

A Neuromorphic Model With Delay-Based Reservoir for Continuous Ventricular Heartbeat Detection

Xiangpeng Liang ¹, Student Member, IEEE, Haobo Li ², Member, IEEE, Aleksandra Vuckovic ³, John Mercer ⁴, and Hadi Heidari ⁵, Senior Member, IEEE

Abstract—There is a growing interest in neuromorphic hardware since it offers a more intuitive way to achieve bio-inspired algorithms. This paper presents a neuromorphic model for intelligently processing continuous electrocardiogram (ECG) signal. This model aims to develop a hardware-based signal processing model and avoid employing digitally intensive operations, such as signal segmentation and feature extraction, which are not desired in an analogue neuromorphic system. We apply delay-based reservoir computing as the information processing core, along with a novel training and labelling method. Different from the conventional ECG classification techniques, this computation model is a end-to-end dynamic system that mimics the real-time signal flow in neuromorphic hardware. The input is the raw ECG stream, while the amplitude of the output represents the risk factor of a ventricular ectopic heartbeat. The intrinsic memristive property of the reservoir empowers the system to retain the historical ECG information for high-dimensional mapping. This model was evaluated with the MIT-BIH database under the inter-patient paradigm and yields 81% sensitivity and 98% accuracy. Under this architecture, the minimum size of memory required in the inference process can be as low as 3.1 MegaByte(MB) because the majority of the computation takes place in the analogue domain. Such computational modelling boosts memory efficiency by simplifying the computing procedure and minimizing the required memory for future wearable devices.

Index Terms—Continuous ventricular heartbeat detection, delay-based reservoir computing, Memory efficient analogue computing, Physical neural network.

Manuscript received October 6, 2021; revised November 12, 2021; accepted November 16, 2021. Date of publication November 19, 2021; date of current version May 20, 2022. This work was supported by the UK EPSRC under Grant EP/W522168/1 - Industrial CASE Account. (Corresponding authors: Hadi Heidari.)

Xiangpeng Liang, Haobo Li, and Aleksandra Vuckovic are with the James Watt School of Engineering, University of Glasgow, U.K.

Hadi Heidari is with the James Watt School of Engineering, University of Glasgow, G12 8QQ Glasgow, U.K. (e-mail: hadi.heidari@glasgow.ac.uk).

John Mercer is with the BHF Cardiovascular Research Centre, University of Glasgow, U.K.

Digital Object Identifier 10.1109/TBME.2021.3129306

I. INTRODUCTION

CARDIOVASCULAR diseases are the major sources of global mortality, which led to 17.9 million deaths in 2016 (WHO) [1]. The electrocardiogram (ECG) is a clinical tool that records the electrical rhythm, rate and activity of the heart and provides a detailed analysis for the diagnosis and the treatment of abnormal heartbeats [2]. Early research focused on manual comparative ECG classification and diagnosis by the cardiologist. In recent decades, the rapid development of machine intelligence opens a novel pathway for automatic arrhythmias analysis and detection [2], [3]. In the future these systems have the potential to be integrated into remote patient devices to stratify clinical need by providing a more personalised healthcare system [2].

As a bio-inspired learning algorithm, neural network provides a computational architecture that mathematically models the electrical activities of a simplified neuron and synapse for information processing and intelligent applications [4]–[8]. However, running a neural network under a software-based system brings the entire information processing to a digital system where the data experiences storing, processing and communication using the bits of 1's and 0's. The software-based artificial intelligence (AI) faces the challenges in the further reduction of computational cost and miniaturization in post Moore's Law era [4], [8], [9]. To overcome this challenge, neuromorphic engineering paves a new way to develop a physical neural network to keep up with the computing needs [7], [10].

Delay-based reservoir computing (DRC) has been proposed as a candidate for physical implementation among all the topologies of neural network [11]. It is categorized under reservoir computing which is derived from Recurrent Neural Network (RNN). By introducing delay lines to form a delay-coupled reservoir, the DRC dramatically reduces the number of nonlinear neurons to one, which facilitates its hardware implementation using analogue and optical components for high-speed and low-power computing [12]–[18]. The DRC architecture is chosen as the main ECG processing core in this work because it is well-suited to process time-dependent signal, and feasible to implement as neuromorphic hardware. To validate the performance of our proposed DRC model, we applied this model to a Ventricular Ectopic Beat (VEB) detection task. Frequent VEB could be a sign for coronary heart disease, rheumatic heart disease and even acute myocardial infarction.

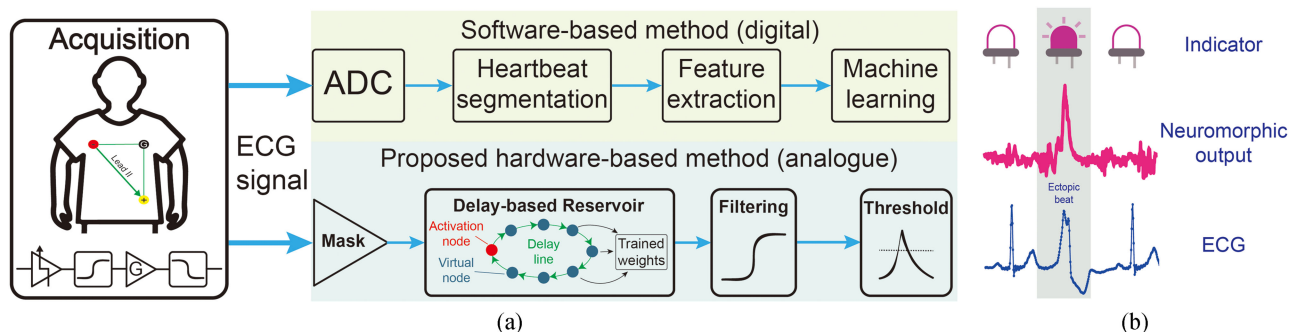


Fig. 1. (a) The comparison between the conventional software-based ECG classification and the proposed hardware-based method. The two standard procedures in literature, heartbeat segmentation and feature extraction were not adopted since they are difficult to achieve by analogue system. Instead, the DRC, along with its readout and pre-processing blocks is used as the information processing core. (b) The conceptual figure of the neuromorphic input ECG and output. The proposed hardware algorithm can receive continuous signal and perform point-by-point abnormal ECG detection. The output is an indication of the type of input ECG. A spike can be observed at the output when an ectopic ECG is received.

There are fundamental differences between the neuromorphic model with DRC architecture proposed in this paper and conventional automatic ECG detection. Conventional software-based ECG classification methods can be divided into five steps: 1) ECG signal pre-processing, 2) analogue-to-digital converter (ADC), 3) heartbeat segmentation, 4) feature extraction and 5) learning/classification [2]. These widely used procedures, such as detection of the QRS complex, signal segmentation and feature extraction, critically rely on digital operations which are not desirable in neuromorphic hardware [2], [4], [19]. For example, the prime step for the majority of automatic ECG classification algorithm is segmenting the entire ECG recording into individual heartbeat according to the detection algorithm of the QRS complex, followed by a feature extraction step where various features are obtained from each individual heartbeat to improve the classification accuracy. These operations are easy to achieve within a digital system by accessing the memory unit and running algorithms in the processor [4], [6], [9]. However, the digital operations and mass memory bring several constraints like latency, throughput and power, hindering the further development of computing performance in conventional architecture [20], [21]. Meanwhile, a prospective wearable device expects that the intelligent computing can also be carried out at the local edge [21]–[23]. Under such circumstances, a neuromorphic analogue processor based on the above-mentioned DRC architecture is well-suited to act as a direct interface to an ECG electrode with less memory requirement. Fig. 1(a) illustrates the block diagram of the proposed system in comparison with the conventional method. The proposed model preserves the hardware-achievable operations and deposes the procedures involving intensive digital components. Fig. 1(b) is a conceptual figure of the input and output of an analogue neuromorphic computer for ECG. Ideally, the indicator can be directly driven by the neuromorphic output that reflects the ECG type. The dynamic system aims to receive a continuous ECG stream and output the abnormal ECG diagnosis result. The differences mentioned above greatly reduce the memory needed to deploy such detection algorithm. Compared with the conventional software-based implementations, the proposed neuromorphic system is expected to offer advantages on energy efficiency and

computing power for machine learning workloads. The proposed model is validated by the MIT-BIH database [24]. The evaluation protocol follows the inter-patient paradigm which was presented by [25]. The result shows that this model obtained high accuracy and sensitivity and has a great potential to facilitate the future development of a pure analogue ECG processing system. The full development of the dynamic system is not covered in this paper and remains a topic of future research. The contribution of this paper is to provide the first fundamental analysis model of such a neuromorphic dynamic system using a DRC architecture specifically designed for ECG signal.

The remaining sections of the paper are organised as follows: Section II introduces the development of the model, including DRC, learning algorithm, memory capacity (MC) as well as a primary evaluation in terms of various parameters. Section III presents testing results using MIT-BIH database as well as a comparison between the proposed model and the state-of-the-art solutions. The discussion and conclusions are provided in Section IV.

II. DRC DESIGN FOR ECG

The development of proposed ECG processing model mainly includes four key aspects: 1) construction of the time-multiplexing reservoir, 2) Lasso regression and shifting labelling, 3) MC analysis, 4) post-processing. The simulation was carried out in MATLAB/Simulink software.

A. Database

The performance of proposed method is evaluated by the MIT-BIH arrhythmia database which is a well-known benchmark task recommended by Association for the Advancement of Medical Instrumentation (AAMI). The database includes 48 two-lead ECG recordings at 360 Hz sampling rate [26]. The AAMI suggested that the heartbeats in the database can be divided into five classes: normal, ventricular, supraventricular, fusion of normal and ventricular and unknown beats. The abnormalities for each type of ECG are clearly labelled in the data stream by at least two cardiologists. In this paper, the goal is to detect VEB type heartbeat which is highly correlated with

coronary heart disease and cardiomyopathy. Furthermore, this work follows the inter-patient paradigm suggesting that the data for training and testing should come from different patients. A significant study [25] suggested that the database can be divided into two groups: DS1 (101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230, where the numbers indicate the recording label) and DS2 (100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, 234). The DS1 is used to train the model while DS2 is for testing. The DS1 and DS2 split for inter-patient evaluation is in line with the use of MIT-BIH database in most ECG studies, which makes our results comparable with the state-of-the-art works. This paradigm is considered to be closer to a realistic scenario where the classifier can be directly used on the ECG signal from unknown patients without calibration. In addition, a single lead can satisfy the model requirement and the modified lead II (MLII) that placing electrodes on the chest was selected since it is an informative and commonly used configuration.

B. Delay-Based Reservoir Computing for ECG Processing

A conventional reservoir computing network consists of an input layer, a reservoir and an output layer. For a reservoir network with d -dimensional input, l -dimensional output and N neurons, only the coefficients between the output and reservoir ($W_{out} \in \mathbb{R}^{l \times N}$) need to be trained by a linear regression method, while the input coefficients ($W_{in} \in \mathbb{R}^{N \times d}$) and reservoir coefficients ($W_{res} \in \mathbb{R}^{N \times N}$) are randomly generated [27], [28]. The complex dynamic and nonlinear transformation in the reservoir would map the input data onto higher dimensional space for classification or prediction. With the internal feedback, the past neuron states can be preserved in the fading memory to affect the computation at the current state [29]–[31]. At each time step n , the states in a standard software-based reservoir are subjected to:

$$r(n) = f[\beta_1 W_{res} r(n-1) + \beta_2 W_{in} u(n)] \quad (1)$$

where $r(n)$ denotes the reservoir states with n^{th} input, $r(n)$ denotes the input at time step n , f represents the activation function, β_1 and β_2 are the feedback and input scaling factors, respectively. Under this dynamic, the interaction between neurons generates the high-dimensional recurrent states denoted by $r(n)$. As a special type of RNN, only the output layer of the reservoir needs to train by linear regression using collected $r(n)$ and target output. In previous studies, reservoir computing has exhibited interesting network properties and excellent performance in temporal signal processing[7]. Meanwhile, exploring reservoir's network typology is of high interest in both software and hardware engineering [27], [32]

In recent years, the reservoir has been developed that can be implemented by only one nonlinear neuron with time-multiplexing and a delayed feedback [11]. The randomly connected middle layer in traditional reservoir computing is replaced by a single neuron and virtual nodes created by a delay lines, namely DRC [11]. This substitution facilitates

the development of a physical reservoir, which is considered to be a candidate of the next-generation neuromorphic signal processors [7]. The 'Delay-based Reservoir' box in Fig. 1(a) briefly illustrates the network typology. In this paper, the DRC is designed to process ECG signal as illustrated in Fig. 2(a). Also, the processing core, a nonlinear dynamical neuron, is shown in Fig. 2(b). These modelling and design will be explained as follows:

1) Pre-Processing: Before sending the signal to the delay-loop, a pre-processing step is required to convert the raw ECG data to a shape specifically created for our DRC system. The filtering step follows the standard procedure: a 2^{nd} order Butterworth high-pass filter with a cut-off frequency at 0.5 Hz and a 12^{th} order finite impulse response filter with a cut-off frequency at 35 Hz, since the bandwidth in the range of 0.5-35 Hz contains most relevant ECG information [33]. Following the filtering, the ECG data was resampled by the sampling rate of 180 Hz to reduce the number of samples and accelerate the modelling.

2) Masked Signal: After pre-processing, a mask step is essential to create a reservoir dynamic in the activation node. Each data point of ECG signal should be multiplied by a binary matrix M (consisting of randomly uniform distribution of 1 and -1) with length equal to N , which is the number of virtual nodes in the reservoir [34], [35]. Assuming that τ is the sampling interval of the ECG signal, the time interval between every two points in the mask is defined as $\theta = \tau/N$ to facilitate the MC quantification in this paper, which will be discussed in the following section. Given the input ECG data is u and the masked data is $J(k)$, the masking algorithm can be described as:

$$J(k) = u \left(\left\lfloor \frac{k}{N} \right\rfloor \right) \cdot \frac{M(k \% N)}{s} + b, \quad for \ k = 1, 2, \dots, LN \quad (2)$$

where $\lfloor \cdot \rfloor$ is the floor function, L is the total length of input u , s and b denote the scaling and bias factors respectively for adjusting the input range according to the linear and nonlinear ranges of activation node, $\%$ is the modulus operation. Afterwards, the resulting $J(k)$ is sampled and held to generate a continuous signal $J(t)$. The signal before and after masking are plotted in Fig. 2(c). In this paper, $\tau = 5.56 \text{ ms}$ is the reciprocal of the sampling rate of 180 Hz. Also, we set the network size $N = 400$ and therefore $\theta = \tau/N = 13.89 \mu\text{s}$. For the hardware implementation, the analogue masked signal can be obtained by periodically switching the signal between the original signal and its negative counterpart according to the mask matrix M , which remains future development.

3) Delay-Coupled Activation Node: The $J(t)$ is sent to the activation node after pre-processing and masking. The design of the activation node is illustrated in Fig. 2(b). The circuit with bipolar junction transistor and resistors forms a Mackey-Glass nonlinear function for the input signal. A passive low pass filter is also connected to prevent the signal from rapidly reaching a plateau. Fig. 2(d) shows the step response of the circuit, the settling time T is obtained from the empirical configuration, where $T = 5 \times \theta$ [10], thus it equates to $69.44 \mu\text{s}$ in our case. This operation plays a crucial role in connecting a number of

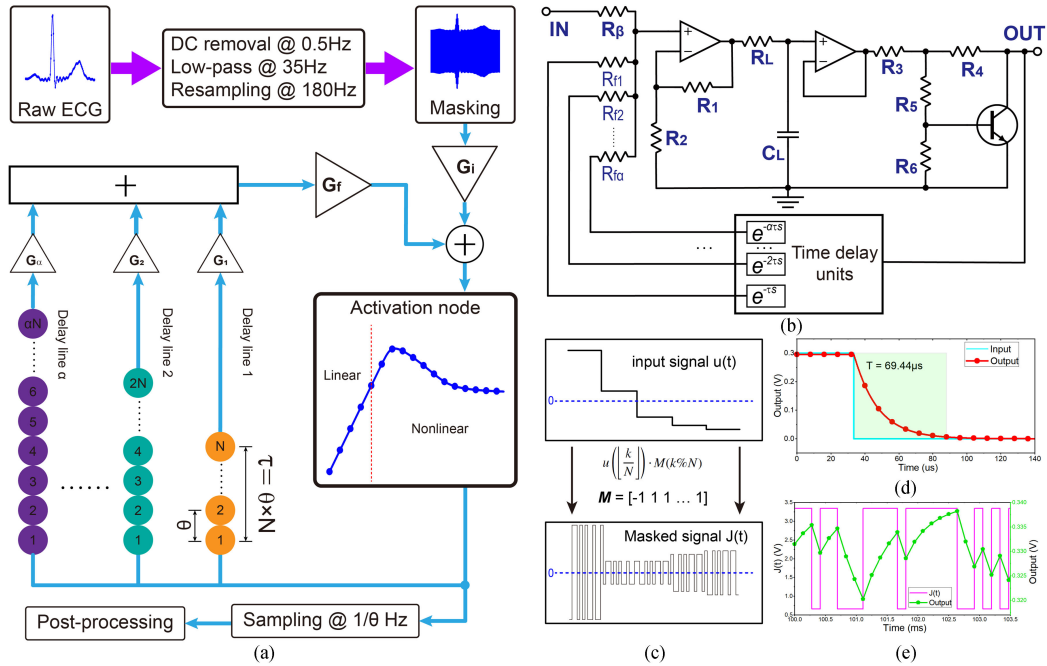


Fig. 2. (a) The system modelling of the DRC. After pre-processing and masking for the continuous raw ECG signal, the time-multiplexing signal will be fed into a activation node subjected to delayed feedback lines. Next, the result can be obtained through post-processing step. (b) The circuit design of the activation node and delay unit, which is the core of the reservoir computing unit. According to the working principle of DRC, the circuit should be able to exhibit nonlinearity (provided by bipolar junction transistor and $R_3 - R_6$) and integration (provided by R_L and C_L) properties. (c) An example of the masked signal generated by multiplying the original input with the mask matrix. (d) The settling period of the activation node allows each virtual node states in DRC to connect to historical states, which creates a similar dynamic in conventional reservoir by using fewer physical components. (e) A segment of input masked signal $J(t)$ and output signal of the DRC mode. The green dots are the node states sampled at $1/\theta$ Hz. The value of each green dot is related to historical several values, which implies the connection between the neighbouring virtual nodes.

neighbouring virtual nodes. The effect of this connection can be observed from the input and output in Fig. 2(e). The green dot line represents the discrete virtual node state Q and the pink signal is the masked input $J(t)$. Both of the sampling intervals and mask separation are θ . Given the settling time $T = 5 \times \theta$, what stands out in this figure is the correlation between current node state and its historical node states, resulting in the connections between neighbouring virtual nodes. Basically, such connections are provided by the integration property of activation node, and they can exhibits reservoir dynamic to map the temporal input onto high-dimensional node states. Furthermore, if T is smaller than θ , each node state will reach a plateau rapidly before the arrival of the next value. In this case, each state is irrelevant to historical states and the connections between nodes no longer exist.

To establish a recurrent connection, α delayed feedback lines are coupled to the output of the activation node. Each delay line delays the output for a certain time length and then feeding it back to the input with another scaling factor G_f . This delay-coupled reservoir dynamic is subject to a delay differential equation below:

$$\begin{aligned} \dot{q}(t) = & -q(t) + f(G_1 q(t - \tau), G_2 q(t - 2\tau), \\ & \dots G_\alpha q(t - \alpha\tau), J(t)) \end{aligned} \quad (3)$$

where $f(x)$ is the nonlinear function of the activation node (activation function) formed by the transistor and its surrounding

resistors, $q(t)$ denotes the node state at time t , $G_1, G_2, \dots, G_\alpha$ are the strengths of each delay line. The delay lines, which can be easily implemented by optical fiber in analogue domain [13], [18], [35], keep the information of historical data points within the loop as a fading memory. Differ from the software-based reservoir (1), the (3) describes the analogue signal in hardware reservoir over continuous time t . When a data point is sent to the delay-coupled node, the output contains not only the information of the current point, but also a certain portion of knowledge from historical inputs. In the absence of nonlinear function, the system can be considered as a positive feedback system and its transfer function is:

$$Q(s) = \frac{1}{1 - \frac{1}{R_L C_L s + 1} D(s)} J(s) \quad (4)$$

$$D(s) = \sum_{n=1}^{\alpha} G_n e^{-n\tau s} = G_1 e^{-\tau s} + \dots + G_\alpha e^{-\alpha\tau s} \quad (5)$$

where $D(s)$ approximates the delay unit, R_L and C_L form the time constant T of the activation node as shown in Fig. 2(b) and (d), $Q(s)$ and $J(s)$ denotes state output and masked input in Laplace domain, respectively. In this work, (4) was solved in MATLAB/Simulink together with the nonlinear function extracted from the activation node to obtain the resulting $q(t)$. The configuration of this delay-coupled activation node determines the volume of historical information that can be preserved in the

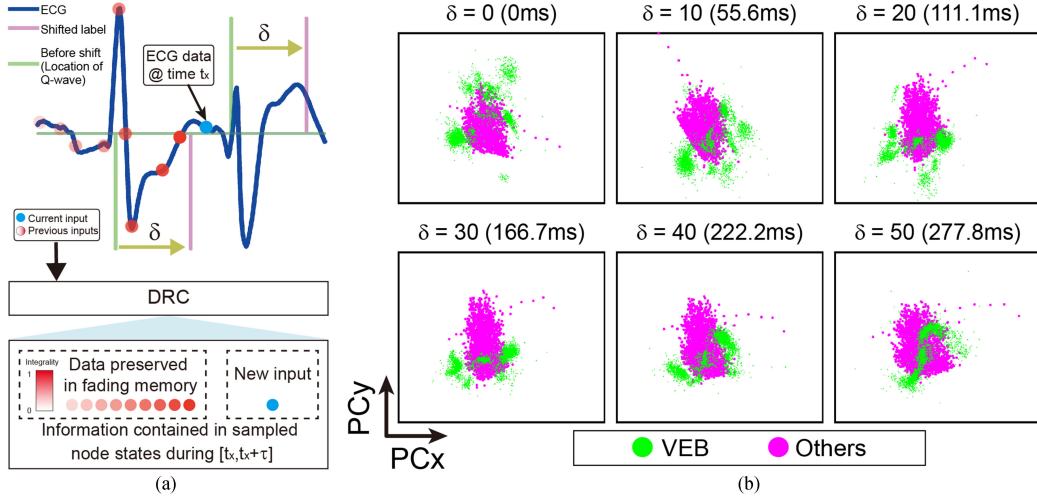


Fig. 3. (a) Schematic of the training data construction and the effect of fading memory. When the blue point is sent to the DRC, the information retained in the node state includes not only the current data (blue), but also the historical data (red). However, the historical data is not intact since the memory is fading. The top graph also illustrates the construction of training labels. The green line highlights the location of VEB (positive value) and other types of heartbeat (negative value). The pink line is a shifted version of green line. (b) Visualization of the node state distribution in high-dimensional feature space at δ points shifted away from the label location from database. PCA was used to reduce the state dimension from 400 to 2 for visualization.

loop, which is known as ‘memory capacity’. Furthermore, the fading memory is capable of preserving the information of the ECG morphology with only one input channel.

C. Lasso Regression

Sampling the continuous node state $q(t)$ obtained by solving (4), and then concatenating every N discrete elements as one matrix can obtain \mathbf{Q} , which is a high-dimensional mapping for the each input data $u(n)$ and the fading memory of historical data ($u(n-1)$, $u(n-2)$...)

$$\mathbf{Q} = \begin{bmatrix} Q(1) \\ Q(2) \\ \vdots \\ Q(n) \end{bmatrix} = \begin{bmatrix} q_1(1) & q_2(1) & \cdots & q_N(1) \\ q_1(2) & q_2(2) & \cdots & q_N(2) \\ \vdots & \vdots & \ddots & \vdots \\ q_1(n) & q_2(n) & \cdots & q_N(n) \end{bmatrix} \quad (6)$$

$$= \begin{bmatrix} q(\theta) & q(2\theta) & \cdots & q(N\theta) \\ q(\tau + \theta) & q(\tau + 2\theta) & \cdots & q(\tau + N\theta) \\ \vdots & \vdots & \ddots & \vdots \\ q(n\tau + \theta) & q(n\tau + 2\theta) & \cdots & q(n\tau + N\theta) \end{bmatrix}$$

where $q_i(j)$ denotes the state of i^{th} virtual node after j^{th} input, and $Q(j)$ is the state matrix for input $u(j)$. The elements in the last matrix ($q(\theta) \dots$) can be obtained by solving (4). Thus, output weight W_{out} of the DRC can be calculated through a linear regression of \mathbf{Q} and a desired output Y . Lasso regression is chosen as the regression method. Proposed by [36], Lasso provides a sparse linear regression by adding L_1 -norm regularisation to ordinary least squares regression for preventing overfitting. After sampling the node states of training data, the output weights can be obtained by minimising the loss function

of the Lasso regression:

$$W_{out} = \underset{W}{\operatorname{argmin}} \left\{ \sum_{i=1}^L (y_i - \sum_{j=1}^N q_i(j) W_j)^2 + \lambda \sum_{j=1}^N |W_j| \right\} \quad (7)$$

where L is the length of training data, y_i is the training label that will be discussed in the next section and λ represents the regularisation parameter that determines the strength of the L_1 penalty [36], [37]. Finally, the output X can be written as:

$$\mathbf{X} = \mathbf{Q}W_{out}, \text{ or } x(n) = Q(n)W_{out} \quad (8)$$

Different from L_2 -norm regularisation (Ridge regression), minimising the absolute value of coefficients will result in a sparse output connection by automatically eliminating redundant coefficients ($W_j = 0$), and that behaves like a coefficient selector. L_1 norm is advantageous and suitable for this application because sparse model reduces the number of components used to build a post-processing circuit in the next stage. In addition, the node states present a certain level of multicollinearity in our simulation. The Lasso regression is well-suited to minimise the effect of multicollinearity.

D. Training Data

The construction of training data and labels is shown in Fig. 3(a). The blue line is the training ECG data with the round dots representing each sample points. As mentioned above, the end-to-end setup receives the input data points one by one. The blue dot indicates the point injected to the system at the time stamp t_x while the red dots are the historical inputs at $t_x - \tau$, $t_x - 2\tau$ The node states $q(t)$ during $[t_x, t_x + \tau]$ contain high-dimensional information of not only the blue dot, but also of the historical inputs (red dots). This is because the information of previous points is preserved in the fading memory created by the delay-coupled loop. However, one data point

cannot be maintained permanently and will be attenuated after each cycle. This feature is crucial for constructing training data. The MIT-BIH database has labelled the location of Q-wave (the central spike for MLI lead) in each heartbeat. There are three steps for defining training label Y :

- The label $y(n)$ is based on the ECG type from the database and has the same length as input data. For the purpose of detecting VEB, the locations of VEB are set to 1, the locations of other types of ECG is set to -1 and the rest of the labels are 0 (the green line in Fig. 3)).
- Right shift the $y(n)$ by δ points (δ /180 ms). The label 1 and -1 are moved from the location of Q-wave to the end of a heartbeat (pink line in Fig. 3). The reason is that the node state $q(n)$ at the end of each heartbeat includes the fading memory of the entire heartbeat, which is analysed with MC in the next subsection.
- The number of VEB (1) and other types of heartbeat (-1) are unbalanced in terms of training labels, leading to the hyperplane that separates the two classes of data in high-dimensional space getting closer to the majority. Therefore, the desired output $y(n)$ should use $\frac{n_1+n_2}{n_1}$ and $-\frac{n_1+n_2}{n_2}$ instead of 1 and -1, where n_1 and n_2 are the number of VEB and other types of heartbeat respectively in the training data set.

Determining the shifting distance δ of labels should consider the state \mathbf{Q} distribution and dispersion between different classes. With the ECG input, the state $Q(n)$ implies the mapping of $(u(n), u(n-1), u(n-2), \dots)$ into a N -dimensional feature space. Generally, the fewer overlaps between different classes of ECG in the feature space, the better discrimination can be obtained by the regression. \mathbf{Q} over different δ is visualised and plotted in Fig. 3(b). At the first step, \mathbf{Q} was collected by feeding the ECG data to the DRC model. The green dots of each graph are the states at the locations of the shifted label which is δ points away from the Q-wave. For example, when $\delta = 0$, the states of green dot were collected at the location of Q-wave. In order to observe the dispersion between the extracted VEB states and normal ECG, the states of red dots were randomly collected within the range of other heartbeat classes including normal beat, left bundle branch block beat and right bundle branch block beat. After collection, the N -dimensional data of red dots are consistent for all six graphs. Here total 20395 heartbeat data points, including 15% VEB (green), were illustrated for each graph. For visualising the high-dimensional data, Principal Component Analysis (PCA) was used to map the data into two dimensions. As shown in Fig. 3(b), the two axes represent the first and second Principal Component (PC). In each graph, the result of PCA also shows that the first two PCs can explain 88.19% (std 1.87%) on average of all variances. When $\delta = 40$, the two classes of data points present less overlap. Therefore, the label was right-shifted for 40 samples (222.2 ms) for training.

E. Memory Capacity

Fig. 3(a) shows that the DRC can retain historical inputs in the fading memory. MC is a key criterion that indicates the volume of information from the previous inputs that can be retained

in the network. MC depends on the parameters and structure of the reservoir. The crucial variables that affect the MC are the nonlinearity of activation node, the strength of each delay line ($G_1, G_2 \dots G_\alpha$), the ratio between the strength of feedback and input ($\beta = G_f/G_i$) and the ratio between each delay line ($\gamma = G_2/G_1$) [38], [39]. Here, the nonlinear activation region leads to a rather low MC and lots of delay lines are needed to compensate for the loss of MC. Therefore, the linear region with two delay lines ($G_3 \dots G_\alpha = 0$) was chosen since our single input channel required high MC and too many delay lines are not optimal in hardware design. In order to keep the system stable, two requirements should be fulfilled: $G_f + G_i = 1$ and $G_1 + G_2 = 1$. Two tasks were employed to analyse the MC of the DRC model: 1) the task of binary sequence reconstruction is a standard method to quantify the MC; 2) a ‘look back’ ECG reconstruction task for validating the ECG morphology preserved in a fading memory.

1) Binary Sequence Reconstruction: A binary input $p(n) = -1$ or 1 was randomly generated to evaluate the MC of the proposed reservoir. In this task, the training output $y_i(n)$ is the matrix of $p(n)$ shifted by i steps for $i = 1, 2 \dots \infty$, which means that each state at the input $p(n)$ will be used to train and reconstruct the historical points of a square wave. The MC can be calculated as the sum of a linear correlation between reconstruction result $x_i(n)$ and the actual shifted sequence $y_i(n)$:

$$y_i(n) = p(n - i) \quad (9)$$

$$\begin{aligned} MC &= \sum_{i=1}^{\infty} m(i) = \sum_{i=1}^{\infty} \rho(x_i(n), y_i(n)) \\ &= \sum_{i=1}^{\infty} \frac{\langle y_i(n)x_i(n) \rangle_n}{\sigma^2(p(n))\sigma^2(x_i(n))} \end{aligned} \quad (10)$$

where ρ and σ denotes the correlation and variance, respectively [39], [40]. Theoretically, the summation of $m(i)$ should be taken from $i = 1$ to ∞ , which is incalculable. Thus, $i \in [1, 600]$ was adopted. The $p(n)$ was randomly generated with the length of 4000 (75% for training and 25% for testing) and consistent throughout the test. Also, the Ridge regression, instead of Lasso, was applied to the square wave reconstruction because a sparse W_{out} trained by Lasso may cause information loss. This means the result cannot accurately reconstruct the square wave and fully reflect the amount of information retained in node states. During training, the shifted binary sequence $y_i(n)$ was used as the label, which means the node state collected at the input $p(n)$ is employed and trained to reconstruct the input at i steps prior to $(p(n - i))$. This task tests the capacity of the DRC to retain the historical samples due to the fact that the memory is fading. Reconstructing the sample becomes increasingly demanding as i becomes higher. Therefore, the MC can be quantified by analysing the correlation between $x_i(n)$ and $y_i(n)$ in (10).

In this task, the two ratios, β and γ , are investigated. The MC is varied by different β , while γ was implemented to control the shape of $m(i)$. Fig. 4(a) shows a reconstructed sample for lower MC when $\beta = 9.1$ and $\gamma = 0.1$. When the shifting steps are small ($i=7$), the result (green dots) can capture most of the original points. As i increases, the reconstructed points depart

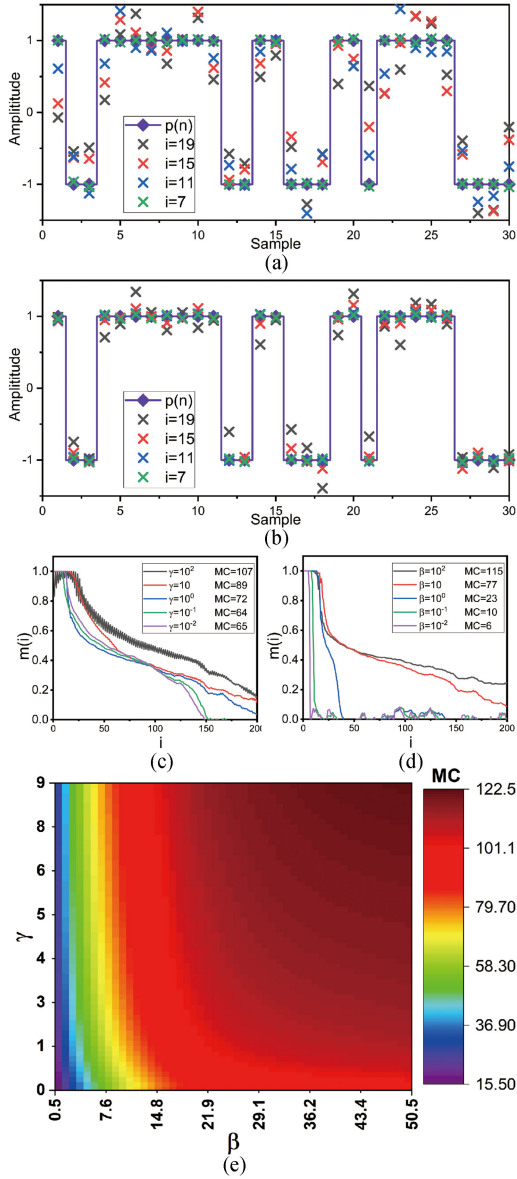


Fig. 4. Result of memory capacity testing using binary sequence reconstruction at (a) lower MC ($\beta = 9.1$ and $\gamma = 0.1$) and (b) higher MC ($\beta = 9.1$ and $\gamma = 10$). Based on the reconstruction results over different i , the correlation graph $m(i)$ can be computed. (c) The $m(i)$ curve with fixed β and (d) The $m(i)$ curve with fixed γ . The corresponding MC values are also provided. (e) The MC value as a function of the two ratios, β and γ .

from the targeted value with the decline of $m(i)$. In Fig. 4(b), raising γ to 10, which means that the percentage of 2nd delay line is increased, results in a small improvement in MC. As can be seen from the graph, most of the reconstructed points are close to the original data $p(n)$, whereas a small fluctuation was also found when $i = 19$. The effect of MC can be observed through the comparison of the two graphs. A higher MC will enhance the reservoir's ability to retain historical inputs in the current node state. Subsequently, a higher γ will improve the MC because the proportion of the 2nd delay line (length = 2τ) is increased. The $m(i)$ can be computed based on the correlation

between reconstructed data and original data over i , according to (10). Firstly, β is fixed and γ is varied from 0.01 to 100. The resulting $m(i)$ is plotted in Fig. 4(c). When $\gamma = 100$ (the 2nd delay line is highly dominant), the unbalanced distribution of $m(i)$ for odd and even i was found. The reason for this issue is that the 2nd delay line facilitates the reconstruction of even i step shifting. In addition, the rest of the lines show that the $m(i)$ is slightly reduced as γ declines, and the distribution of $m(i)$ can also be tuned by γ . Secondly, γ is fixed and β is varied. The Fig. 4(d) shows that β has a much stronger effect on MC compared to γ . In the next step, the MC can be quantified by taking the summation of the correlation $m(i)$, which is visualised in Fig. 4(e). In conclusion, the MC varies more significantly when β changes. The higher the β and γ , the stronger MC can be achieved. More specifically, MC mostly depends on the β , while γ can slightly change the distribution of $m(i)$.

2) 'Look Back' ECG Reconstruction: The 'look back' ECG reconstruction is a multistep backward signal reconstruction task. The purpose of this task is to numerically prove the fading amount of ECG information retained in the delayed feedback loop. If the state matrix collected at the end of each heartbeat can reconstruct the entire past heartbeat episode, this state matrix should include a certain amount of historical ECG information and thus can be used to classify different heartbeat types. In this task, the node states at the end of each ECG beats were used for training to reconstruct the historical n_1 points. For example, the node state at time t_x in Fig. 3(a) that was extracted as one of the $Q_M(n)$ and $y_M(n)$ is a matrix of ECG data segments from $t_x - \tau n_1$ to t_x , where $Q_M(n)$ and the $y_M(n)$ are the node state and the training label for this task. In total, n_2 groups of node state and their corresponding ECG slots were collected to calculate the output weight W_M for training. During the testing, the node states at the end of a heartbeat were collected to reconstruct ECG using (8). Assuming that the t^E is an array recording the location of each heartbeat in an ECG dataset, the elements in $Q_M(n)$ and $y_M(n)$ can be written as:

$$Q_M = \begin{bmatrix} Q_M(1) \\ Q_M(2) \\ \vdots \\ Q_M(n_2) \end{bmatrix} = \begin{bmatrix} q(t_1^E + \theta) & q(t_1^E + 2\theta) & \cdots & q(t_1^E + N\theta) \\ q(t_2^E + \theta) & q(t_2^E + 2\theta) & \cdots & q(t_2^E + N\theta) \\ \vdots & \vdots & \ddots & \vdots \\ q(t_{n_2}^E + \theta) & q(t_{n_2}^E + 2\theta) & \cdots & q(t_{n_2}^E + N\theta) \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} y_M(1) \\ y_M(2) \\ \vdots \\ y_M(n_2) \end{bmatrix} = \begin{bmatrix} u(t_1^E) & u(t_1^E - \tau) & \cdots & u(t_1^E - n_1\tau) \\ u(t_2^E) & u(t_2^E - \tau) & \cdots & u(t_2^E - n_1\tau) \\ \vdots & \vdots & \ddots & \vdots \\ u(t_{n_2}^E) & u(t_{n_2}^E - \tau) & \cdots & u(t_{n_2}^E - n_1\tau) \end{bmatrix} \quad (12)$$

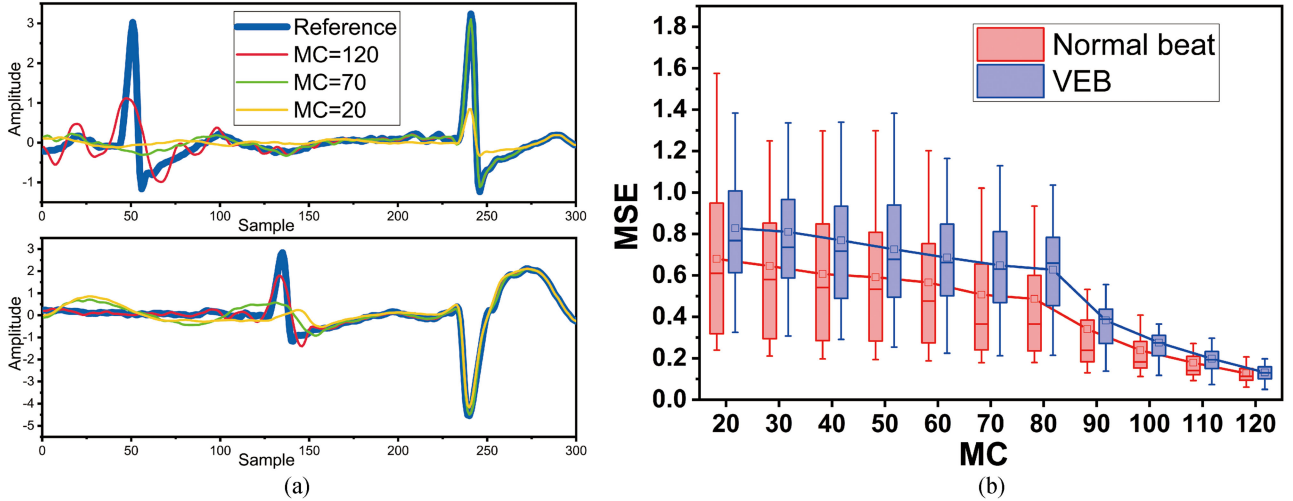


Fig. 5. (a) The result of the ‘look back’ ECG reconstruction task under different MC values. Under DRC models with different MC, the state matrix collected at the end of a heartbeat was used to reconstruct the past 400 ECG points (approximately two continuous heartbeats). The top graph shows the reconstruction of two normal beats and the bottom graph shows one VEB and one normal beat. The blue line is the reconstruction target. (b) The MSE between the reconstructed line and the reference data over MC value. The values of normal beat and VEB are plotted separately. The reconstruction results demonstrate the memristive property of the DRC model that preserving the historical information within the network.

where $n_1 = 400$ (~ 2.2 s) and $n_2 = 2000$ were used in this task. Using the \mathbf{Q}_M and \mathbf{Y}_M , the weights W_{out}^M and the reconstructed output can be calculated by ridge regression and (8). The reconstruction results at MC equal to 120, 70 and 20 along with the original waveform are plotted in Fig. 5(a). The upper graph illustrates the reconstruction of two normal beats and the lower one shows a ventricular beat followed by a normal beat. When MC = 120, the output can perfectly copy the later beat and tend to follow most of the earlier beat. In contrast, the low MC (20) can only retain the rough shape of the later beat and has almost no information about the earlier beat. This phenomenon can be quantified by taking Mean-Squared Error (MSE) between the reference and reconstructed data. A more comprehensive experimental result of ECG reconstruction using 1000 heartbeat episodes from the database is shown in Fig. 5(b). It reveals that when MC is increased, there is a continuous decrease in MSE as well as the interquartile range. Also, a normal beat is easier to reconstruct as compared to a ventricular beat. The result of this task further proves that the node state generated at the end of one ECG beat contains the information of the whole ECG under the DRC model reported in previous sections. The accuracy of reconstructing the ECG and the length of rebuildable data based on the MC value is verified in the last task. On the other hand, the earlier data are harder to retain in the network. Moreover, it is worth mentioning that, for up-sampled or analogue ECG signal, the value of $\theta \times N$ should not decrease as τ in order to maintain enough MC in time domain.

F. Post-Processing

After obtaining the output \mathbf{X} in (8), ideally, a spike would appear when a VEB is sent to the model, whereas the output should keep flat for other types of ECG. This is because we set a spike at the end of every VEB in the training shifted label. However,

fluctuation always happens throughout the output signal since some of the components for other types of ECG are similar to the VEB components. The Fig. 3(b) also demonstrates this issue as there always exists a certain amount of overlap. Thankfully, most of the spikes of VEB are higher than the unwanted noise. Therefore, a thresholding approach was adopted to capture the spikes of interest. Before the threshold, another two filters, which are the same as the high-pass and low-pass filters in the pre-processing step, were applied to the output $x(n)$ to support the thresholding. In this step, the location of the possible VEB is highlighted by spikes after thresholding.

III. PERFORMANCE MEASURES AND RESULTS

A. Performance Matrix for VEB Detection

The performance matrix for the VEB detection is recommended by AAMI. The metrics include sensitivity (Se), positive predictivity (PP), specificity (Sp), and accuracy (Acc), which are the standard statistic tools for evaluating the ECG classification on MIT-BIH database. In addition, F1 score, the harmonic mean of Se and PP is also used to optimise the parameters. These values can be calculated using the values of True Positive (TP), True negative (TN), False Positive (FP) and False Negative (FN) [26]:

$$Se = TP / (TP + FN) \quad (13)$$

$$PP = TP / (TP + FP) \quad (14)$$

$$Sp = TN / (TN + FP) \quad (15)$$

$$Acc = (TP + TN) / (TP + TN + FN + FP) \quad (16)$$

$$F1 = 2(Se \cdot PP) / (Se + PP) \quad (17)$$

The design goal is to minimise the FN and FP and maximise the TP and TN. It is worth mentioning that, in the absence of heart-beat segmentation, the performance quantification in this work

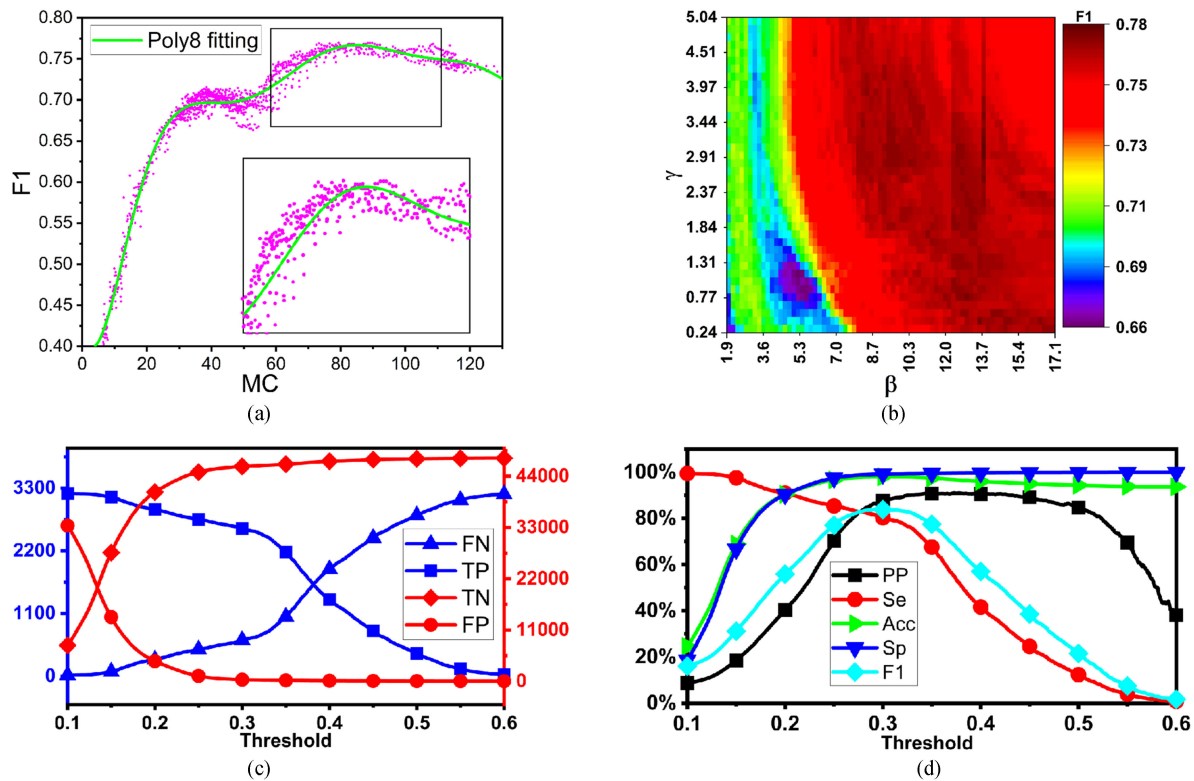


Fig. 6. Result of the parameter optimization and performance matrix. (a) The F1 score over MC. The green line stands for the curve fitting of the pink dots using an 8th order polynomial. The data was obtained from the simulation of MC over the two ratios and its resulting F1 scores. (b) The F1 score as a function of β and γ . The ranges of β and γ are the values producing MC from approximately 70 to 80, which is the range of the optimal F1 was computed in (a). The threshold evaluation in terms of (c) TP, TN, FP, FN and (d) performance matrix.

is slightly different from the beat-by-beat comparison reported in the literature. Instead, the model output is a point-by-point indication because the raw ECG signal is continuously fed into the model which acts like a dynamical nonlinear system. To count the number, a threshold was applied to the output $x(n)$ generated by ECG input. The threshold can highlight the spike of the output signal which may suggest the location of a VEB. If the location of one spike matches the VEB annotated in the database, this VEB will be counted as a TP. Similar approaches were also used to count TN, FP and FN.

B. Optimisation

The MC allows the model to load the ECG morphology to the recurrent network using only one-dimensional input without signal segmentation and feature extraction. However, a high MC will cause redundant information, such as previous multiple beats, preserved in the loop. Prior to calculating the final result, the F1 score was chosen as a standard to optimise the performance. In order to speed up the optimisation process, a descriptive dataset including 1138 normal beats and 420 VEBs randomly collected from DS1 was used to optimise the parameters. The F1 over different MC was firstly simulated. Fig. 6(a) provides the relationship between MC and F1. The green line presents a curve fitting using 8th order polynomial. The highest point in this simulation has been amplified and plotted. Before the maximum point, F1 keeps rising as MC increased. This is

because the gain of MC enhances the model's ability to keep the heartbeat information. Afterwards, the F1 is reduced with the growth of MC because the redundant information makes the VEB detection more challenging. The highest MC achieved by current double delay model is less than 130. The highest F1 occurs when MC is around 75. Based on this value and the results in Fig. 4(e), the β and γ generating MC approximately from 70 to 100 are evaluated by F1 (Fig. 6(b)). The highest F1 was obtained by $\beta = 13.8$ and $\gamma = 3.01$ in the condition that MC = 91.8.

The ratios discussed above had been used to deploy the model in which all ECG data in DS1 were tested. The output threshold versus the performance matrix are plotted in Fig. 6(c) and (d). At a low threshold value, the number of spikes including errors and noises were captured, resulting in a high FP and TP. In this case, unwanted noises were incorrectly detected as VEBs spikes. In contrast, a high threshold leads to the large number of missing VEBs (high FN and FP). Based on (13)–(17), the performance matrix is plotted in Fig. 6(d). The optimal F1 can be obtained when threshold is 0.3.

C. Results

The result of MC test and optimisation have been discussed in the previous sections. After optimisation, the entire testing dataset DS2 was fed to the model. In addition, four examples of the output signal together with the optimised threshold value,

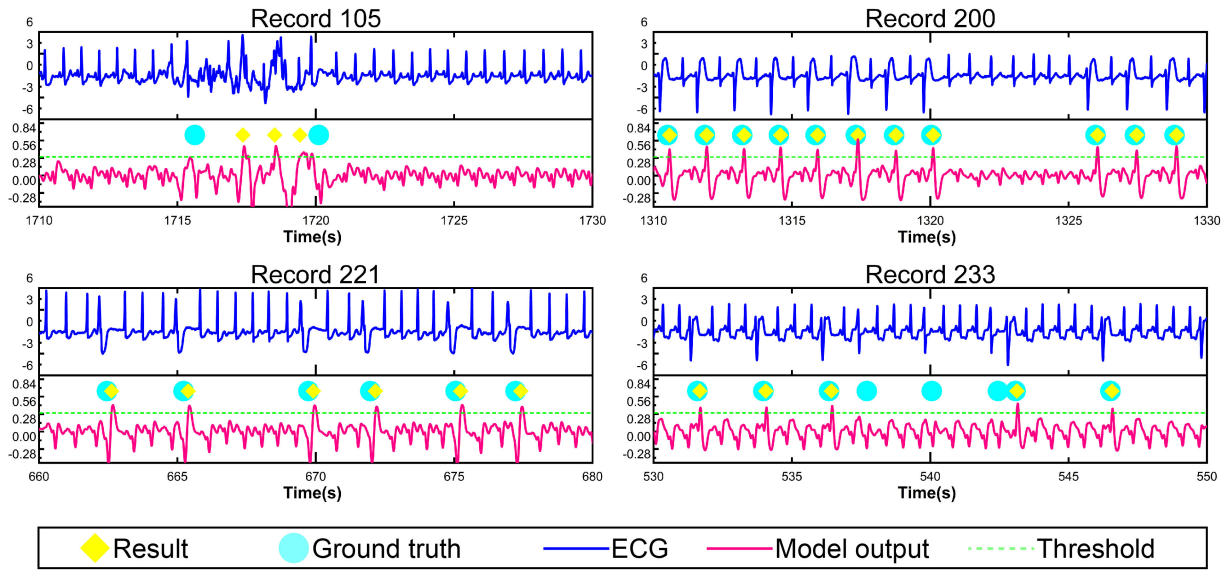


Fig. 7. Four episodes of the ECG and their processing output. The top row shows the input ECG signal. The bottom row shows the output of DRC model. The red dots denote the locations of VEB from database. The yellow rhombuses are the VEB detection results. For example, the artifact in Record 105 led to several FPs, and the multiform VEB in 233 resulted in FPs. Meanwhile, most VEB can be successfully detected in Record 200 and 221.

input signal and ground truth (the shifted labels of VEB in DS2) are shown in Fig. 7. One pair of matched dot and rhombus is counted as a TP, while FP is counted when no spike is found in the range of other types of heartbeat. As can be seen from the graph, most of the VEBs can be detected by the output spikes. At the same time, the values outside the VEB were kept low. In every TP, short displacements between the ground truth and result always existed because of the continuous point-by-point detection. However, the Record 233 contains the highest amount of multiform VEBs, which can only be partially detected due to its sharper waves. The unreadable artefact of the data such as few episodes in Record 105, which has also been reported in the database description, resulted in high TN and FN. The final performance matrix was calculated by the gross TP, TN, FP and FN, which is listed in Table I. The proposed hardware-based model yielded $Se = 80.9\%$, $PP = 87.5\%$, $Sp = 99.2\%$ and $Acc = 98.0\%$.

D. Minimum Memory Needed for Inference

In wearable devices, the trained VEB detection model can be deployed in a edge device where the hardware cost needs to be considered. Minimizing the memory that needs to be accessed for the detection is a crucial approach to reduce the hardware cost. While the state-of-the-art systems require massive memory for storing network parameters and performing nonlinear calculation, the memory needed for deploying the proposed model is significantly lower since the major proportion of nonlinear computing occurs in the analogue domain. The absence of signal segmentation and feature extraction reduces the memory requirement to zero for the data before the input player. Furthermore, the fixed physical reservoir layer significantly saves the number of parameters for the network activities. In the proposed

TABLE I
THE RESULT OF VEB DETECTION

Record	Se (%)	PP (%)	Sp (%)	Acc (%)	F1 (%)	# of VEB
100	100.0	33.3	99.9	99.9	50.0	1
103	-	0.0	99.8	99.8	-	0
105	12.2	4.1	95.4	94.1	6.1	41
111	100.0	8.3	99.5	99.5	15.4	1
113	-	-	100.0	100.0	-	0
117	-	0.0	99.7	99.7	-	0
121	100.0	4.2	98.8	98.8	8.0	1
123	100.0	50.0	99.8	99.8	66.7	3
200	91.8	100.0	100.0	97.4	95.7	826
202	5.3	14.3	99.7	98.9	7.7	19
210	39.7	96.3	99.9	95.4	56.2	194
212	-	0.0	99.7	99.7	-	0
213	96.8	74.5	97.3	97.3	84.2	220
214	60.9	100.0	100.0	95.5	75.7	256
219	82.8	98.1	100.0	99.5	89.8	64
221	97.7	100.0	100.0	99.6	98.9	396
222	-	0.0	97.2	97.2	-	0
228	98.1	96.2	99.2	99.0	97.1	362
231	100.0	28.6	99.8	99.8	44.4	2
232	-	0.0	98.8	98.8	-	0
233	71.1	99.7	99.9	90.8	83.0	831
234	0.0	0.0	99.9	99.8	-	3
Overall	80.9	87.5	99.2	98.0	84.0	3220

model, only the output of every time-multiplexing step needs to be stored and multiplied by the corresponding weight for the ‘multiply and accumulate’ operation. Thus, the total number of parameters in the model is 401 (400 weights and 1 network output at the current time step). Assuming that the data type is double precision floating point (8 bytes, as the simulation setup), the minimum size of memory for inference is 3.13 MegaByte(MB). In addition, under the proposed architecture, the memory is proportional to the number of virtual nodes. It means that the memory requirement would not exponentially increase

TABLE II
COMPARISON TABLE OF RECENT RESEARCHES USING INTRA-PATIENT PARADIGM AND MIT-BIH DATABASE

Work	Hardware-based?	Digital operation		Processing core	Resampling rate	Feature set	Effectiveness for VEB				Minimum memory per class (MB)
		Segmentation	Feature extraction				Se (%)	Acc (%)	Pp (%)	Sp (%)	
[41]	No	Yes	Yes	DBN	360	Intervals, morphology (with PCA)	85.8	97.9	-	98.8	>1522
[42]	No	Yes	Yes	PSO +SVM	360	Raw data, wavelets, TVCG	87.3	92.4	59.4	4.1	-
[43]	No	Yes	Yes	SVM	360	Intervals, SST, etc.	77.5	82.7	79.1	-	-
[33]	No	Yes	Yes	ESN	250	Intervals, std, average, etc	88.7	98.3	89.2	99.1	>8810
[44]	No	Yes	Yes	ESN	250	Raw ECG, RR-interval	92.7	-	95.7	-	>508
This work	Yes	No	No	DRC	180	Raw ECG	80.9	98	87.5	99.2	3.1

if we expend the network size for detecting more heartbeat types or higher performance matrix.

E. Comparison With the State-of-The-Art

The methodologies of automatic ECG classification have been widely explored in recent years. Using the open-access MIT-BIH arrhythmia databases [24], previous researches were done on automatically classifying different types of ECG by the intelligent algorithms such as support vector machine [42], [43], [45], echo state network [33], [44], decision tree [46], and neural network [3], [41]. As summarised in Table II, the selected publications are the state-of-the-art studies with the following criteria: 1) published in the past five years, 2) evaluated by MIT-BIH arrhythmia database, 3) inter-patient evaluation paradigm, 4) different types of processing core algorithms and 5) yielded good results. As can be seen from the table, the prior ECG signal processing studies mainly focused on software-based methods. The minimum size of memory was calculated using the method reported in the last section. The data type was double precision floating point unless a specific data type or fixed-point operation was used. For example, it has been reported that the ring ESN was designed to detect arrhythmia in [44]. First, the segmented heartbeat was 60 samples (240 ms). Second, total 63-dimensional vector including raw data and features were sent to the input channels. Next, given the reservoir size is 1000, the size of input weight is $63 \times 1000 = 63000$. In the reservoir layer, the ring ECG hugely reduces the number of weights to 1000 in comparison of 1000×1000 in normal ESN. Combining with another 1000 data points for storing node states, the total number of variables in this network is $60 + 63 + 63 \times 1000 + 1000 = 65123$, and the corresponding memory size divided by the number of heartbeat types under detection is 508.8 MB. This estimated number indicates the minimum number after fully optimizing the system, and the actual system should require larger memory. The Support Vector Machine (SVM)-based classifier involves complex operation using nonlinear kernel function to map the data or features to higher dimensional space. These operations will increase the burden of both processor and memory [42], [43]. The parameter reported is not sufficient to estimate the minimum memory. The algorithms with large network size such as deep

belief network (DBN) [42] and 1500-neuron ESN [33] demands high memory and intensive nonlinear computing, which may not be suitable to deploy in a wearable edge device.

Compared to the state-of-the-art automatic ECG processing study, an important difference is that the proposed model simulates a dynamic neuromorphic system receiving continuous ECG signal, rather than designing an offline machine learning framework. It is advantageous in a number of ways: 1) the two digital operations, signal segmentation and feature extraction, are removed from the processing model. This difference facilitates the implementation of a pure analogue neuromorphic processor; 2) it allows raw ECG signal from single lead flow into the model for detecting VEB and meanwhile obtains the acceptable result of effectiveness, which can be also considered as near-sensor computing; 3) it simulates a neuromorphic dynamic system, rather than a pure algorithm, which implies the processing and VEB detection happens in real-time; 4) the relatively lower sampling rate reduce the overall system frequency for real-time processing; 5) inherited from reservoir computing, the training of this model is relatively easy and fast. These advantages can be measured by the minimum memory size. The proposed system requires significantly lower memory, only 3.1 MB, for running the detection algorithm when the fully trained model is deployed.

IV. DISCUSSION AND CONCLUSIONS

In this paper, an ECG signal processing model based on DRC for hardware implementation has been proposed for the first time. This model was evaluated by an abnormal heartbeat detection task with the MIT-BIH arrhythmia database. Considering the notion of neuromorphic engineering, the model design refrained from using digital signal processing components. The novelty of the proposed model can be summarised as follows:

- i) Conventionally, heartbeat segmentation and feature extraction are two routines of automatic ECG classification in the previous studies. They were sidestepped in this model since these operations are much more friendly for software implementation rather than hardware. Instead, the method proposed in this paper is to analytically test the MC and to preserve the desired amount of ECG morphology in the recurrent loop, which has been tested

by two MC validation tasks. Accurate modelling of MC is crucial for designing the DRC model for processing a specific type of signal.

- ii) This model is an end-to-end dynamic system: the input is raw ECG signal and output is a point-by-point indication of signal class. As discussed in Table II, the performance matrix in this work is comparable to the software-based methods in the literature considering that the hardware-based method is not as flexible as a software-based method.
- iii) The main signal processing takes place in the analogue domain. It avoids suffering the intensive data transmission and processing in memory and processor, and therefore the memory needed for executing detection algorithm can be greatly reduced compared with previous studies. This advantage would potentially be useful to form a low-power neuromorphic computer [20], [21], [23].

Given the merits discussed above, the main weakness is the limited computing ability and task performance. In the absence of segmentation and feature extraction, the heartbeat information is preserved by the inherent memristive property of the network. These differences raise the task difficulty compared with the software-based algorithms where mass memory and accurate digital computing were used, resulting in a non-ideal performance. The higher computing performance could be obtained by scaling the network size in actual hardware, which remains a future challenge.

In conclusion, this model provides a fundamental analysis of using DRC as computing architecture for ECG processing. To the best of our knowledge, this work is the first hardware-based model acting like a dynamic system which can highlight VEB from continuous ECG input. The model and the analysis of its dynamic property (such as MC) will facilitate the future development of a neuromorphic wearable device, for instance, an analogue end-to-end abnormal ECG detector with built-in reservoir computing algorithm, to form a next-generation long-term ECG monitoring device at edge. Full development of such system remains a topic in future exploration. To achieve this, further efforts should involve: 1) developing pre- and post-processing front-end circuit; 2) analogue delayed feedback circuit; 3) power consumption evaluation and optimisation; 4) ASIC implementation.

REFERENCES

- [1] World Health Organization, "Cardiovascular diseases (CVDS)," 2017. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-cvds>
- [2] E. J. d. S. Luz *et al.*, "ECG-based heartbeat classification for arrhythmia detection: A survey," *Comput. Methods Programs Biomed.*, vol. 127, pp. 144–164, 2016.
- [3] M. M. A. Rahhal *et al.*, "Deep learning approach for active classification of electrocardiogram signals," *Inf. Sci.*, vol. 345, pp. 340–354, 2016.
- [4] J. Grollier, D. Querlioz, and M. D. Stiles, "Spintronic nanodevices for bioinspired computing," *Proc. IEEE*, vol. 104, no. 10, pp. 2024–2039, Oct. 2016.
- [5] X. Liang *et al.*, "Fusion of wearable and contactless sensors for intelligent gesture recognition," *Adv. Intell. Syst.*, vol. 1, no. 7, 2019, Art. no. 1900088.
- [6] C. D. Schuman *et al.*, "A survey of neuromorphic computing and neural networks in hardware," 2017, *arXiv:1705.06963*.
- [7] G. Tanaka *et al.*, "Recent advances in physical reservoir computing: A review," *Neural Netw.*, vol. 115, pp. 100–123, 2019.
- [8] M. R. Azghadi *et al.*, "Hardware implementation of deep network accelerators towards healthcare and biomedical applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 14, no. 6, pp. 1138–1159, Dec. 2020.
- [9] J. Torrejon *et al.*, "Neuromorphic computing with nanoscale spintronic oscillators," *Nature*, vol. 547, pp. 428–431, 2017.
- [10] M. C. Soriano *et al.*, "Delay-based reservoir computing: Noise effects in a combined analog and digital implementation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 388–393, Feb. 2015.
- [11] L. Appeltant *et al.*, "Information processing using a single dynamical node as complex system," *Nature Commun.*, vol. 2, no. 1, pp. 1–6, 2011.
- [12] D. Brunner *et al.*, "Parallel photonic information processing at gigabyte per second data rates using transient states," *Nature Commun.*, vol. 4, no. 1, pp. 1–7, 2013.
- [13] F. Duport *et al.*, "Fully analogue photonic reservoir computer," *Sci. Rep.*, vol. 6, no. 1, 2016, Art. no. 22381.
- [14] A. Hirose, R. Nakane, and G. Tanaka, "Keynote speech: Information processing hardware, physical reservoir computing and complex-valued neural networks," in *Proc. IEEE Int. Meeting Future Electron Devices*, 2019, pp. 19–24.
- [15] L. Larger *et al.*, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Phys. Rev. X*, vol. 7, no. 1, 2017, Art. no. 011015.
- [16] J. Moon *et al.*, "Temporal data classification and forecasting using a memristor-based reservoir computing system," *Nature Electron.*, vol. 2, no. 10, pp. 480–487, 2019.
- [17] Y. Zhong *et al.*, "Dynamic memristor-based reservoir computing for high-efficiency temporal signal processing," *Nature Commun.*, vol. 12, no. 1, pp. 1–9, 2021.
- [18] Y. Paquot *et al.*, "Optoelectronic reservoir computing," *Sci. Rep.*, vol. 2, pp. 1–6, 2012.
- [19] A. S. Sedra *et al.*, *Microelectronic Circuits*. New York, NY, USA: Oxford Univ. Press, 1998.
- [20] G. Indiveri and S. Liu, "Memory and information processing in neuromorphic systems," *Proc. IEEE*, vol. 103, no. 8, pp. 1379–1397, Aug. 2015.
- [21] J. D. Kendall *et al.*, "The building blocks of a brain-inspired computer," *Appl. Phys. Rev.*, vol. 7, no. 1, 2020, Art. no. 011305.
- [22] S. Bains, "The business of building brains," *Nature Electron.*, vol. 3, no. 7, pp. 348–351, 2020.
- [23] E. Covi *et al.*, "Adaptive extreme edge computing for wearable devices," *Front. Neurosci.*, vol. 15, 2021, Art. no. 611300.
- [24] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.
- [25] P. De Chazal, M. O'Dwyer, and R. B. Reilly, "Automatic classification of heartbeats using ECG morphology and heartbeat interval features," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 7, pp. 1196–1206, Jul. 2004.
- [26] ANSI/AAMI, "Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," in *ANSI/AAMI EC38*, vol. 1998, 1998.
- [27] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks—with an erratum note," German Nat. Res. Center Inf. Technol., Bonn, Germany, GMD Tech. Rep. 148, 2001.
- [28] H. Jaeger *et al.*, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.
- [29] F. Palumbo *et al.*, "Human activity recognition using multisensor data fusion based on reservoir computing," *J. Ambient Intell. Smart Environ.*, vol. 8, pp. 87–107, 2016.
- [30] Z. Tong *et al.*, "Reservoir computing with untrained convolutional neural networks for image recognition," in *Proc. Int. Conf. Pattern Recognit.*, 2018, pp. 1289–1294.
- [31] W. Wang *et al.*, "Wearable Wristworn gesture recognition using echo state network," in *Proc. IEEE Int. Conf. Electron. Circuits Syst.*, 2019, pp. 875–878.
- [32] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 131–144, Jan. 2011.
- [33] S. Ortín *et al.*, "Automated real-time method for ventricular heartbeat classification," *Comput. Methods Programs Biomed.*, vol. 169, pp. 1–8, 2019.
- [34] L. Appeltant *et al.*, "Constructing optimized binary masks for reservoir computing with delay systems," *Sci. Rep.*, vol. 4, no. 1, 2014, Art. no. 3629.

- [35] Y. Kuriki *et al.*, "Impact of input mask signals on delay-based photonic reservoir computing with semiconductor lasers," *Opt. Exp.*, vol. 26, no. 5, pp. 5777–5788, 2018.
- [36] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. Ser. B. (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.
- [37] V. Ceperic *et al.*, "Reducing complexity of echo state networks with sparse linear regression algorithms," in *Proc. UKSim-AMSS 16th Int. Conf. Comput. Modell. Simul.*, 2014, pp. 26–31.
- [38] M. Inubushi *et al.*, "Reservoir computing beyond memory-nonlinearity trade-off," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 10199.
- [39] S. Ortín *et al.*, "Reservoir computing with an ensemble of time-delay reservoirs," *Cogn. Comput.*, vol. 9, no. 3, pp. 327–336, 2017.
- [40] M. C. Soriano *et al.*, "Minimal approach to neuro-inspired information processing," *Front. Comput. Neurosci.*, vol. 9, 2015, Art. no. 68.
- [41] Z. Wu *et al.*, "A novel method for classification of ECG arrhythmias using deep belief networks," *Int. J. Comput. Intell. Appl.*, vol. 15, no. 04, 2016, Art. no. 1650021.
- [42] G. Garcia *et al.*, "Inter-patient ECG heartbeat classification with temporal VCG optimized by PSO," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 10543.
- [43] C. L. Herry *et al.*, "Heart beat classification from single-lead ECG using the synchrosqueezing transform," *Physiol. Meas.*, vol. 38, no. 2, pp. 171–187, 2017.
- [44] M. Alfaras *et al.*, "A fast machine learning model for ECG-based heartbeat classification and arrhythmia detection," *Front. Phys.*, vol. 7, 2019, Art. no. 103.
- [45] S. Raj *et al.*, "Cardiac arrhythmia beat classification using DOST and PSO tuned SVM," *Comput. Methods Programs Biomed.*, vol. 136, pp. 163–177, 2016.
- [46] R. Ghorbani Afkhami *et al.*, "Cardiac arrhythmia classification using statistical and mixture modeling features of ECG signals," *Pattern Recognit. Lett.*, vol. 70, pp. 45–51, 2016.