equations (DAEs) consisting of Eqs. (13), (2), and (3). The DAE was numerically solved with the DAE solver ode15i in the software package MATLAB[41]. For a technical reason, a null signal in time period $[0, \Delta t_{\text{relax}}]$ for relaxation was added to the main input voltage signal in time period $[\Delta t_{\text{relax}}, \Delta t_{\text{relax}} + \Delta t_{\text{main}}]$ in our simulations. The main input signal with a maximum absolute voltage V_{max} was generated from an input time series of length L_{in} after scaling and interpolation. The parameter values are listed in Table 1.

Readout. The vector $\mathbf{j}_{\text{m}}(t)$ of N_{m} currents is measured from the memristor network in time period $[\Delta t_{\text{relax}}, \Delta t_{\text{relax}} + \Delta t_{\text{main}}]$. All the N_{m} currents (or partial N_{x} currents) are used to construct a state collection matrix X for the readout through a task-specific postprocessing (see "Methods" section for specific tasks). Once the state collection matrix X_k corresponding to the k-th input time series data is obtained for $k = 1, \ldots, N_{\text{train}}$, the overall state collection matrix is obtained as follows:

$$X_{\text{train}} = [X_1, \ldots, X_k, \ldots, X_{N_{\text{train}}}]. \tag{14}$$

Correspondingly, the overall teacher collection matrix is set as follows:

$$D_{\text{train}} = [D_1, \ldots, D_k, \ldots, D_{N_{\text{train}}}], \tag{15}$$

where D_k is the teacher matrix for the k-th input data. The column count of D_k is the same as that of X_k and its row count is the same as the number of classes, N_{c}, in the classification task. If the k-th input data has label $c_k \in \{1, \ldots, N_{\text{c}}\}$, then each column of D_k is given by a one-hot vector $[0, \ldots, 1, \ldots, 0]^\top \in \mathbb{R}^{N_{\text{c}}}$ where only the c_k-th element is 1 and the others are 0.

In the training (i.e. learning) phase, the error between the system output $Y_{\text{train}} = W^{\text{out}} X_{\text{train}}$ and the target output D_{train} is minimized by linear regression. An optimal solution is obtained as follows:

$$\hat{W}^{\text{out}} = D_{\text{train}}^\dagger X_{\text{train}}, \tag{16}$$

where $D_{\text{train}}^\dagger$ represents the pseudoinverse matrix of D_{train}.

In the testing (i.e. inference) phase, the classification ability of the trained system is evaluated for N_{test} unknown time series data in the testing dataset. For the testing dataset, X_{test} and D_{test} are composed in a similar way to Eqs. (14) and (15), respectively. The system outputs for the testing data are computed as $Y_{\text{test}} = [Y_1, \ldots, Y_k, \ldots, Y_{\text{test}}] = \hat{W}^{\text{out}} X_{\text{test}}$. From each column of Y_k, the row index that gives the largest value is recorded. The predicted class $\hat{c}_k$ is determined as the most frequent value in the recorded row indices for all the columns of Y_k. By comparing the true class c_k and the predicted class $\hat{c}_k$ for all the N_{test} testing data, the classification accuracy is computed.

Waveform classification. The sine and triangular waveform data were generated with the following equations:

$$s_{\text{sin}}(t) = \sin(2\pi f t) \quad \text{for } t \in [0, 1],$$
$$s_{\text{tri}}(t) = \text{sawtooth}(2\pi f t + \pi/2, 1/2) \quad \text{for } t \in [0, 1],$$

where the "sawtooth" is a function in MATLAB and the frequency f was randomly drawn from the uniform distribution in $[f_0(1 - \delta), f_0(1 + \delta)]$. The parameter values were set at $f_0 = 5$ and $\delta = 0.4$. The whole dataset includes 100 sine and 100 triangular waveform data, all of which have length $L_{\text{data}}(= 100)$. It was separated into N_{train} training data including randomly chosen $N_{\text{train}}/2$ data from each class and the remaining $N_{\text{test}}(= 200 - N_{\text{train}})$ testing data.

In the postprocessing step, the current signals were converted to sequences of length L_{out} via sampling, and then to positive-valued sequences by taking their absolute values. The state collection matrix $X \in \mathbb{R}^{N_{\text{m}} \times L_{\text{out}}}$ is constructed by concatenating those sequences for each input time series. The 10-fold cross validation was used to evaluate the classification accuracy.

ECG classification. This task was performed with the ECG200 dataset from the UCR Timeseries Classification Archive[47], which was originally formatted by Olszewski[60]. The dataset contains a total of 200 ECG signal data having class labels corresponding to normal and abnormal heartbeats. The dataset consists of $N_{\text{train}}(= 100)$ training data and $N_{\text{test}}(= 100)$ testing data, all of which have length $L_{\text{data}}(= 96)$. The training dataset includes 31 abnormal and 69 normal data. The testing dataset includes 36 abnormal and 64 normal data.

Spoken digit recognition. This task was performed with the NIST TI-46 Word corpus collected at Texas Instruments in a quiet acoustic enclosure using an Electro-Voice RE-16 Dynamic Cardioid microphone at 12.5 kHz sample rate with 12-bit quantization[50]. From each time series data, the main sound signal of length L_{data} were extracted by removing the silence part with the VOICEBOX, a Speech Processing Toolbox for MATLAB[61]. The length L_{data} differs depending on the data, ranging from 48 to 102. As shown in Fig. 6a, each sound signal was transformed into a cochleagram based on the Lyon's passive ear model[48] implemented with the Auditory Toolbox in MATLAB[62]. The cochleagram is represented as a matrix $P \in \mathbb{R}^{N_{\text{f}} \times L_{\text{data}}}$ where N_{f} is the number of frequency channels. It was converted to a masked data $P_{\text{mask}} = QP \in \mathbb{R}^{S_{\text{mask}} \times L_{\text{data}}}$, where $Q \in \mathbb{R}^{S_{\text{mask}} \times N_{\text{f}}}$ represents a binary mask of elements 0 and 1. We set $N_{\text{f}} = 78$. Each of the L_{data} column-wise sequences of length $L_{\text{in}}(= S_{\text{mask}})$ was converted to a voltage signal by scaling and interpolation, and then fed into the memristive reservoir.

Data availability

The data that support the findings of this study are available from the authors upon reasonable request. The ECG200 dataset is available from the UCR Timeseries Classification Archive (https://www.cs.ucr.edu/~eamonn/time_series_data_2018/). The TI-46 word corpus is available from the Linguistic Data Consortium (https://catalog.ldc.upenn.edu/LDC93S9).

Code availability

A MATLAB code for simulating memristor networks and reproducing some results will be made publicly available upon publication in the following link: [https://github.com/GTANAKA-LAB/Memristor-Network-Reservoir].

References

1. Verstraeten, D., Schrauwen, B., d'Haene, M. & Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Netw.* **20**, 391–403 (2007).
2. Schrauwen, B., Verstraeten, D. & Van Campenhout, J. An overview of reservoir computing: Theory, applications and implementations. In *Proceedings of the 15th European Symposium on Artificial Neural Networks*, 471–482 (2007).
3. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
4. Nakajima, K. & Fischer, I. *Reservoir Computing* (Springer, 2021).
5. Jaeger, H. The echo state approach to analysing and training recurrent neural networks-with an erratum note. German National Research Center for Information Technology GMD Technical Report, Bonn, Germany 148, 34 (2001).
6. Jaeger, H. & Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
7. Maass, W., Natschläger, T. & Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560 (2002).
8. Natschläger, T., Markram, H. & Maass, W. Computer models and analysis tools for neural microcircuits. *Neurosci. Databases* **20**, 123–138 (2003).
9. Lukoševičius, M. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade* 659–686 (Springer, 2012).
10. Tanaka, G. *et al.* Recent advances in physical reservoir computing: A review. *Neural Netw.* **115**, 100–123 (2019).
11. Vandoorne, K. *et al.* Toward optical signal processing using photonic reservoir computing. *Opt. Express* **16**, 11182–11192 (2008).
12. Dockendorf, K. P., Park, I., He, P., Príncipe, J. C. & DeMarse, T. B. Liquid state machines and cultured cortical networks: The separation property. *Biosystems* **95**, 90–97 (2009).
13. Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R. & Maass, W. Towards a theoretical foundation for morphological computation with compliant bodies. *Biol. Cybern.* **105**, 355–370 (2011).
14. Kulkarni, M. S. & Teuscher, C. Memristor-based reservoir computing. In *2012 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, 226–232 (2012).
15. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 468 (2011).
16. Soriano, M. C. *et al.* Delay-based reservoir computing: Noise effects in a combined analog and digital implementation. *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 388–393 (2014).
17. Larger, L. *et al.* High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. *Phys. Rev. X* **7**, 011015 (2017).
18. Dion, G., Mejaouri, S. & Sylvestre, J. Reservoir computing with a single delay-coupled non-linear mechanical oscillator. *J. Appl. Phys.* **124**, 152132 (2018).
19. Fernando, C. & Sojakka, S. Pattern recognition in a bucket. In *European Conference on Artificial Life*, 588–597 (Springer, 2003).
20. Nakane, R., Tanaka, G. & Hirose, A. Reservoir computing with spin waves excited in a garnet film. *IEEE Access* **6**, 4462–4469 (2018).
21. Nakane, R., Hirose, A. & Tanaka, G. Spin waves propagating through a stripe magnetic domain structure and their applications to reservoir computing. *Phys. Rev. Res.* **3**, 033243 (2021).
22. Chua, L. Memristor-the missing circuit element. *IEEE Trans. Circ. Theory* **18**, 507–519 (1971).
23. Chua, L. O. & Kang, S. M. Memristive devices and systems. *Proc. IEEE* **64**, 209–223 (1976).
24. Sillin, H. O. *et al.* A theoretical and experimental study of neuromorphic atomic switch networks for reservoir computing. *Nanotechnology* **24**, 384004 (2013).
25. Bürger, J., Goudarzi, A., Stefanovic, D. & Teuscher, C. Computational capacity and energy consumption of complex resistive switch networks. *AIMS Mater. Sci.* **2**, 530–545 (2015).
26. Lilak, S. *et al.* Spoken digit classification by in-materio reservoir computing with neuromorphic atomic switch networks. *Front. Nanotechnol.* **3**, 38 (2021).
27. Du, C. *et al.* Reservoir computing using dynamic memristors for temporal information processing. *Nat. Commun.* **8**, 2204 (2017).
28. Moon, J. *et al.* Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nat. Electron.* **2**, 480–487 (2019).
29. Zhong, Y. *et al.* Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing. *Nat. Commun.* **12**, 1–9 (2021).
30. Jang, Y. H. *et al.* Time-varying data processing with nonvolatile memristor-based temporal kernel. *Nat. Commun.* **12**, 1–9 (2021).
31. Paquot, Y. *et al.* Optoelectronic reservoir computing. *Sci. Rep.* **2**, 287 (2012).
32. Luz, E. J. D. S., Schwartz, W. R., Cámara-Chávez, G. & Menotti, D. Ecg-based heartbeat classification for arrhythmia detection: A survey. *Comput. Methods Programs Biomed.* **127**, 144–164 (2016).
33. Rodan, A. & Tiňo, P. Minimum complexity echo state network. *IEEE Trans. Neural Netw.* **22**, 131–144 (2011).
34. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* **453**, 80–83 (2008).
35. Joglekar, Y. N. & Wolf, S. J. The elusive memristor: Properties of basic electrical circuits. *Eur. J. Phys.* **30**, 661 (2009).
36. Kuzum, D., Yu, S. & Wong, H. P. Synaptic electronics: Materials, devices and applications. *Nanotechnology* **24**, 382001 (2013).
37. Bürger, J. & Teuscher, C. Variation-tolerant computing with memristive reservoirs. In *Proceedings of the 2013 IEEE/ACM International Symposium on Nanoscale Architectures*, 1–6 (IEEE Press, 2013).
38. Adam, G. C., Khiat, A. & Prodromakis, T. Challenges hindering memristive neuromorphic hardware from going mainstream. *Nat. Commun.* **9**, 1–4 (2018).
39. Fei, W., Yu, H., Zhang, W. & Yeo, K. S. Design exploration of hybrid CMOS and memristor circuit by new modified nodal analysis. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **20**, 1012–1025 (2012).

40. Ascher, U. M. & Petzold, L. R. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations* Vol. 61 (SIAM, 1998).
41. MATLAB. *(R2019b)* (The MathWorks Inc., 2019).
42. Takeda, S. *et al.* Photonic reservoir computing based on laser dynamics with external feedback. In *International Conference on Neural Information Processing*, 222–230 (Springer, 2016).
43. Torrejon, J. *et al.* Neuromorphic computing with nanoscale spintronic oscillators. *Nature* **547**, 428–431 (2017).
44. Tanaka, G. *et al.* Waveform classification by memristive reservoir computing. In *International Conference on Neural Information Processing*, 457–465 (Springer, 2017).
45. Pan, J. & Tompkins, W. J. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.* **20**, 230–236 (1985).
46. Alfaras, M., Soriano, M. C. & Ortín, S. A fast machine learning model for ECG-based heartbeat classification and arrhythmia detection. *Front. Phys.* **7**, 103 (2019).
47. Dau, H. A. *et al.* The UCR time series classification archive (2018). https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
48. Lyon, R. A computational model of filtering, detection, and compression in the cochlea. In *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7, 1282–1285 (IEEE, 1982).
49. Ma, Q., Zhuang, W., Shen, L. & Cottrell, G. W. Time series classification with echo memory networks. *Neural Netw.* **117**, 225–239 (2019).
50. Liberman, M. *et al.* TI 46-word (Linguistic Data Consortium, 1993).
51. Verstraeten, D., Schrauwen, B., Stroobandt, D. & Van Campenhout, J. Isolated word recognition with the liquid state machine: A case study. *Inf. Process. Lett.* **95**, 521–528 (2005).
52. Araujo, F. A. *et al.* Role of non-linear data processing on speech recognition task in the framework of reservoir computing. *Sci. Rep.* **10**, 1–11 (2020).
53. Zhao, M., Gao, B., Tang, J., Qian, H. & Wu, H. Reliability of analog resistive switching memory for neuromorphic computing. *Appl. Phys. Rev.* **7**, 011301 (2020).
54. Christensen, D. V. *et al.* 2022 roadmap on neuromorphic computing and engineering. *Neuromorph. Comput. Eng.* **2**, 022501 (2022).
55. Kantz, H. & Schreiber, T. *Nonlinear Time Series Analysis* Vol. 7 (Cambridge University Press, 2004).
56. Jaeger, H. *Short Term Memory in Echo State Networks* (GMD-Forschungszentrum Informationstechnik, 2001).
57. Legenstein, R. & Maass, W. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Netw.* **20**, 323–334 (2007).
58. Dambre, J., Verstraeten, D., Schrauwen, B. & Massar, S. Information processing capacity of dynamical systems. *Sci. Rep.* **2**, 1–7 (2012).
59. McDonald, N. R., Pino, R. E., Rozwood, P. J. & Wysocki, B. T. Analysis of dynamic linear and non-linear memristor device models for emerging neuromorphic computing hardware design. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, 1–5 (IEEE, 2010).
60. Olszewski, R. T. *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data* (Carnegie Mellon University, 2001).
61. Brookes, M. *et al.* Voicebox: Speech processing toolbox for MATLAB. www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.
62. Slaney, M. Auditory toolbox. Tech. Rep, Interval Research Corporation 10, 1194 (1998).

Acknowledgements

Author contributions

G.T. conceived this study. G.T. and R.N. devised the methods. G.T. conducted numerical experiments and analyses. All authors wrote and approved the manuscript.

Competing interests

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-13687-z.

Correspondence and requests for materials should be addressed to G.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.