frontiers in Physiology

ORIGINAL RESEARCH
published: 27 September 2021
doi: 10.3389/fphys.2021.734178

Long-Time Prediction of Arrhythmic Cardiac Action Potentials Using Recurrent Neural Networks and Reservoir Computing

Shahrokh Shahi[1]*, *Christopher D. Marcotte*[1], *Conner J. Herndon*[2], *Flavio H. Fenton*[2], *Yohannes Shiferaw*[3] *and Elizabeth M. Cherry*[1]

[1] School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA, United States, [2] School of Physics, Georgia Institute of Technology, Atlanta, GA, United States, [3] Department of Physics & Astronomy, California State University, Northridge, CA, United States

The electrical signals triggering the heart's contraction are governed by non-linear processes that can produce complex irregular activity, especially during or preceding the onset of cardiac arrhythmias. Forecasts of cardiac voltage time series in such conditions could allow new opportunities for intervention and control but would require efficient computation of highly accurate predictions. Although machine-learning (ML) approaches hold promise for delivering such results, non-linear time-series forecasting poses significant challenges. In this manuscript, we study the performance of two recurrent neural network (RNN) approaches along with echo state networks (ESNs) from the reservoir computing (RC) paradigm in predicting cardiac voltage data in terms of accuracy, efficiency, and robustness. We show that these ML time-series prediction methods can forecast synthetic and experimental cardiac action potentials for at least 15–20 beats with a high degree of accuracy, with ESNs typically two orders of magnitude faster than RNN approaches for the same network size.

Keywords: reservoir computing, recurrent neural network, echo state network, time series forecasting, cardiac action potential

OPEN ACCESS

Edited by:
Hans Dierckx,
KU Leuven Kulak, Belgium

Reviewed by:
Edward Joseph Vigmond,
Université de Bordeaux, France
Kunichika Tsumoto,
Kanazawa Medical University, Japan

***Correspondence:**
Shahrokh Shahi
shahi@gatech.edu

Specialty section:
This article was submitted to Computational Physiology and Medicine, a section of the journal Frontiers in Physiology

Received: *30 June 2021*
Accepted: *27 August 2021*
Published: *27 September 2021*

Citation:
Shahi S, Marcotte CD, Herndon CJ, Fenton FH, Shiferaw Y and Cherry EM (2021) Long-Time Prediction of Arrhythmic Cardiac Action Potentials Using Recurrent Neural Networks and Reservoir Computing. Front. Physiol. 12:734178. doi: 10.3389/fphys.2021.734178

1. INTRODUCTION

Cardiac electrical signals, known as action potentials, exhibit complex non-linear dynamics, including period-doubling bifurcations in their duration (Guevara et al., 1984; Watanabe et al., 2001) and amplitude (Chen et al., 2017), along with higher-order period-doublings (Gizzi et al., 2013) and chaotic behavior (Chialvo et al., 1990). Potentially life-threatening states like fibrillation often are preceded by such long-short oscillations in action potential duration or amplitude known as alternans in the medical literature (Nolasco and Dahlen, 1968; Pastore et al., 1999; Gizzi et al., 2013; Chen et al., 2017). A number of methods for control of cardiac alternans have been developed (Rappel et al., 1999; Christini et al., 2006; Berger et al., 2007; Garzón et al., 2009; Garzon et al., 2014; Kulkarni et al., 2018), and while some have been demonstrated in cardiac experimental preparations (Christini et al., 2006; Kulkarni et al., 2018), they have not yet found clinical application in part because of the limited length scales over which control can be accomplished (Echebarria and Karma, 2002; Garzon et al., 2014; Otani, 2017). An alternative

strategy focusing on preventing rather than controlling alternans could be more attractive clinically, but such an approach would require accurate prediction of when such dynamics would occur.

Data-driven approaches can be used to forecast systems like cardiac action potentials by inferring the dynamics from observed data represented as time series (Kutz, 2013). Along with conventional techniques for time-series modeling and forecasting like autoregression approaches (Stock and Watson, 2001; Ing, 2003) and dynamic mode decomposition (Schmid, 2010), machine-learning methods have become increasingly used for predicting dynamical system states (Kutz, 2013; Chattopadhyay et al., 2020; Dubois et al., 2020). Recent years have seen significant advances in the field of machine learning, especially deep learning techniques. Recurrent neural networks (RNNs) have been successfully employed in dynamical domains, as the recurrent connections in the network provide a notion of memory and allow them to naturally embed temporal information. However, RNNs are still trained using the computationally expensive technique of back-propagation through time and remain prone to vanishing and exploding gradient problems. Gated RNNs can help overcome some of these problems; for example, to overcome the vanishing gradient problem, gated RNNs take advantage of memory cell architecture and a gating mechanism allowing the network to select which information should be kept and which forgotten (Hochreiter and Schmidhuber, 1997). This process enables the network to learn the long-term dependencies in sequential temporal data. Two widely used gated RNN approaches include long short-term memory (LSTM) networks and gated recurrent units (GRUs).

An alternative approach for modeling and predicting dynamical systems is reservoir computing (RC) (Lukoševičius and Jaeger, 2009; Sun et al., 2020), where, in contrast to other RNN architectures, the training remains limited to the output layer and the remaining parameters are selected randomly. Despite this simplification compared to other RNN architectures, RC techniques, including the commonly used echo state network (ESN) approach (Jaeger, 2002; Lukoševičius, 2012), have been used successfully to provide accurate multi-step-ahead predictions in non-linear and chaotic time series with very low computational costs (Bianchi et al., 2017; Han et al., 2021). Variations of ESNs, including clustered ESNs, where the reservoir consists of multiple sparsely connected sub-reservoirs (Deng and Zhang, 2006; Junior et al., 2020), and hybrid ESNs, which include input from a mathematical model and are a type of physics-informed machine learning technique (Oh, 2020; Willard et al., 2020), have been shown to have good performance in some cases (Pathak et al., 2018; Doan et al., 2019).

In this work, we show that it is possible to accurately predict future sequences of cardiac action potentials from complex voltage activity obtained *in silico* and in *ex-vivo* experiments. We further compare the performance of several machine-learning techniques for a multi-step prediction of complex cardiac action potential time series. In particular, we consider the accuracy and computational efficiency of LSTMs and GRUs along with ESNs, including a clustered architecture and a physics-informed hybrid option, for different network sizes.

2. METHODS

Below we provide a brief overview of machine-learning-based time series forecasting methods, describe the datasets we use, and give the details of our specific implementations.

2.1. Time Series Forecasting Methods

In this section, we provide a brief summary of the machine-learning approaches we use to forecast cardiac action potential time series.

2.1.1. Gated Recurrent Neural Networks

Recurrent neural networks (RNN) were introduced as a special class of neural networks in which the recurrent connections allow information to persist in the network. However, they suffer from vanishing and exploding gradient problems, which limit their ability to learn long-term dependencies in temporal sequences. Gated RNNs like long short-term memory networks (LSTMs) were developed to remedy such problems. These networks employ memory cells and a gating mechanism to address exactly these issues. **Supplementary Figure 1A** illustrates the information flow in an LSTM cell. In an LSTM network, a hidden state h_t is calculated using a map formalism:

$$\begin{aligned} i_t &= \sigma(\boldsymbol{W_i} x_t + \boldsymbol{U_i} h_{t-1} + b_i), \\ f_t &= \sigma(\boldsymbol{W_f} x_t + \boldsymbol{U_f} h_{t-1} + b_f), \\ o_t &= \sigma(\boldsymbol{W_o} x_t + \boldsymbol{U_o} h_{t-1} + b_o), \\ \tilde{c_t} &= \tanh(\boldsymbol{W_c} x_t + \boldsymbol{U_c} h_{t-1} + b_c), \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c_t}, \\ h_t &= \tanh(c_t) \odot o_t, \end{aligned} \tag{1}$$

where i_t, f_t, and o_t denote the input, forget, and output gates, at time t, respectively; x_t is the input vector; $\boldsymbol{W}$ and $\boldsymbol{U}$ are the weight matrices that along with biases b are adjusted during the learning process, c_t is the cell state (the internal memory of the LSTM unit), and $\tilde{c_t}$ is the cell input activation vector. In these equations, each σ function is sigmoidal and $\odot$ denotes Hadamard element-wise multiplication.

Gated recurrent units (GRUs) also were introduced to avoid vanishing and exploding gradient problems and share many similarities in architecture and performance with LSTM networks. The GRU memory cell can be considered as a simplification of an LSTM cell (see **Supplementary Figure 1B**). Compared to an LSTM memory cell, in a GRU unit, the input and forget gates are combined into a single update gate. This simplification considerably reduces the number of trainable weights and makes GRUs more computationally efficient; at the same time, the prediction does not experience a considerable deterioration in most cases and in some applications may even improve (Bianchi et al., 2017). The GRU equations are given by

$$\begin{aligned} z_t &= \sigma(\boldsymbol{W_z} x_t + \boldsymbol{U_z} h_{t-1} + b_z), \\ r_t &= \sigma(\boldsymbol{W_r} x_t + \boldsymbol{U_r} h_{t-1} + b_r), \\ \tilde{h_t} &= \tanh(\boldsymbol{W_h} x_t + \boldsymbol{U_h}(r_t \odot h_{t-1}) + b_h), \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t}, \end{aligned} \tag{2}$$

where z_t and r_t are update and reset gates, respectively, and $\tilde{h}_t$ is the candidate state. During the training process, the weight matrices $\boldsymbol{W}$ and $\boldsymbol{U}$ and the bias vector b are adjusted, thereby enabling the update and reset gates to select which information should be kept through time and which information is irrelevant for the problem and can be forgotten.

2.1.2. Echo State Networks

ESNs are a simple yet successful RNN architecture in which most of the network parameters are initialized randomly and remain untrained. **Supplementary Figures 2A,B** demonstrates the main components of an ESN. The hidden layer in an ESN is called the reservoir, which is a randomly initialized RNN.

The reservoir state h_t is updated according to

$$h_t = (1-\alpha)h_{t-1} + \alpha \tanh\left(\boldsymbol{W^{in}} x_t + \boldsymbol{W} h_{t-1}\right), \tag{3}$$

where $\boldsymbol{W^{in}}$ and $\boldsymbol{W}$ are the input weight and reservoir weight matrices, respectively; both are initialized randomly and remained untrained. We utilize an extension of the standard ESN formalism that includes a "leaky" update model, which explicitly includes a linear history term. The input signal is denoted by x_t and the constant parameter $\alpha \in [0,1]$ is known as the leaking rate. The output of the network is calculated by the following equation:

$$y_t = f^{out}\left(\boldsymbol{W^{out}}\left[x_t; h_t\right]\right), \tag{4}$$

where f^{out} is the output layer activation function, which is chosen here as a unity function. The output weights $\boldsymbol{W^{out}}$ are obtained here by regularized least-square regression with Tikhonov regularization to avoid overfitting.

Since the initial success of reservoir computing techniques and ESNs, a variety of network topologies have been proposed in the literature, including clustered reservoirs and deep ESNs. The main components of the network are similar to the baseline ESN architecture except for the reservoir topology; for clustered ESNs, the randomized connections between neurons form a set of sub-reservoirs sparsely connected to each other. The network topology is schematically illustrated in **Supplementary Figures 2C,D** and the update and training equations are the same as those for the baseline ESN (Equations 3 and 4).

We also consider a hybrid ESN approach, which is a physics-informed machine learning approach in which a knowledge-based model is integrated into an ESN; the model and ESN operate simultaneously during the training and prediction. The architecture of this approach is presented in **Supplementary Figure 2E**. For our application, the network in this design is driven with three input signals: $u_1(t)$, the pacing stimulus exciting the network at prescribed intervals; $u_2(t) = V_{\mathrm{KB}}(t)$, the knowledge-based model providing the voltage dynamics of a cardiac cell; and $u_3(t) = V(t)$, the synthetic or experimental voltage measurements. The knowledge-based model can be a much simpler (typically imperfect) model that provides an approximation of the dynamical behavior of the system, such as the two-variable Mitchell-Schaeffer (Mitchell and Schaeffer, 2003) or three-variable Fenton-Karma (Fenton and Karma, 1998) model, to increase the predictive ability of the network. Consequently, the time evolution of the reservoir state h_t is given by the same Equation 3, where the input signal vector is formed as follow,

$$x_t = \left[(u_1(t); u_2(t); u_3(t)\right]. \tag{5}$$

Here we use the Corrado-Niederer update of the Mitchell-Schaeffer model (Corrado and Niederer, 2016) with $\tau_{in} = 0.3$ ms, $\tau_{out} = 6$ ms, $\tau_{open} = 120$ ms, $\tau_{close} = 150$ ms, and $v_{gate} = 0.13$.

2.2. Datasets

To evaluate and compare the performance of these approaches in forecasting cardiac action potential time series, the methods are applied to two synthetic datasets derived from cardiac cell models and to an experimental dataset. We describe the three datasets used below.

2.2.1. Fenton-Karma Model-Derived Dataset

As one dataset, we use a time series of randomly timed action potentials generated using the Fenton-Karma (FK) model (Fenton and Karma, 1998), which includes a voltage variable and two gating variables. The model uses the Beeler-Reuter fitting of the FK model (parameter set 3 in Fenton et al., 2002) and is paced with cycle lengths drawn from a normal distribution using a 2-ms square stimulus current with magnitude 0.4 for 100 beats. To ensure a wide range of action potential durations, the cycle length distribution is centered at 320 ms with a standard deviation of 50 ms. The differential equations of the model are solved using the forward Euler method with a fixed time step of 0.1 ms; this time series is coarsened to obtain the synthetic voltage dataset (see section 2.3.4). The voltage data is then coupled with the stimulus timing so that a multivariate dataset is used, as explained in section 2.3.5.

Figure 1A shows the voltage trace that together with the corresponding stimulus input form the FK dataset; **Figure 1B** shows the corresponding action potential duration (APD) values. Data selected for training are shown in blue and testing data are shown in black. Just over 80 action potentials are used for training and about 20 for testing. Because the cycle lengths used include values both above and below the bifurcation to alternans, the resulting APDs included in the dataset span a range of about 200 ms. **Figure 1C** shows a blowup of the shaded regions in **Figures 1A,B** to illustrate the irregular timing of stimuli and variation in voltage responses within the training data.

2.2.2. Noble Model-Derived Dataset

As a different type of model-derived dataset, we use the four-variable Noble model (Noble, 1962) for a Purkinje cell in the absence of external pacing. To provide variation in action potential timing and duration, the Noble model is coupled to the three-variable Lorenz model (Lorenz, 1963) in the chaotic regime ($\rho = 28$, $b = 8/3$, and $\sigma = 10$). Time is effectively rescaled in the Lorenz system by multiplying each of the three differential equations by a factor of 0.001. The anionic current conductance in the Noble model was set to be proportional to the z variable of the Lorenz model, thereby driving oscillations

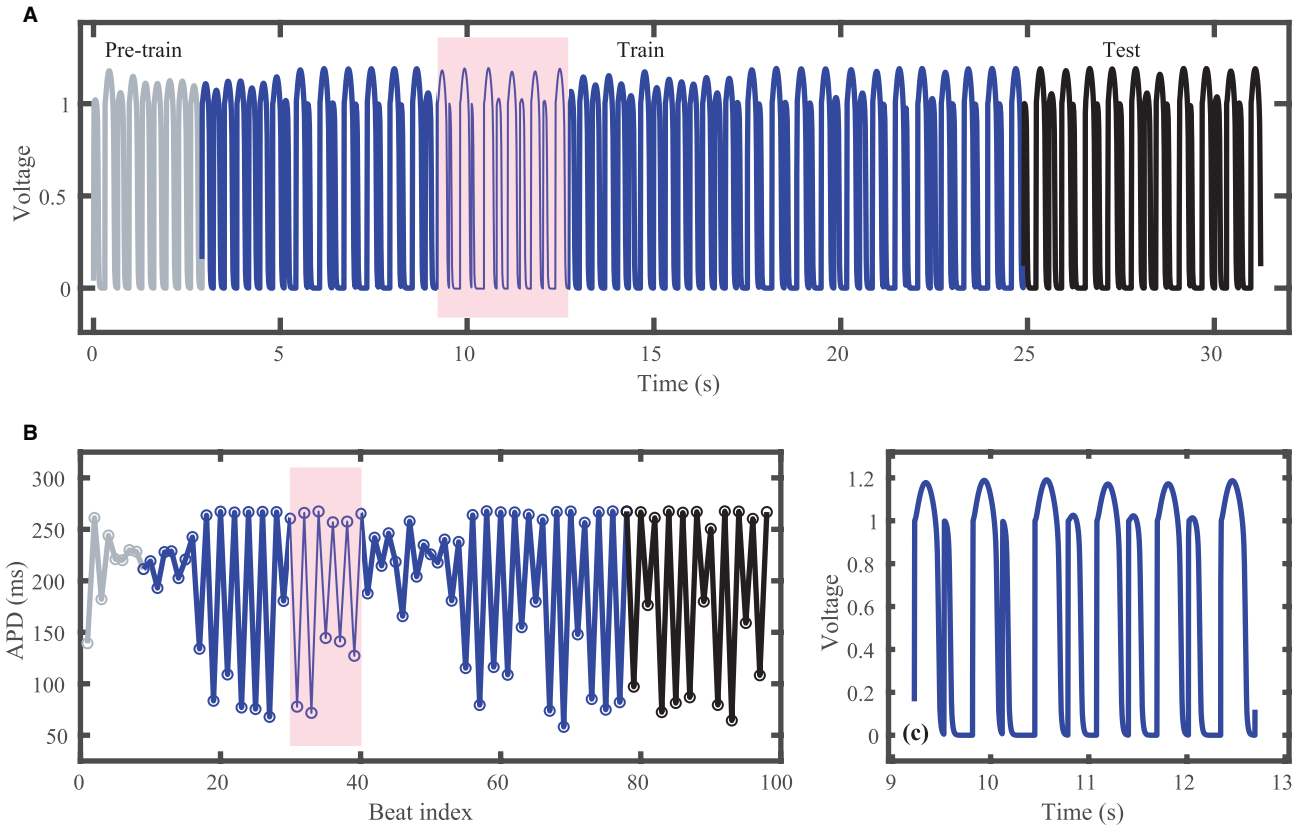

FIGURE 1 | Synthetic action potential time series generated by the Fenton-Karma (FK) model with randomly distributed cycle lengths. **(A)** Voltage time series including unused pre-training data (gray), training data (blue), and testing (prediction) data (black). **(B)** Corresponding APDs, following the same color scheme as **(A)**. **(C)** Zoomed in voltage trace corresponding to the shaded region in **(A,B)** showing the irregular action potential shapes and durations.

in the anionic current magnitude in concert with the Lorenz oscillations. Specifically, the conductance was set to 0 for $z = 0$ and to 0.2 for $z = 60$. This extension provides two important features of this dataset: first, the variation in cycle lengths is driven by a chaotic, rather than a random process, and second, there is no need for application of an external stimulus, as action potentials occur when the cell is quiescent and the Lorenz-driven current brings the voltage above the threshold for excitation. Therefore, in this case, only a univariate time series of voltage data is provided, with no external stimulus data. All other model parameters remain as specified in Noble (1962).

The Noble dataset voltage trace and action potentials are shown in **Figure 2**, with the training portion (around 65 action potentials) shown in blue and the testing portion (14 action potentials) in black. Because of the inclusion of the chaotic Lorenz model as a driving force, the Noble model-derived dataset demonstrates considerable variation in action potentials, with no consistent pattern. APDs vary between about 310 and 345 ms.

2.2.3. Experimental Dataset

The third dataset consists of irregular activity recorded from zebrafish hearts subjected to constant diastolic interval (DI) pacing (Cherry, 2017; Zlochiver et al., 2017); see **Figure 3**. All experimental procedures were approved by the office of Research Integrity Assurance of Georgia Tech under IACUC A100416. Zebrafish (*Danio rerio*) of either sex were anesthetized via cold water bath. Following anesthesia, hearts were quickly excised and immersed in Tyrode's solution (in mM: NaCl 124, KCl 4, $NaHCO_3$ 24, $NaH_2PO_4{\cdot}H_2O$ 0.9, $MgCl_2{\cdot}6H_2O$ 2, dextrose 5.5). Blebbistatin, used to stop contraction without major effects on electrophysiology (Fenton et al., 2008; Kappadan et al., 2020), was added to Tyrode's solution 20–30 min prior to data acquisition to help suppress heart motion. The heart was held in place by insect pins which attached the bulbus arteriosus to the bottom of a Sylgard-lined Petri dish. Stimulation was applied through AgCl bipolar electrodes placed on opposite sides of the heart close by to stimulate via electric field.

Intra- and extracellular voltages were acquired by two glass micropipettes containing 2.5 M KCl solution fastened into microelectrode holders (MEH3SFW, World Precision Instruments). Ag/AgCl half cells within the microelectrode holders sent signals to be buffered by pair of DC-powered preamplifiers (Electro 705, World Precision Instruments), which were connected together to output a differential measurement of transmembrane voltage. This transmembrane voltage was

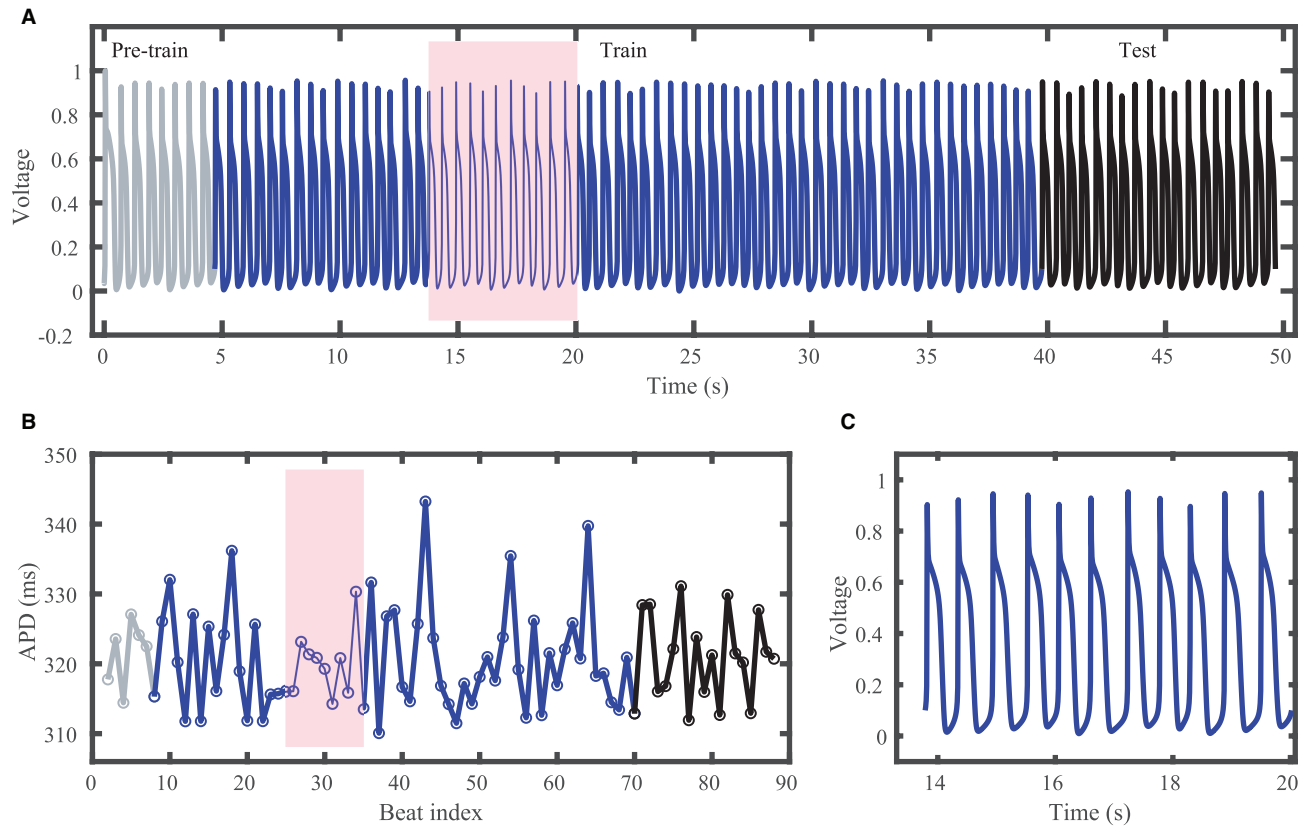

FIGURE 2 | Synthetic action potential time series generated by the Noble model driven by the chaotic Lorenz model. **(A)** Voltage time series including unused pre-training data (gray), training data (blue), and testing (prediction) data (black). **(B)** Corresponding APDs, following the same color scheme as **(A)**. **(C)** Zoomed in voltage trace corresponding to the shaded region in **(A,B)** showing the irregular action potential shapes and durations.

then split into two paths. One path led through a BNC breakout board (BNC-2110, National Instruments) to be read by a DAQ (PCIe-6341, National Instruments) and written to a file at 10,000 samples/s by a computer running a custom-built MATLAB script. Transmembrane voltage was also sent through a custom-built circuit that applied a gain and offset to the signals before being read by an Arduino Due. The Arduino Due then interpreted signals on-the-fly and determined when the heart should be stimulated to enforce a user-set DI, which was communicated to a current source stimulus isolator (Isostim A320, World Precision Instruments) that stimulated the heart.

Although constant-DI pacing can lead to stable and predictable APDs (Kulkarni et al., 2018), in our recordings APDs were highly variable despite the constant DI maintained. This high variability may result from the much smaller DIs used compared to the values used by Kulkarni et al. (2018), which were very close to the alternans bifurcation period. **Figure 3** shows the experimental dataset voltage trace and APDs, including over 100 training (blue) and over 20 testing (black) action potentials. Stimulus artifacts were removed using spline interpolation in a pre-processing step.

2.3. Implementation Details

All methods were implemented in MATLAB (R2020b) and were run on the same computer equipped with a 1.4 GHz Quad-Core Intel Core i5 processor and 8 GB of RAM, operating with macOS Big Sur (Version 11.4).

2.3.1. Hyperparameter Selection

The optimum values of various hyperparameters required for each method were tuned through an extensive grid search, the set of values for which are given in **Supplementary Table 1**. The ranges of the hyperparameter values used for the grid search and the number of values tested were chosen according to the results of initial experiments to generate reasonable results and also factor in the observed level of sensitivity of the employed approaches to each hyperparameter. Optimal hyperparameter values were obtained for each network size and dataset; see **Supplementary Tables 2–6**. Therefore, the results presented reflect the best attainable performance of each method for a given network size for each dataset.

2.3.2. Gated RNN Implementations

The LSTM and GRU networks are constructed using the MATLAB Deep Learning toolbox, where the network topologies

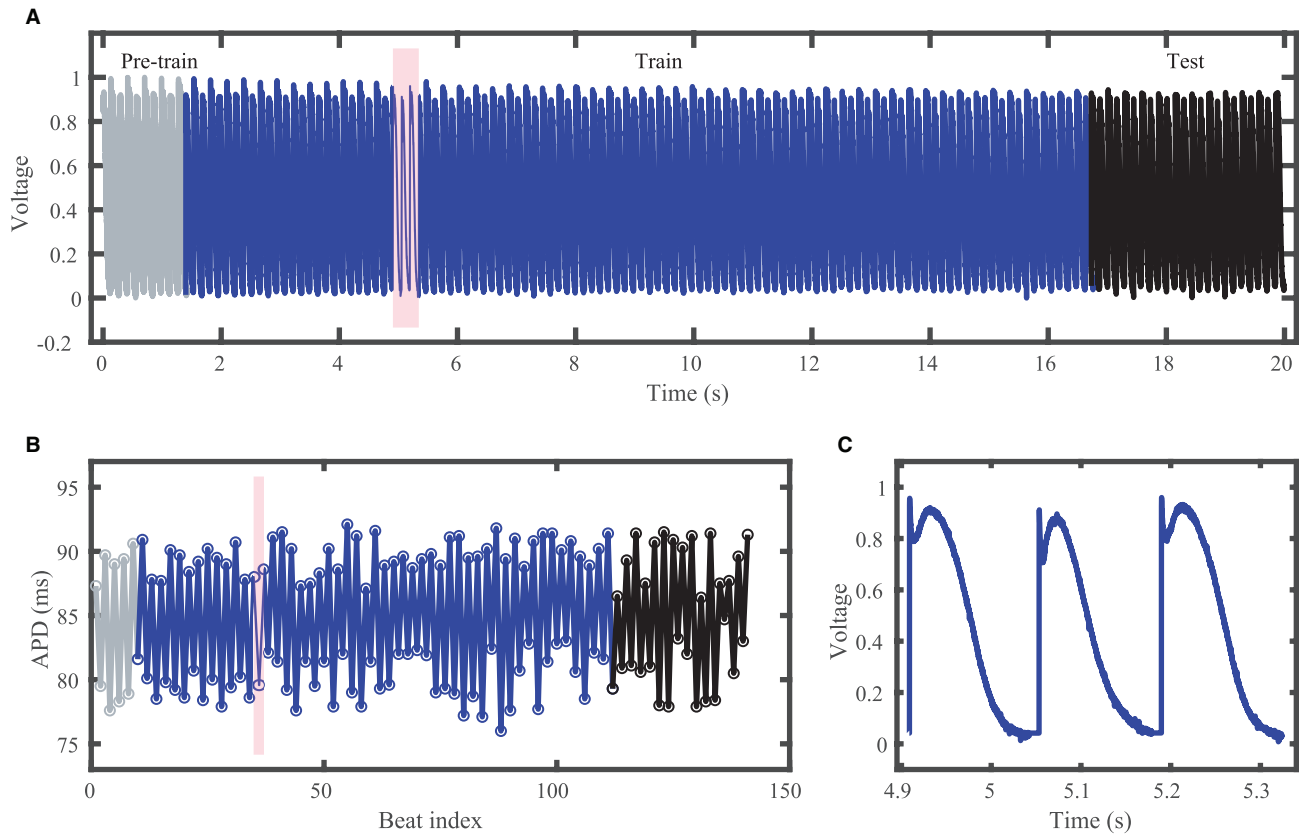

FIGURE 3 | Experimental zebrafish action potential time series featuring irregular alternans characterized by action potentials with irregular cycle lengths and durations. **(A)** Voltage time series including unused pre-training data (gray), training data (blue), and testing (prediction) data (black). **(B)** Corresponding APDs, following the same color scheme as **(A)**. **(C)** Zoomed in voltage trace corresponding to the shaded region in **(A,B)** showing the irregular action potential shapes and durations.

are specified by a graph of layers. To predict the action potential time series multiple steps into the future, a sequence-to-sequence regression LSTM architecture is employed that entails several main components. First, a sequential input layer is required to feed the input time series into the network. Then, an LSTM layer is used to learn the long-term dependencies between the time steps of the input sequential data. Finally, a fully connected layer connects the LSTM layer to a regression output layer to complete the design. The architecture of the GRU networks is the same as for the LSTM network except for employing a GRU layer instead of an LSTM layer. In addition to the single-layer architectures, multi-layer networks with multiple stacked gated layers are also tested in this work.

The main hyperparameters to configure in gated RNNs include the number of hidden layers and hidden units, the optimizer for the training network, and the hyperparameters related to the optimization solver, such as the maximum number of epochs, learning rate, learning rate drop factor, and regularization factor. Due to the high computational costs of gated RNNs, running an exhaustive grid search on all hyperparameters is not pragmatically feasible. Therefore, based on our initial experiments, some of these hyperparameters are set while the grid search determines the optimum values of those demonstrating a more significant role in the performance of the network. Accordingly, the Adam optimizer (Kingma and Ba, 2014) is employed for training the network with the MATLAB default training configurations and the maximum number of 30 epochs. Then, the grid search is employed to determine the optimum number of hidden layers and the initial learning rate (**Supplementary Tables 2**, **3**).

The trained network then can be used to predict the response of the system for the next time step. To forecast voltage values multiple steps ahead, a recursive approach is adopted in which at each time step, the response is predicted using the trained network and the network state is updated correspondingly. This predicted value is featured as the input for the next time step prediction. This procedure is repeated to predict the voltage response for the entire prediction horizon.

2.3.3. Echo State Network Implementations

The baseline ESN technique is implemented based on the original tutorial presented by Jaeger (2002) and the practical guide presented by Lukoševičius (2012). The reservoir graph is generated using the Erdős–Rényi algorithm (Bollobás, 2001), after which it is rescaled and updated to satisfy the echo state property of the network (Yildiz et al., 2012) ensuring that the

effect of initial conditions should vanish progressively and the reservoir state should asymptotically depend only on the input signals. The same procedure is conducted to construct the reservoirs in the clustered and hybrid ESNs. More specifically, in the hybrid ESN, the reservoir graph is generated with the same randomized approach, while in the clustered ESN, the sub-reservoir clusters are generated first, then connected to each other randomly, where an additional hyperparameter specifies the probability of the inter-cluster connections.

During the training of an ESN, the first initial steps of the network states are discarded to wash out the initial states and ensure the network dynamics are fully developed. Here, the first ten beats are considered as the transient phase and their corresponding state values are not used for training the networks.

Compared to the gated RNNs, the number of hyperparameters that play a more significant role in network performance in RC techniques is considerably higher, and the performance of the network highly depends on finding a good set of hyperparameters, including the number of neurons in the reservoir, connection probability used in the Erdős–Rényi graph generation step, reservoir spectral radius, input weight scale, leaking rate, and ridge regression regularization factor. Additionally, the number of clusters and the knowledge-based model are additional parameters to consider in clustered and hybrid ESNs, respectively. Although the size of the hyperparameter grid search space grows exponentially in RC techniques and is much higher compared to that of gated RNNs, because of the lower computational efforts required in ESN approaches, we obtained grid search results two times faster than for the gated RNNs.

Note that due to the random nature of ESNs and the intrinsic sensitivity of the network to the initial values of the parameters, the results for each network size are averaged over 10 experiments with different seed values for the random number generator.

2.3.4. Data Resampling

Although it is common to obtain data at a particular fixed time resolution, such a resolution often is not optimal; for example, it may contain so many points that it is difficult to obtain a good fit. More generally, an imbalanced distribution of data points in a time series can significantly deteriorate the performance of time-series prediction techniques. In such situations, a certain range of values are overrepresented compared to the rest of the time series, giving rise to a bias toward the values or behaviors that occur more frequently in the sequence. For example, in the case of an action potential time series, sampling that is uniform in time causes the upstroke phase, associated with the rapid depolarization of the cell membrane potential, to be underrepresented compared to the rest of the time series. Therefore, it is expected that prediction techniques may fail to correctly capture the upstroke phase in such cases and thus may produce a poor forecast overall. A common approach for tackling such issues is the use of resampling strategies (Moniz et al., 2017), which operate on the training dataset to make the distribution of the data points more balanced in terms of their information content.

In this work, we implement an under-sampling technique in which each data point is only included in the dataset if its voltage is sufficiently distinct from the last included data point, thereby ensuring more data points where the voltage changes rapidly. In this approach, two consecutive data points are considered distinct if the difference between the voltage values is greater than or equal to a threshold. If the threshold is set to zero, the dataset remains the same. In contrast, a very large threshold will result in great information loss and important features will be removed from the action potential time series. Therefore, the resampling threshold is also treated as a hyperparameter so that its optimum value is determined along with the other hyperparameters by the grid search. We also include a data point if the time since the last included data point exceeds a separate threshold, which is also treated as a hyperparameter, to ensure there is a sufficient density of points in portions of the action potential where the voltage changes slowly. **Supplementary Table 1** illustrates the possible values of the resampling thresholds used in the grid search. We found a significant increase in the predictive accuracy when using this resampling strategy and it is used for all results shown here.

2.3.5. Univariate vs. Multivariate Time Series Prediction

In practice, the action potential forecasting task entails predicting one variable (voltage) over time, resulting in a univariate time series. However, the input time series can be either a univariate or multivariate time series. The former occurs when the input data is assumed to be endogenous and is not driven by an external stimulus. The latter portrays cases in which cardiac cells are stimulated exogenously; in such a case, the pacing stimulus can also be introduced to the network along with the cardiac voltage signal. In this work, both scenarios are considered. Accordingly, the univariate input models are employed for forecasting the Noble dataset, where the auto-oscillatory nature of this model eliminates the requirements of applying an external stimulus. In contrast, both the FK dataset, which uses random stimulus timings, and the experimental dataset, which uses varying stimulus timings owing to the constant DIs but variable APDs, are used with multivariate time series prediction, which incorporates the pacing stimulus signal. In the case of the experimental data, the timing of applied stimuli is not directly available; thus, a pre-processing step is applied to detect the starting point of each beat in time and then a 2-ms stimulus current with a relative magnitude of 0.2 is used to generate the pacing stimulus signal. This process generates a stimulus current that is then resampled so that stimulus values are available for each resampled voltage data point. Our initial experiments demonstrate that the magnitude of the stimulus does not affect the quality of the predictions in this setup, but introducing the stimulus signal considerably improves the predictive ability in the first place.

Supplementary Figures 2A,C illustrate the architectures of the baseline and clustered ESNs, respectively, that are used for the univariate input case. To accommodate the pacing stimulus signal in multivariate input settings, these architectures are updated to include one more feature in the input layer (**Supplementary Figures 2B,D**). Introducing the stimulus

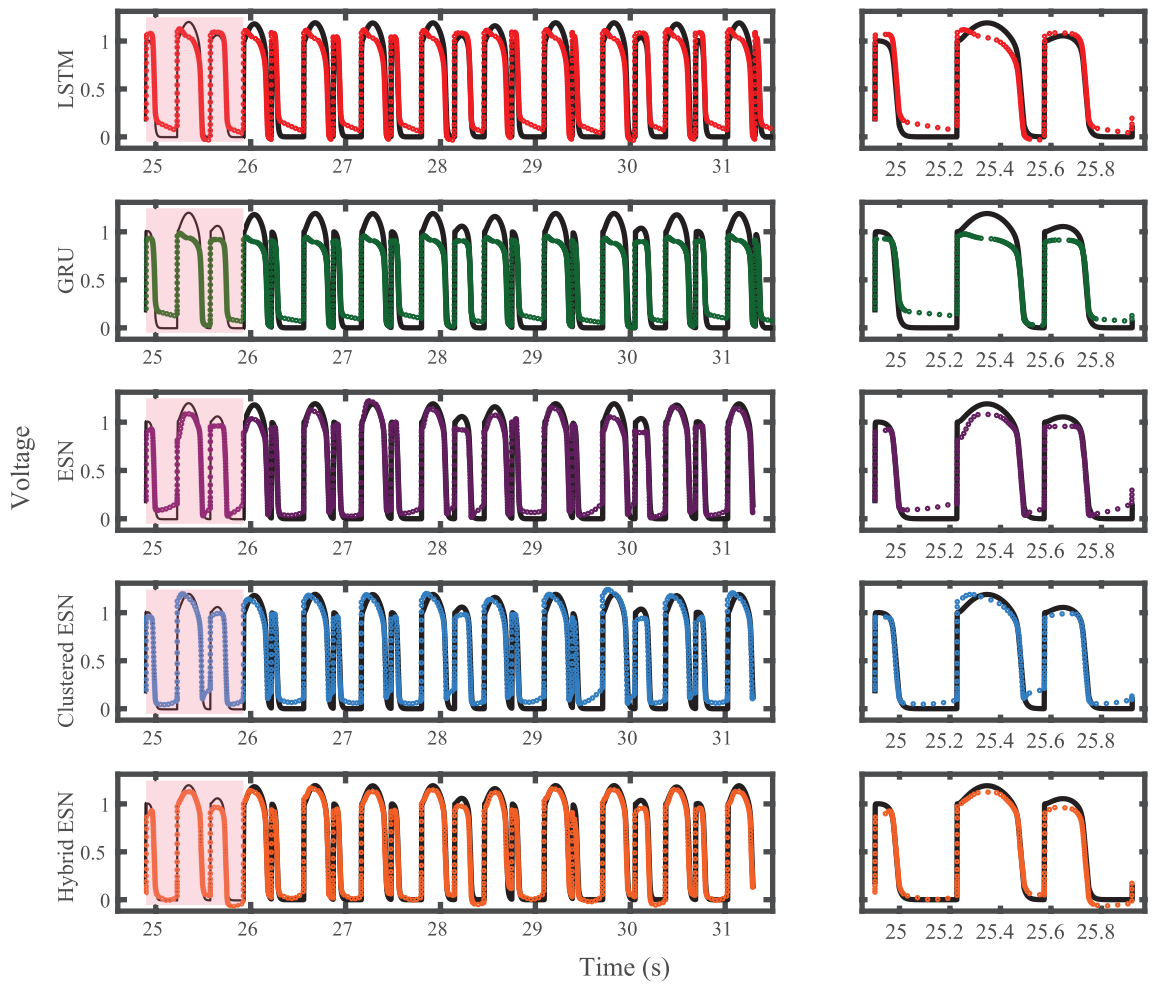

FIGURE 4 | FK dataset action potential prediction results obtained for the five methods using a fixed network size of 100 neurons. Test data are shown in black for reference and the predictions in color. **(Left)** All predicted APs. **(Right)** Zoomed view of the first three predicted APs.

information into the hybrid ESN approach is inevitable because in this architecture, the knowledge-based model should be synchronized with the input action potential time series to operate simultaneously. Therefore, the hybrid ESN is essentially developed for a multivariate input case (**Supplementary Figure 2E**). Accordingly, the pre-processing described step can be employed to extract the timing of the stimulus current. Similarly, in gated RNNs, the multivariate input case can be handled by adjusting the number of inputs in the sequential input layer.

2.3.6. Evaluation Metrics

To assess prediction accuracy, we use the root mean square error (RMSE) metric:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{V}_i - V_i\right)^2}. \quad (6)$$

where V_i and $\hat{V}_i$ are the target and predicted outputs, respectively, and n denotes the length of the test dataset. Note that the voltage values of the Noble and experimental datasets have been linearly rescaled to be between zero and one. The FK model is already scaled so that its upstroke reaches a maximum of one; no further rescaling is performed. As discussed in section 2.3.4, in all cases, the dataset values used here are not uniformly spaced in time.

We also assess error by comparing action potential durations (APDs). We define an APD as the time interval over which the voltage during an action potential is continuously larger than the threshold value, which is selected as 0.3 for the synthetic datasets (FK and noble) and 0.35 for the experimental dataset.

3. RESULTS

3.1. FK Model-Derived Dataset

The FK model-derived dataset (hereafter referred to as the FK dataset), shown previously in **Figure 1**, includes irregularity in action potential shapes and durations through the use of cycle

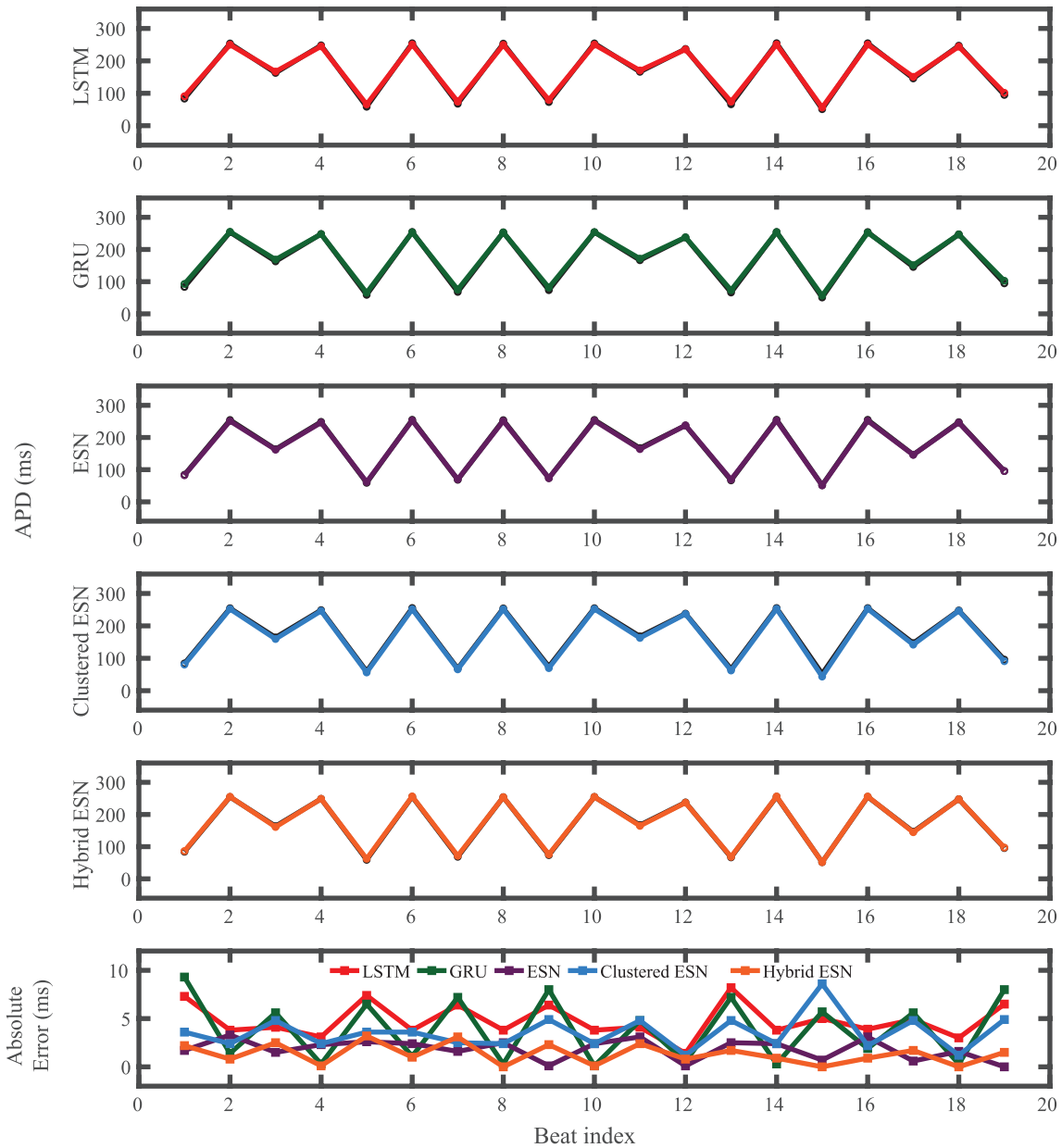

FIGURE 5 | FK dataset APD prediction results obtained for the five methods using a fixed network size of 100 neurons. APDs from data used for testing are shown in black for reference and predicted APDs are shown in color. Absolute error in APD prediction is shown in the bottom subplot, with color corresponding to prediction method.

lengths drawn from a normal distribution. **Figure 4** shows the 19 action potentials predicted by the five methods (LSTM, GRU, ESN, clustered ESN, and hybrid ESN) for a fixed network size of 100 hidden units. All five methods match the action potential upstrokes and downstrokes well. As a result, the predictions for APD by all methods have very low error (below 10 ms) across all 19 beats with no growth over time, as shown in detail in **Figure 5**, despite the irregular alternans present in the dataset. However, different methods exhibit different prediction accuracies for the plateau and rest phases of the action potentials. Specifically, the hybrid ESN does the best job of matching voltage values during these phases; the ESN approaches produce good results during the plateau but show depolarization preceding each action potential rather than remaining at a stable rest potential. The LSTM and GRU methods show the largest discrepancies, including plateau height mismatches and significant slowing in repolarization leading to elevated resting potentials.

Network size has a limited effect on overall accuracy as measured by RMSE. **Figure 6** shows that there is no clear trend in error as the network size is increased, except that the hybrid

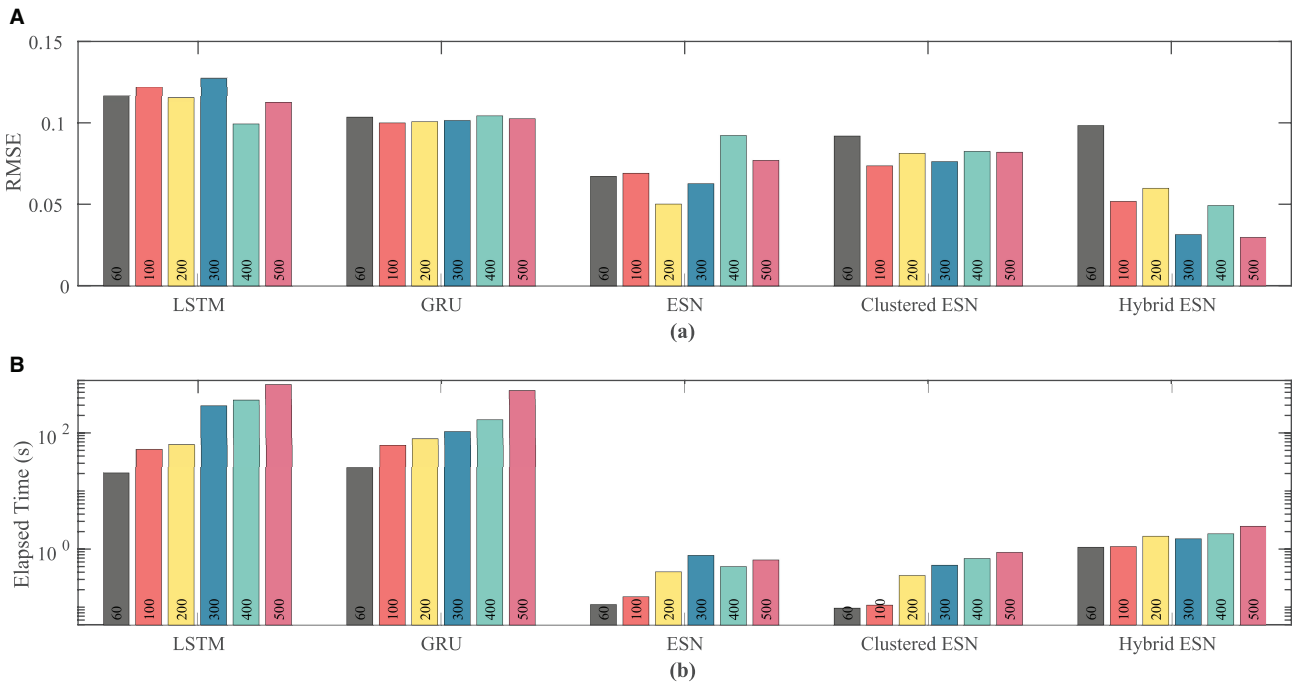

FIGURE 6 | Comparison of RMSE **(A)** and computational time **(B)** for each method and network size tested for the FK dataset.

ESN error is much lower for networks with at least 100 neurons. The hybrid ESN also has the lowest or near lowest error for all cases with at least 100 neurons, while the ESN and clustered ESN methods generally achieve slightly lower error than the LSTM and GRU methods.

The lower plot in **Figure 6** demonstrates that for a fixed network size the ESN and clustered ESNs accomplish the combination of training and prediction tasks much faster—by roughly two orders of magnitude—than the LSTM and GRU methods, with the hybrid ESN placing in between. These timing differences across the methods are maintained across all network sizes. However, within each individual method, the time for training and prediction increases with the number of neurons, except for the hybrid ESN method, for which such a trend is less clear.

3.2. Noble Model-Derived Dataset

The results of using the five methods to predict action potentials in the Noble dataset (shown in **Figure 2**) using a fixed network size of 100 can be seen in **Figure 7**, which shows the voltage traces, and in **Figure 8**, which shows the predicted APDs and absolute error in APD. The LSTM predictions generally achieve good agreement throughout the testing phase, with some discrepancies during the plateau and an overestimation of phase 4 depolarization. The GRU predictions are similar except that they repolarize less completely and consistently underestimate the peak upstroke voltage. The ESN and clustered ESN methods show improved accuracy with relatively minor discrepancies. In contrast, the hybrid ESN exhibits a very different action potential shape more in line with the capabilities of the knowledge-based model and consistently overestimates APD.

Figure 9 shows that the ESN and clustered ESNs achieve the lowest RMSE across different network sizes. The hybrid ESN and LSTM methods perform relatively well across most network sizes but produce larger RMSE values for some network sizes. GRUs have the highest error for most network sizes for this dataset. It is difficult to discern a clear trend in accuracy with increased network size; prediction method differences appear to have stronger effects.

As with the FK model, the RNN approaches consistently take longer than the ESN and clustered ESN approaches, with the hybrid ESN in between. For network sizes of at least 100 hidden units, the ESN and clustered ESN methods require about two orders of magnitude less computational time than the RNN methods. For all approaches, there is a modest increase in computational time for 100 or more hidden units as the network size increases, with the exception of the hybrid ESN, for which the computational time is approximately constant across all network sizes tested.

3.3. Experimental Dataset

For the experimental dataset, obtained from zebrafish paced using a constant-DI protocol and shown in **Figure 3**, all five methods are able to reconstruct most of the action potential features, as demonstrated in **Figure 10** with 100 hidden units. With each method, the discrepancies in predicted voltage values occur mostly during the portions of the action potentials with smaller voltage derivatives, the plateau and the rest phase. All methods underestimate the plateau height and fall short of repolarizing fully, with the hybrid ESN continuing to repolarize slowly throughout what should be the rest phase while the

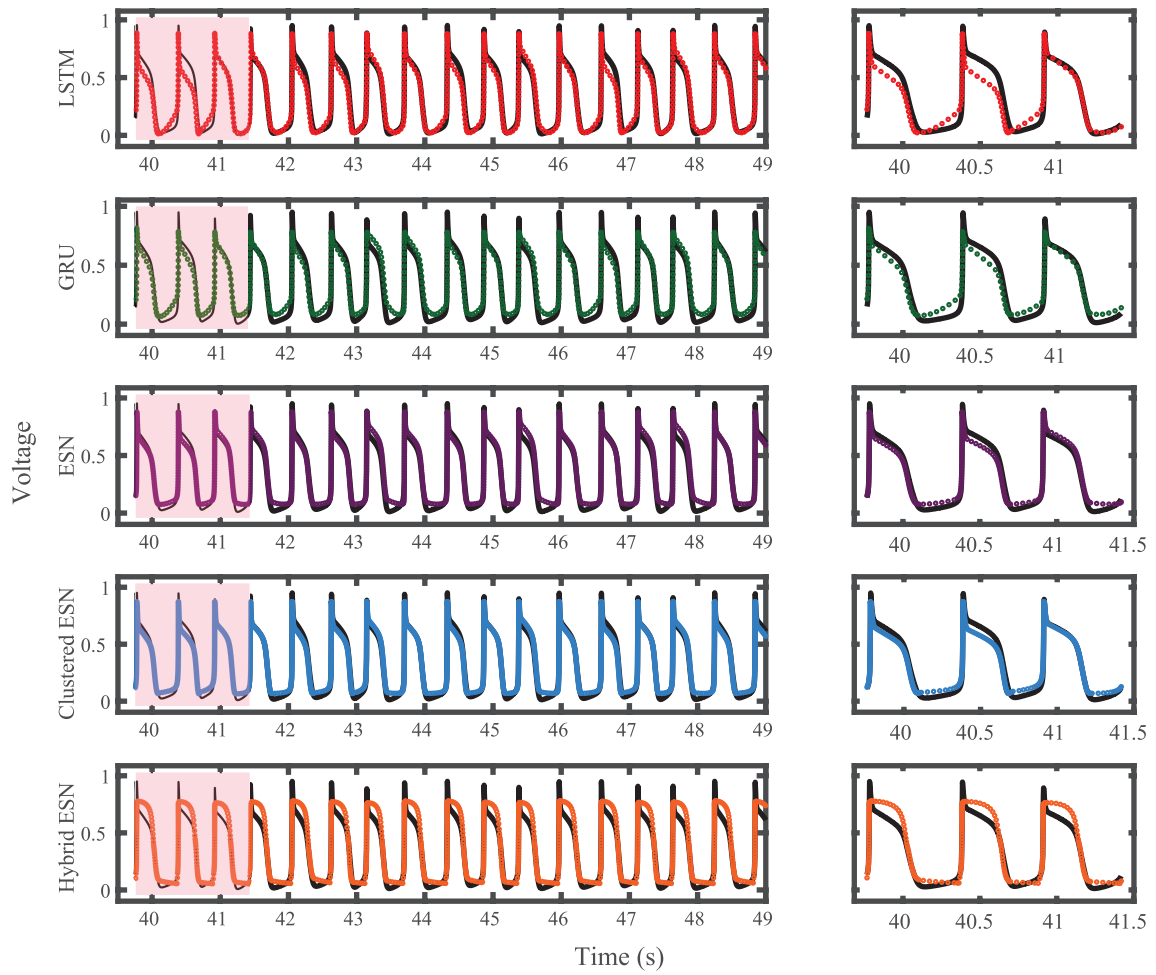

FIGURE 7 | Noble dataset action potential prediction results obtained for the five methods using a fixed network size of 100 neurons. Test data are shown in black for reference and the predictions in color. **(Left)** All predicted APs. **(Right)** Zoomed view of the first three predicted APs.

other methods produce depolarization during this phase. The performance does not vary significantly over the full prediction series, although some individual action potentials are not fit well. As shown in **Figure 11**, the largest APD differences from the true values occur for the hybrid ESN, with the clustered ESN yielding especially good results. The ESN and hybrid ESN consistently underestimate APD values, and the APD values predicted by the LSTM and GRU methods are generally less extreme than the true APDs (that is, the long APDs are predicted to be shorter and the short APDs are predicted to be longer).

As the number of neurons is changed, there is no clear effect on accuracy, except possibly for the hybrid ESN, which appears to have a slight trend toward lower RMSE with more neurons; see **Figure 12**. In contrast, the time required for training and prediction shows the same trend as for the other data sets, with the LSTM and GRU approaches requiring about two orders of magnitude more time for the same network size than the ESN and clustered ESN approaches, and the hybrid ESN in between. All the methods show a trend toward increasing time with increasing network size, except for the hybrid ESN, which as before appears insensitive to network size.

4. DISCUSSION

In this paper, we tested five different ML time series prediction methods, two based on RNNs and three based on a type of reservoir computing, to predict irregular voltage dynamics arising from random or chaotic effects for three cardiac datasets. We found that for clinically relevant intervals (1–3 s, or 6–19 beats at 160 ms) for the detection of cardiac arrhythmia in embedded devices (Madhavan and Friedman, 2013), we were able to predict voltage traces that closely match the true dynamics. We showed that for all datasets considered, all five prediction methods could produce accurate forecasts of both voltage and APD for around 15–20 action potentials (as long as was tested). With the exception of GRU predictions for the Noble dataset, RMSE values on the order of 0.1 normalized voltage units or lower could be attained for every combination of dataset and

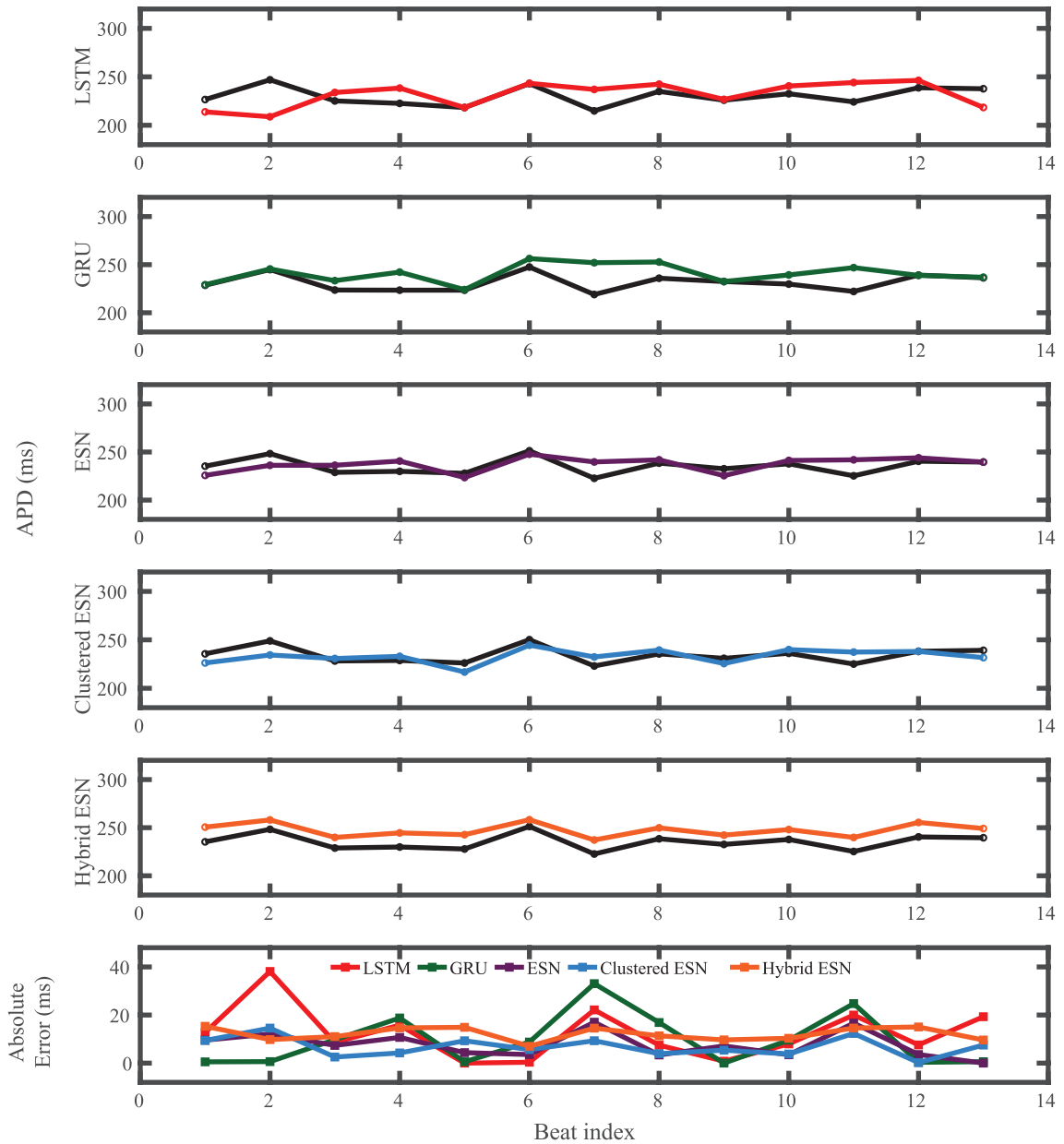

FIGURE 8 | Noble dataset APD prediction results obtained for the five methods using a fixed network size of 100 neurons. APDs from data used for testing are shown in black for reference and predicted APDs are shown in color. Absolute error in APD prediction is shown in the bottom subplot, with color corresponding to prediction method.

prediction method. APD errors were typically less than 5 ms for both the FK and experimental datasets, with larger typical errors of around 20 ms (sometimes more) for the Noble dataset. Over the measured interval, none of the methods exhibited any long-term trend in the APD error, indicating that the methods have seen sufficient training data to accurately model the real action potential response to stimulation and the associated APD distribution.

In addition, we demonstrated that the ESN approaches achieved lower error than the RNN approaches for the synthetic datasets and that the hybrid ESN achieved the best accuracy for the experimental dataset. Furthermore, the accuracy obtained, as measured by RMSE, was largely independent of network size. The time required for training and prediction typically was about two orders of magnitude lower for the ESN and clustered ESN architectures compared to the LSTM and GRU approaches, with the hybrid ESN timing in between. Computation time also generally grew with the network size, with the exception of the hybrid ESN, where computational time was essentially constant across all network sizes considered. We expect this insensitivity

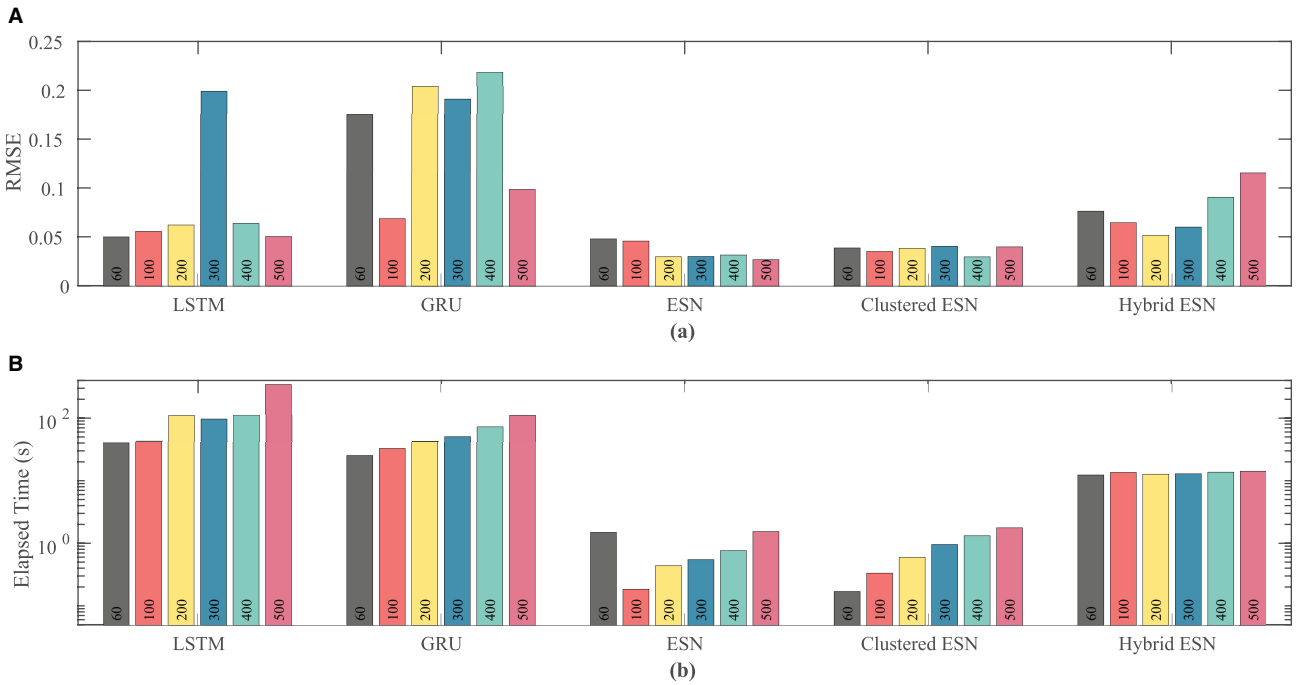

FIGURE 9 | Comparison of RMSE **(A)** and computational time **(B)** for each method and network size tested for the Noble dataset.

occurred because the time associated with solving the knowledge-based model (essentially a set of differential equations describing a cardiac cell), which is independent of network size, was more expensive than the cost of the ESN itself and thus dominated the total training and prediction time.

We investigated different types of dynamics, including those influenced by underlying randomness and chaos. We found the lowest RMSE values for the FK and experimental datasets and the highest RMSE values for the chaotically-driven Noble dataset. For the FK dataset, the hybrid ESN typically achieved the lowest error for all but the smallest network size, with the other ESN approaches achieving slightly less error than the LSTM and GRU methods. The Noble dataset elicited a particularly poor performance for the GRU method, with RMSE values typically three or four times larger than for the other methods; also, the hybrid ESN did not perform as well as the other ESNs. However, it is possible that use of a univariate time series in this case contributed to lower accuracy, rather than just the chaotic dynamics alone. For the experimental dataset, which likely has elements of both randomness and chaos, the hybrid ESN generally achieved the lowest error, with the other methods producing similar RMSE values. Overall, our results indicate that the ESN architecture provides better performance than LSTM and GRU approaches for the voltage forecasting task.

4.1. Effects of Algorithmic Choices

The action potential time series used in this work were highly imbalanced. In this study, although we found the need to downsample the original data for use with testing and training, we did not perform systematic studies regarding how to optimize this task. As a general observation, starting from the initial highly imbalanced time series, by increasing the sampling spacing and reducing the number of data points, the training and testing errors were reduced. However, the information loss caused by removing data points is the obvious side effect if the spacing becomes too large. Our choice of tying the time spacing to changes in voltage ensured good resolution during rapidly changing parts of action potentials, including the upstroke, but led to a lack of points during the rest and plateau phases, contributing to apparent errors during these times of slow changes in voltage. Further studies are required to investigate various spacing and resampling strategies to propose an optimal approach.

Although our results illustrate that ESNs provide the best prediction accuracy together with the lowest computational times in most cases for the methods and datasets considered, the ESN approach shows the most sensitivity to the hyperparameter and network parameter values. Our grid search results demonstrated a wide variability in the prediction performance obtained by various ESNs with very similar configurations. This motivates more study to improve the robustness of this approach. Among the three RC techniques used in this work, the hybrid ESN showed the least sensitivity to the hyperparameter values. We expect that the knowledge-based model promotes the predictive ability of the network by generating an approximate action potential, which the network perturbs to resolve the precise AP shape.

Incorporating the pacing stimulus into a multivariate input setup considerably improved the prediction performance of the network over using a univariate voltage input and extended the

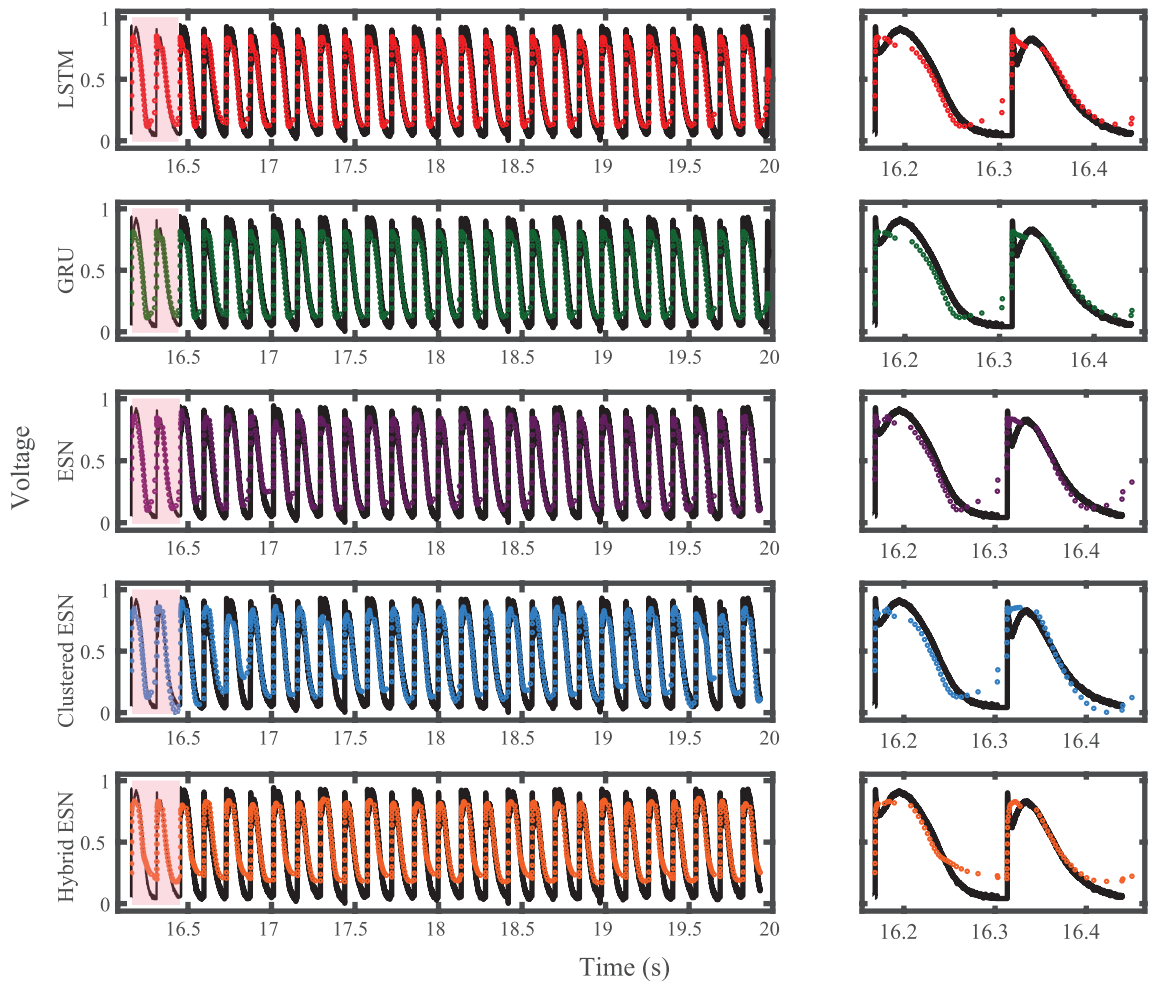

FIGURE 10 | Experimental dataset action potential prediction results obtained for the five methods using a fixed network size of 100 neurons. Test data are shown in black for reference and the predictions in color. **(Left)** All predicted APs. **(Right)** Zoomed view of the first two predicted APs.

forecasting horizon to a higher number of beats. In the absence of the stimulus information, depending on the dynamics of the system, the predictions remain accurate only for the first few beats after the training.

4.2. Limitations and Future Work

Our study contains a number of limitations. First, we studied a limited number of datasets. It is possible that different types of dynamics (e.g., more strongly chaotic) could lead to different results, and in particular experimental data from other sources could prove more difficult to predict. In addition, we did not study how much training data was needed to obtain good results. Furthermore, it remains an important open question how long the action potential predictions will remain accurate without deteriorating, although in this case we have found lower bounds.

We also considered a small number of time series prediction methods. There are many variations on these methods (Chandra et al., 2021; Han et al., 2021) and it is possible that performance improvements could be achieved. Even choosing different settings for the methods considered, such as a different number of clusters for the clustered ESNs, potentially could affect performance. There are also different types of prediction methods that we did not consider. For example, ESNs have been connected to vector autoregression (VAR) (Bollt, 2021), thereby motivating additional studies of VAR for prediction. It also would be interesting to study the accuracy of predictions of APD obtained by training on APD values only.

For the hybrid ESN, we only considered the use of one knowledge-based model, the Corrado-Niederer update of the Mitchell-Schaeffer model. It is possible that different model choices could affect the accuracy or computational time of the hybrid method; for example, an even simpler two-variable model like the FitzHugh-Nagumo model could potentially make the hybrid ESN approach more competitive with the other ESNs considered here, while a knowledge-based model that is matched to the data-generating model might present a near-trivial prediction task. Additionally, more complex cardiac cell models with detailed calcium dynamics may have an impact on

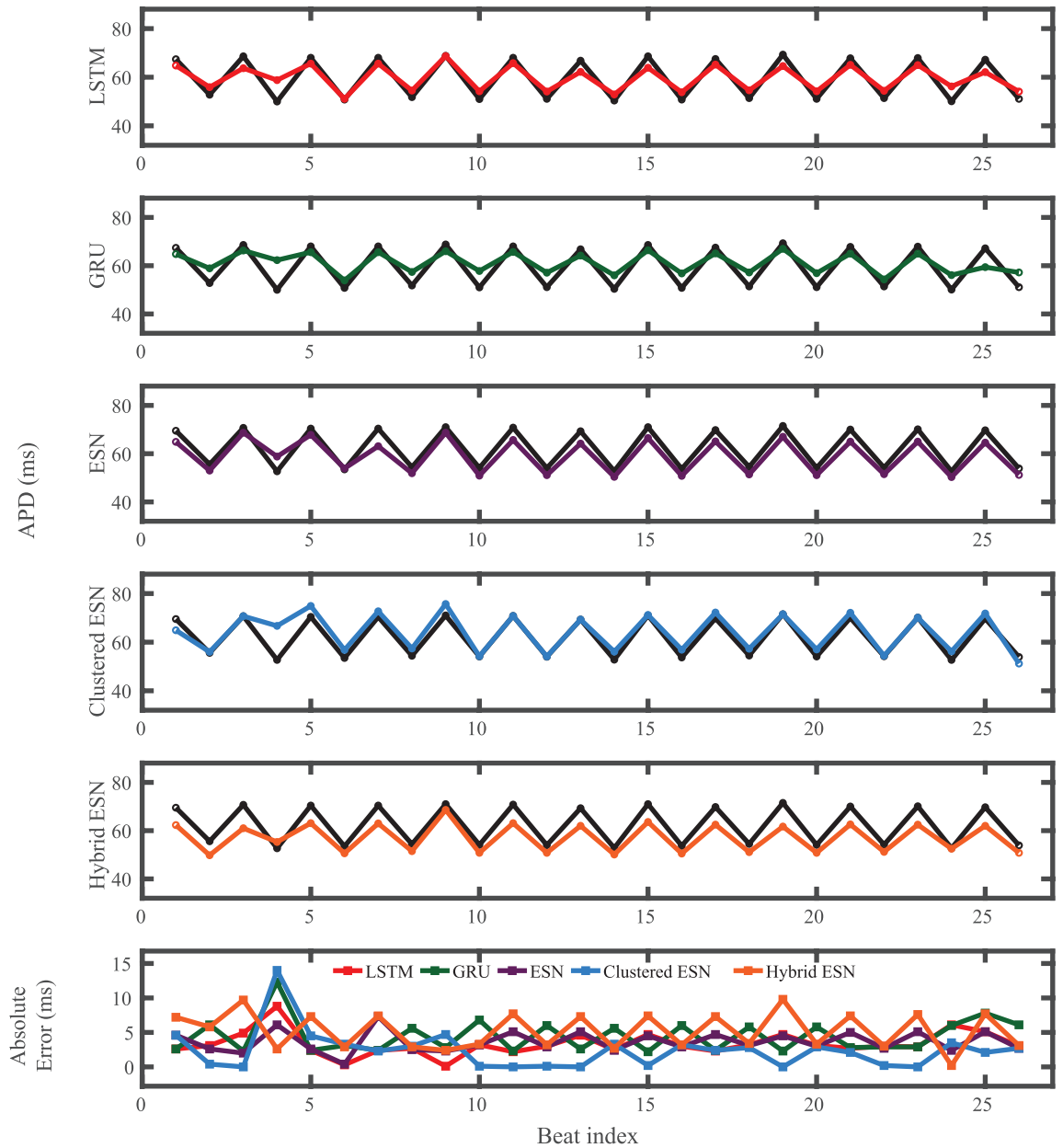

FIGURE 11 | Experimental dataset APD prediction results obtained for the five methods using a fixed network size of 100 neurons. APDs from data used for testing are shown in black for reference and predicted APDs are shown in color. Absolute error in APD prediction is shown in the bottom subplot, with color corresponding to prediction method.

long-term tissue memory. In practice, this long-term change in the cardiac cell may lessen the predictive power of the presented ML models over long time intervals.

We also note that there is a close connection between ML-based methods and data assimilation. In the cardiac case, Kalman filter-based methods including data assimilation have been used thus far for reconstruction (Muñoz and Otani, 2010, 2013; Hoffman et al., 2016; Hoffman and Cherry, 2020; Marcotte et al., 2021), but they also can be used for forecasting, as is more typical in data assimilation's original weather forecasting context (Hunt et al., 2007). It may be beneficial to pursue approaches that seek to merge data assimilation and machine learning for this task (Albers et al., 2018; Brajard et al., 2020; Gottwald and Reich, 2021).

Along with extensions of our present work to address the issues discussed above, in the future we intend to consider predicting cardiac voltage dynamics during the development of arrhythmias. We expect this goal may necessitate the use of spatially extended models of cardiac tissue as part of the prediction process, although handling the information from spatial neighbors requires very large networks that will pose new computational challenges. The combination

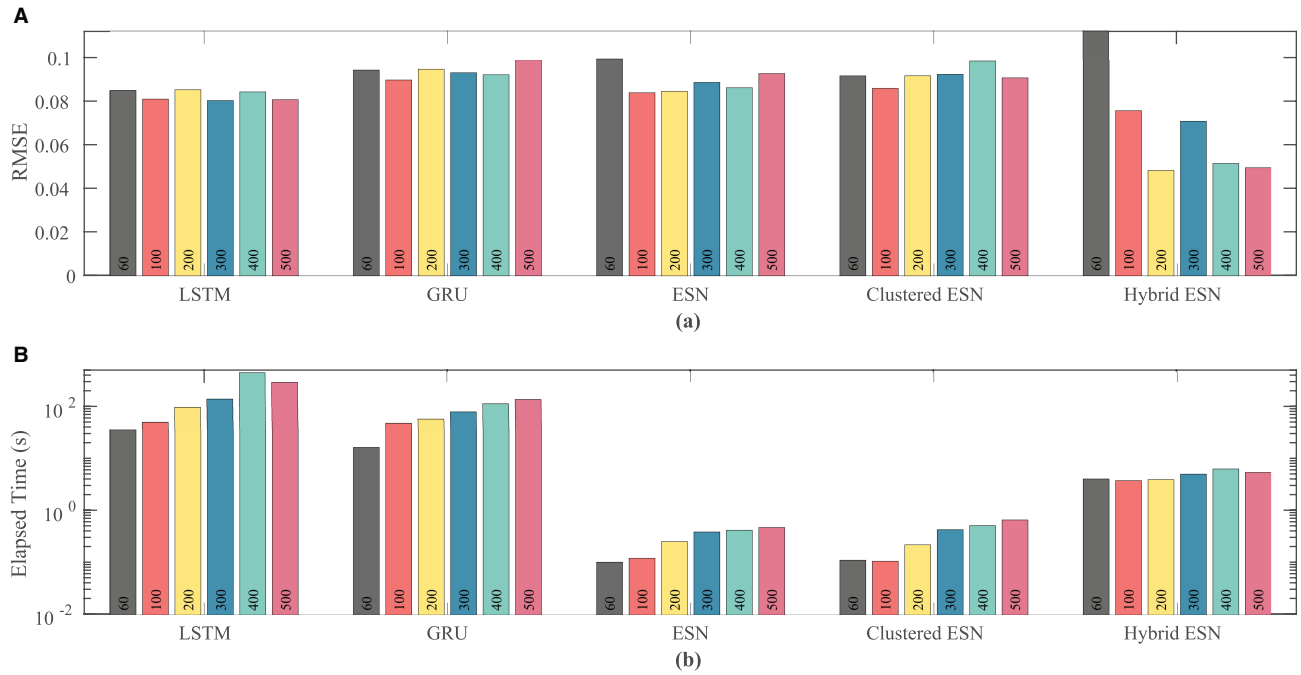

FIGURE 12 | Comparison of RMSE **(A)** and computational time **(B)** for each method and network size tested for the experimental dataset.

of ESNs and local states (Pathak et al., 2018; Zimmermann and Parlitz, 2018) or specialized deep-learning architectures (Herzog et al., 2018) may be useful in tackling such problems, but these methods remain computationally demanding and may require new approaches. In addition, we may need to carefully consider the types of dynamics included in the training data in order to accurately predict transitions between different types of dynamics, such as the transition from normal rhythm to tachycardia or the transition from tachycardia to fibrillation. Accurate prediction of such transitions may lead to advances in control designed to prevent the development of fatal arrhythmias.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The animal study was reviewed and approved by the Office of Research Integrity Assurance of Georgia Tech under IACUC A100416.

AUTHOR CONTRIBUTIONS

SS, FF, YS, and EC contributed to developing the study concept. SS, CM, and EC contributed to study design. SS and EC developed the synthetic datasets and analyzed and data. CH and FF obtained the experimental datasets. SS, FF, and EC wrote the manuscript. CM, CH, and YS critically revised the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This study was supported in part by the National Science Foundation grants CMMI-2011280 (EC) and CMMI-1762553 (FF) and by the National Institutes of Health grant 1R01HL143450-01 (EC and FF).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphys.2021.734178/full#supplementary-material

REFERENCES

Albers, D. J., Levine, M. E., Stuart, A., Mamykina, L., Gluckman, B., and Hripcsak, G. (2018). Mechanistic machine learning: how data assimilation leverages physiologic knowledge using Bayesian inference to forecast the future, infer the present, and phenotype. *J. Am. Med. Informat. Association* 25, 1392–1401. doi: 10.1093/jamia/ocy106

Berger, C. M., Cain, J. W., Socolar, J. E. S., and Gauthier, D. J. (2007). Control of electrical alternans in simulations of paced myocardium

using extended time-delay autosynchronization. *Phys. Rev. E* 76:041917. doi: 10.1103/PhysRevE.76.041917

Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., and Jenssen, R. (2017). *Other Recurrent Neural Networks Models*. Cham: Springer International Publishing. doi: 10.1007/978-3-319-70338-1_4

Bollobás, B. (2001). "Random graphs," in *Cambridge Studies in Advanced Mathematics* (Cambridge: Cambridge University Press). doi: 10.1017/CBO9780511814068

Bollt, E. (2021). On explaining the surprising success of reservoir computing forecaster of chaos? the universal machine learning dynamical system with contrast to VAR and DMD. *Chaos* 31:013108. doi: 10.1063/5.0024890

Brajard, J., Carrassi, A., Bocquet, M., and Bertino, L. (2020). Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model. *J. Comput. Sci.* 44:101171. doi: 10.1016/j.jocs.2020.101171

Chandra, R., Goyal, S., and Gupta, R. (2021). Evaluation of deep learning models for multi-step ahead time series prediction. *IEEE Access* 9, 83105–83123. doi: 10.1109/ACCESS.2021.3085085

Chattopadhyay, A., Hassanzadeh, P., and Subramanian, D. (2020). Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Process. Geophys.* 27, 373–389. doi: 10.5194/npg-27-373-2020

Chen, D. D., Gray, R. A., Uzelac, I., Herndon, C., and Fenton, F. H. (2017). Mechanism for amplitude alternans in electrocardiograms and the initiation of spatiotemporal chaos. *Phys. Rev. Lett.* 118:168101. doi: 10.1103/PhysRevLett.118.168101

Cherry, E. M. (2017). Distinguishing mechanisms for alternans in cardiac cells using constant-diastolic-interval pacing. *Chaos* 27:093902. doi: 10.1063/1.4999354

Chialvo, D. R., Gilmour, R. F. Jr., and Jalife, J. (1990). Low dimensional chaos in cardiac tissue. *Nature* 343, 653–657. doi: 10.1038/343653a0

Christini, D. J., Riccio, M. L., Culianu, C. A., Fox, J. J., Karma, A., Gilmour, J.r., et al. (2006). Control of electrical alternans in canine cardiac Purkinje fibers. *Phys. Rev. Lett.* 96:104101. doi: 10.1103/PhysRevLett.96.104101

Corrado, C., and Niederer, S. A. (2016). A two-variable model robust to pacemaker behaviour for the dynamics of the cardiac action potential. *Math. Biosci.* 281, 46–54. doi: 10.1016/j.mbs.2016.08.010

Deng, Z., and Zhang, Y. (2006). "Complex systems modeling using scale-free highly-clustered echo state network," in *Proceedings of the International Joint Conference on Neural Networks*, IJCNN 2006, part of the IEEE World Congress on Computational Intelligence (Vancouver, BC: IEEE), 3128–3135. doi: 10.1109/IJCNN.2006.247295

Doan, N. A. K., Polifke, W., and Magri, L. (2019). "Physics-informed echo state networks for chaotic systems forecasting," in *Computational Science - ICCS 2019. ICCS 2019. Lecture Notes in Computer Science*, Vol. 11539, eds J. M. F. Rodrigues, P. J. S. Cardoso, J. Monteiro, R. Lam, V. V. Krzhizhanovskaya, M. H. Lees, J. J. Dongarra, and P. M.A. Sloot (Cham: Springer International Publishing), 192–198.

Dubois, P., Gomez, T., Planckaert, L., and Perret, L. (2020). Data-driven predictions of the lorenz system. *Phys. D Nonlinear Phenomena* 408:132495. doi: 10.1016/j.physd.2020.132495

Echebarria, B., and Karma, A. (2002). Spatiotemporal control of cardiac alternans. *Chaos* 12, 923–930. doi: 10.1063/1.1501544

Fenton, F., and Karma, A. (1998). Vortex dynamics in three-dimensional continuous myocardium with fiber rotation: filament instability and fibrillation. *Chaos* 8, 20–47. doi: 10.1063/1.166311

Fenton, F. H., Cherry, E. M., Hastings, H. M., and Evans, S. J. (2002). Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity. *Chaos* 12, 852–892. doi: 10.1063/1.1504242

Fenton, F. H., Cherry, E. M., and Kornreich, B. G. (2008). Termination of equine atrial fibrillation by quinidine: an optical mapping study. *J. Vet. Cardiol.* 10, 87–103. doi: 10.1016/j.jvc.2008.10.002

Garzón, A., Grigoriev, R. O., and Fenton, F. H. (2009). Model-based control of cardiac alternans on a ring. *Phys. Rev. E* 80:021932. doi: 10.1103/PhysRevE.80.021932

Garzon, A., Grigoriev, R. O., and Fenton, F. H. (2014). Continuous-time control of alternans in long Purkinje fibers. *Chaos* 24:033124. doi: 10.1063/1.4893295

Gizzi, A., Cherry, E. M., Gilmour, R. F. Jr., Luther, S., Filippi, S., et al. (2013). Effects of pacing site and stimulation history on alternans dynamics and the development of complex spatiotemporal patterns in cardiac tissue. *Front. Cardiac Electrophysiol.* 4:71. doi: 10.3389/fphys.2013.00071

Gottwald, G. A., and Reich, S. (2021). Supervised learning from noisy observations: combining machine-learning techniques with data assimilation. *Phys. D Nonlinear Phenomena* 423:132911. doi: 10.1016/j.physd.2021.132911

Guevara, M. R., Ward, G., Shrier, A., and Glass, L. (1984). Electrical alternans and period-doubling bifurcations. *Comput. Cardiol.* 11, 167–170.

Han, Z., Zhao, J., Leung, H., Ma, K. F., and Wang, W. (2021). A review of deep learning models for time series prediction. *IEEE Sens. J.* 21, 7833–7848. doi: 10.1109/JSEN.2019.2923982

Herzog, S., Wörgötter, F., and Parlitz, U. (2018). Data-driven modeling and prediction of complex spatio-temporal dynamics in excitable media. *Front. Appl. Math. Stat.* 4:60. doi: 10.3389/fams.2018.00060

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hoffman, M. J., and Cherry, E. M. (2020). Sensitivity of a data-assimilation system for reconstructing three-dimensional cardiac electrical dynamics. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 378:20190388. doi: 10.1098/rsta.2019.0388

Hoffman, M. J., LaVigne, N. S., Scorse, S. T., Fenton, F. H., and Cherry, E. M. (2016). Reconstructing three-dimensional reentrant cardiac electrical wave dynamics using data assimilation. *Chaos* 26:013107. doi: 10.1063/1.4940238

Hunt, B. R., Kostelich, E. J., and Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: a local ensemble transform Kalman filter. *Phys. D* 230, 112–126. doi: 10.1016/j.physd.2006.11.008

Ing, C.-K. (2003). Multistep prediction in autoregressive processes. *Econometr. Theory* 19, 254–279. doi: 10.1017/S0266466603192031

Jaeger, H. (2002). *Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the "echo state network" Approach*. Bonn: GMD-Forschungszentrum Informationstechnik.

Junior, L. O., Stelzer, F., and Zhao, L. (2020). "Clustered echo state networks for signal observation and frequency filtering," in *Anais do VIII Symposium on Knowledge Discovery, Mining and Learning* (Porto Alegre: SBC), 25–32, doi: 10.5753/kdmile.2020.11955

Kappadan, V., Telele, S., Uzelac, I., Fenton, F., Parlitz, U., Luther, S., et al. (2020). High-resolution optical measurement of cardiac restitution, contraction, and fibrillation dynamics in beating vs. blebbistatin-uncoupled isolated rabbit hearts. *Front. Physiol.* 11:464. doi: 10.3389/fphys.2020.00464

Kingma, D. P., and Ba, J. (2014). ADAM: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kulkarni, K., Lee, S. W., Kluck, R., and Tolkacheva, E. G. (2018). Real-time closed loop diastolic interval control prevents cardiac alternans in isolated whole rabbit hearts. *Ann. Biomed. Eng.* 46, 555–566. doi: 10.1007/s10439-018-1981-2

Kutz, J. N. (2013). *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*. New York, NY: Oxford University Press, Inc.

Lorenz, E. N. (1963). Deterministic non-periodic flow. *J. Atmos. Sci.* 20, 130–141. doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2

Lukoševičius, M. (2012). "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade: Second Edition*, eds G. Montavon, G. B. Orr, and K. R.Müller (Berlin; Heidelberg: Springer), 659–686. doi: 10.1007/978-3-642-35289-8_36

Lukoševičius, M., and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* 3, 127–149. doi: 10.1016/j.cosrev.2009.03.005

Madhavan, M., and Friedman, P. A. (2013). Optimal programming of implantable cardiac-defibrillators. *Circulation* 128, 659–672. doi: 10.1161/CIRCULATIONAHA.112.000542

Marcotte, C. D., Fenton, F. H., Hoffman, M. J., and Cherry, E. M. (2021). Robust data assimilation with noise: applications to cardiac dynamics. *Chaos* 31:013118. doi: 10.1063/5.0033539

Mitchell, C. C., and Schaeffer, D. G. (2003). A two-current model for the dynamics of cardiac membrane. *Bull. Math. Biol.* 65, 767–793. doi: 10.1016/S0092-8240(03)00041-7

Moniz, N., Branco, P., and Torgo, L. (2017). Resampling strategies for imbalanced time series forecasting. *Int. J. Data Sci. Anal.* 3, 161–181. doi: 10.1007/s41060-017-0044-3

Muñoz, L. M., and Otani, N. F. (2010). Enhanced computer modeling of cardiac action potential dynamics using experimental data-based feedback. *Comput. Cardiol.* 37, 837–840.

Muñoz, L. M., and Otani, N. F. (2013). "Kalman filter based estimation of ionic concentrations and gating variables in a cardiac myocyte model," in *Computing in Cardiology 2013* (Zaragoza), 53–56.

Noble, D. (1962). A modification of the Hodgkin-Huxley equations applicable to Purkinje fibre action and pace-maker potentials. *J. Physiol.* 160, 317–352. doi: 10.1113/jphysiol.1962.sp006849

Nolasco, J. B., and Dahlen, R. W. (1968). A graphic method for the study of alternation in cardiac action potentials. *J. Appl. Physiol.* 25, 191–196. doi: 10.1152/jappl.1968.25.2.191

Oh, D. K. (2020). Toward the fully physics-informed echo state network-an ode approximator based on recurrent artificial neurons. *arXiv preprint arXiv:2011.06769*.

Otani, N. F. (2017). Theory of the development of alternans in the heart during controlled diastolic interval pacing. *Chaos* 27:093935. doi: 10.1063/1.5003250

Pastore, J. M., Girouard, S. D., Laurita, K. R., Akar, F. G., and Rosenbaum, D. S. (1999). Mechanism linking t-wave alternans to the genesis of cardiac fibrillation. *Circulation* 99, 1385–1394. doi: 10.1161/01.CIR.99.10.1385

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. (2018). Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* 120:024102. doi: 10.1103/PhysRevLett.120.024102

Rappel, W. J., Fenton, F., and Karma, A. (1999). Spatiotemporal control of wave instabilities in cardiac tissue. *Phys. Rev. Lett.* 83, 456–459. doi: 10.1103/PhysRevLett.83.456

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* 656, 5–28. doi: 10.1017/S0022112010001217

Stock, J. H., and Watson, M. W. (2001). Vector autoregressions. *J. Econ. Perspect.* 15, 101–115. doi: 10.1257/jep.15.4.101

Sun, C., Song, M., Hong, S., and Li, H. (2020). A review of designs and applications of echo state networks. *arXiv preprint arXiv:2012.02974*.

Watanabe, M. A., Fenton, F. H., Evans, S. J., Hastings, H. M., and Karma, A. (2001). Mechanisms for discordant alternans. *J. Cardiovasc. Electrophysiol.* 12, 196–206. doi: 10.1046/j.1540-8167.2001.00196.x

Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. (2020). Integrating physics-based modeling with machine learning: a survey. *arXiv preprint arXiv:2003.04919*.

Yildiz, I. B., Jaeger, H., and Kiebel, S. J. (2012). Re-visiting the echo state property. *Neural Netw.* 35, 1–9. doi: 10.1016/j.neunet.2012.07.005

Zimmermann, R. S., and Parlitz, U. (2018). Observing spatio-temporal dynamics of excitable media using reservoir computing. *Chaos* 28:043118. doi: 10.1063/1.5022276

Zlochiver, S., Johnson, C., and Tolkacheva, E. G. (2017). Constant DI pacing suppresses cardiac alternans formation in numerical cable models. *Chaos* 27:093903. doi: 10.1063/1.4999355

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary Material

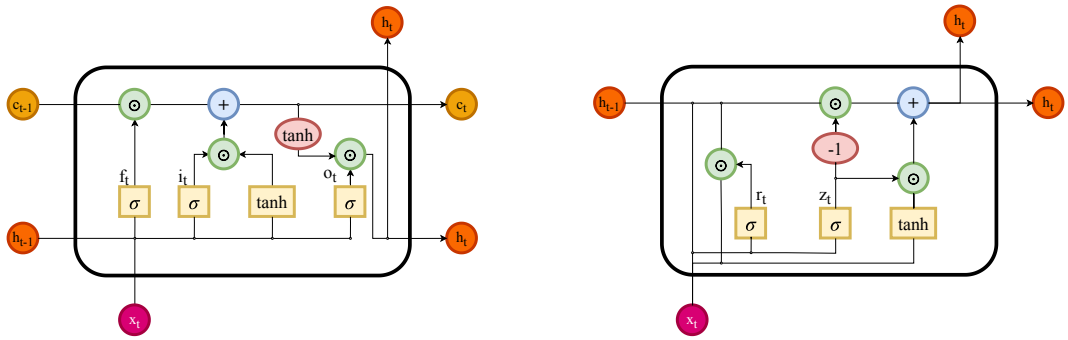

Figure S1. Architectures for the recurrent neural network approaches. **(a)** Long-short term memory network components. **(b)** Gated recurrent unit components.

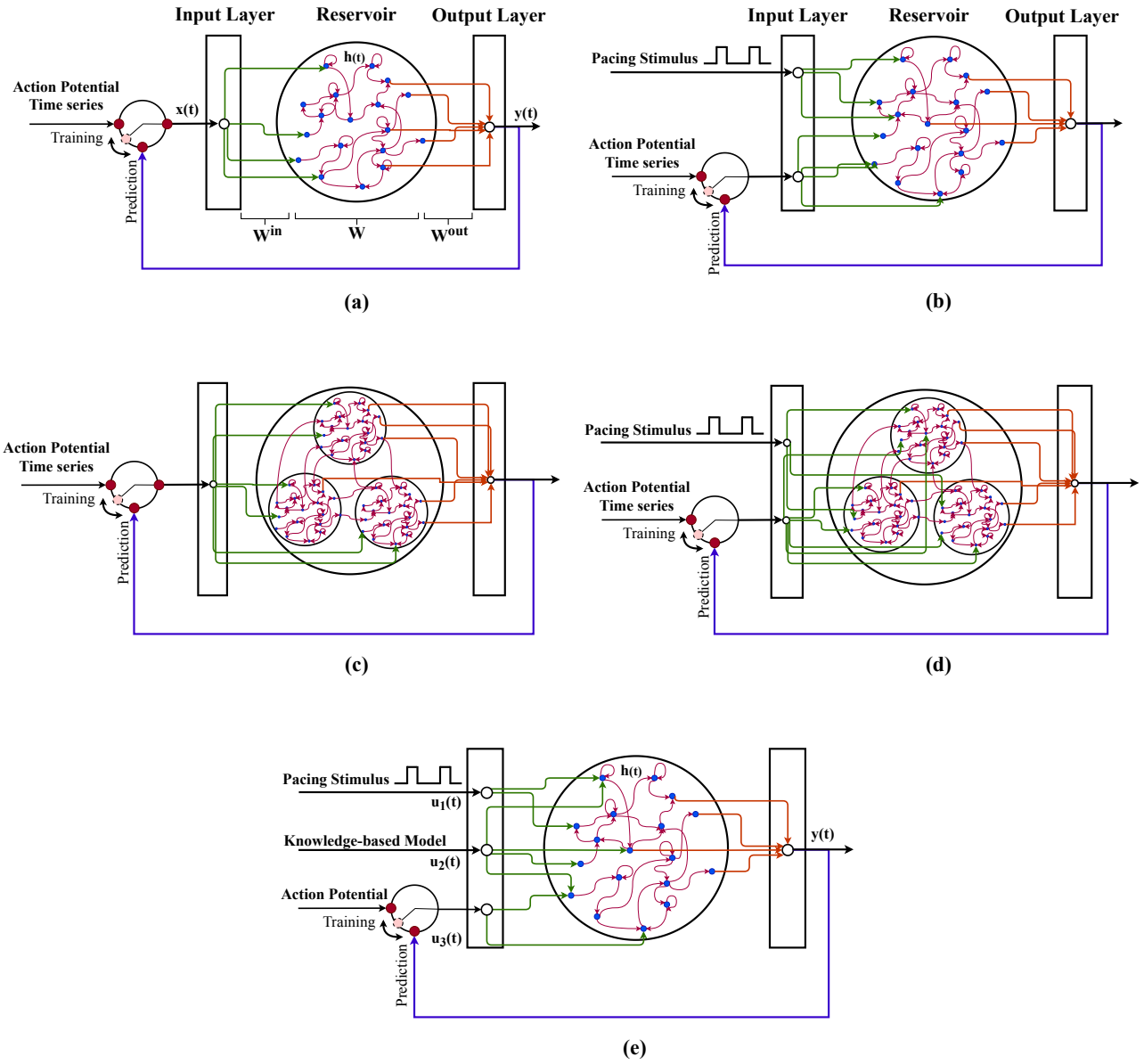

Figure S2. Architectures for the reservoir computing approaches. **(a)** ESN components for univariate time series. **(b)** ESN components for multivariate time series including stimulus information. **(c)** Clustered ESN components for univariate time series. **(d)** Clustered ESN components for multivariate time series including stimulus information. **(e)** Hybrid ESN components for multivariate time series including stimulus information.

.

Table S1. Hyperparameter values used for the grid search optimization for each prediction method. The resampling voltage threshold δ defines the minimum difference between the voltage values of each two consecutive data points, which is used as the first criterion for resampling the action potential time series. The resampling time threshold τ determines the maximum time gap (in ms) between each two consecutive data points. The learning rate η is the initial learning rate used by the Adam optimizer for training the gated RNNs. Input weight scales σ_{in}^1, σ_{in}^2, and σ_{in}^3 are the scalars that are multiplied by the corresponding columns of the input weight matrix in the ESNs to adjust the magnitude of the input signals including the action potential, the pacing stimulus, and the knowledge-based model, if applicable. The reservoir weight matrix is also scaled such that its spectral radius, defined as the largest among the absolute values of the eigenvalues, is equal to the selected spectral radius ρ. The leaking rate α determines the amount of excitation discarded by the leaky integrator neurons and used to control the rate of the reservoir update dynamics discretized in time. The regularization parameter λ determines the ridge regression regularization factor employed for calculation of the readout weights in ESNs. The connection probability pr is the probability of having an edge between each two neurons in the reservoir, which controls the sparsity of the reservoir graph. The inter-cluster probability pr_c is the probability of having an edge between each two nodes from different sub-reservoirs in the clustered ESN approach.

Methods	Parameters	Values
LSTM	Resampling voltage threshold (δ)	$\{0.00, 0.01, 0.02, 0.03, 0.04\}$
	Resampling time threshold (τ)	$\{20, 30, 40, 50\}$
	Number of layers	$\{1, 2, 4\}$
	Learning rate (η)	$\{0.001, 0.002, 0.005, 0.010, 0.150\}$
GRU	Resampling voltage threshold (δ)	$\{0.00, 0.01, 0.02, 0.03, 0.04\}$
	Resampling time threshold (τ)	$\{20, 30, 40, 50\}$
	Number of layers	$\{1, 2, 4\}$
	Learning rate (η)	$\{0.001, 0.002, 0.005, 0.010, 0.150\}$
ESN	Resampling voltage threshold (δ)	$\{0.00, 0.01, 0.02, 0.03, 0.04\}$
	Resampling time threshold (τ)	$\{20, 30, 40, 50\}$
	Input weight scale (action potential, σ_{in}^1)	$\{0.02, 0.05, 0.10, 0.20, 0.50, 0.80\}$
	Input weight scale (pacing stimulus , σ_{in}^2)	$\{0.02, 0.05, 0.10, 0.20, 0.50, 0.80\}$
	Spectral radius (ρ)	$\{0.80, 0.85, 0.90, 0.99, 1.05, 1.25, 1.50\}$
	Leaking rate (α)	$\{0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00\}$
	Regularization (λ)	$\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	Connection probability (pr)	$\{0.01, 0.02, 0.05, 0.10, 0.15, 0.20\}$
Clustered ESN	Resampling voltage threshold (δ)	$\{0.00, 0.01, 0.02, 0.03, 0.04\}$
	Resampling time threshold (τ)	$\{20, 30, 40, 50\}$
	Input weight scale (action potential, σ_{in}^1)	$\{0.02, 0.05, 0.10, 0.20, 0.50, 0.80\}$
	Input weight scale (pacing stimulus , σ_{in}^2)	$\{0.02, 0.05, 0.10, 0.20, 0.50, 0.80\}$
	Number of clusters (n_c)	$\{2, 3, 4, 5\}$
	Spectral radius (ρ)	$\{0.80, 0.85, 0.90, 0.99, 1.05, 1.25, 1.50\}$
	Leaking rate (α)	$\{0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00\}$
	Regularization (λ)	$\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	Intra-cluster connection probability (pr)	$\{0.60, 0.7, 0.80, 0.85, 0.90, 0.95, 0.98\}$
	Inter-cluster connection probability (pr_c)	$\{0.01, 0.02, 0.05, 0.10, 0.15, 0.20\}$
Hybrid ESN	Resampling voltage threshold (δ)	$\{0.00, 0.01, 0.02, 0.03, 0.04\}$
	Resampling time threshold (τ)	$\{20, 30, 40, 50\}$
	Input weight scale (action potential, σ_{in}^1)	$\{0.02, 0.05, 0.10, 0.20, 0.50, 0.80\}$
	Input weight scale (pacing stimulus , σ_{in}^2)	$\{0.02, 0.05, 0.10, 0.20, 0.50, 0.80\}$
	Input weight scale (knowledge based model, σ_{in}^3)	$\{0.02, 0.05, 0.10, 0.20, 0.50, 0.80\}$
	Spectral radius (ρ)	$\{0.80, 0.85, 0.90, 0.99, 1.05, 1.25, 1.50\}$
	Leaking rate (α)	$\{0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00\}$
	Regularization (λ)	$\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	Connection probability (pr)	$\{0.01, 0.02, 0.05, 0.10, 0.15, 0.20\}$

Table S2. Optimal hyperparameters found by grid search for the LSTM method for each dataset and network size. Hyperparameter definitions are given in Table S1.

Action potential	Network size	δ	τ	η	$layers$
Fenton-Karma	60	0.04	40	0.010	1
	100	0.04	40	0.010	1
	200	0.04	40	0.005	1
	300	0.01	30	0.002	4
	400	0.02	40	0.002	1
	500	0.01	30	0.002	4
Noble	60	0.04	40	0.002	4
	100	0.04	40	0.002	4
	200	0.02	40	0.002	4
	300	0.04	40	0.002	2
	400	0.04	40	0.010	4
	500	0.02	30	0.005	4
Experimental Data	60	0.01	40	0.005	2
	100	0.01	40	0.002	2
	200	0.01	40	0.002	2
	300	0.01	30	0.005	4
	400	0.01	40	0.002	1
	500	0.01	30	0.002	2

Table S3. Optimal hyperparameters found by grid search for the GRU method for each dataset and network size. Hyperparameter definitions are given in Table S1.

Action potential	Network size	δ	τ	η	$layers$
Fenton-Karma	60	0.04	40	0.002	1
	100	0.04	40	0.002	2
	200	0.02	40	0.002	1
	300	0.04	40	0.010	4
	400	0.04	40	0.005	4
	500	0.02	30	0.002	4
Noble	60	0.04	40	0.005	1
	100	0.04	40	0.002	4
	200	0.04	40	0.010	1
	300	0.04	40	0.002	1
	400	0.04	40	0.005	1
	500	0.04	40	0.002	4
Experimental Data	60	0.01	40	0.010	1
	100	0.01	40	0.005	4
	200	0.01	40	0.002	4
	300	0.01	40	0.002	1
	400	0.01	40	0.002	1
	500	0.01	30	0.002	1

Table S4. Optimal hyperparameters found by grid search for the ESN method for each dataset and network size. Hyperparameter definitions are given in Table S1.

Action potential	Network size	δ	τ	σ_{in}^1	σ_{in}^2	pr	ρ	α	λ
Fenton-Karma	60	0.02	40	0.02	0.10	0.05	1.05	0.70	10^{-6}
	100	0.03	40	0.10	0.10	0.05	1.05	0.80	10^{-6}
	200	0.01	40	0.02	0.10	0.10	1.05	0.80	10^{-5}
	300	0.01	40	0.02	0.10	0.05	1.05	0.80	10^{-5}
	400	0.02	30	0.10	0.10	0.02	0.85	1.00	10^{-4}
	500	0.02	40	0.02	0.10	0.02	1.05	0.70	10^{-4}
Noble	60	0.02	40	0.10	-	0.05	1.05	0.80	10^{-6}
	100	0.02	30	0.10	-	0.10	1.05	0.80	10^{-7}
	200	0.01	40	0.05	-	0.02	1.05	0.80	10^{-6}
	300	0.01	40	0.05	-	0.05	1.05	1.00	10^{-5}
	400	0.02	30	0.10	-	0.05	0.99	0.80	10^{-6}
	500	0.01	30	0.10	-	0.05	1.05	0.80	10^{-5}
Experimental Data	60	0.01	40	0.10	0.10	0.02	1.05	0.80	10^{-6}
	100	0.01	40	0.02	0.10	0.10	0.95	1.00	10^{-6}
	200	0.01	40	0.10	0.02	0.10	1.05	0.90	10^{-4}
	300	0.01	40	0.10	0.10	0.05	1.05	0.70	10^{-4}
	400	0.01	30	0.02	0.10	0.05	1.05	0.70	10^{-6}
	500	0.01	40	0.10	0.10	0.05	0.95	0.80	10^{-4}

* For the Noble dataset, stimulus information is not used so the input weight scale for the stimulus current is not applicable.

Table S5. Optimal hyperparameters found by grid search for the clustered ESN method for each dataset and network size. Hyperparameter definitions are given in Table S1.

Action potential	Network size	n_c	δ	τ	σ_{in}^1	σ_{in}^2	pr_c	pr	ρ	α	λ
Fenton-Karma	60	3	0.03	40	0.10	0.10	0.02	0.98	0.90	1.00	10^{-5}
	100	3	0.02	40	0.02	0.10	0.02	0.98	1.05	0.70	10^{-6}
	200	2	0.02	40	0.10	0.10	0.05	0.98	1.05	1.00	10^{-4}
	300	2	0.02	40	0.10	0.02	0.05	0.95	0.85	0.80	10^{-6}
	400	2	0.02	40	0.10	0.10	0.10	0.98	1.05	0.70	10^{-4}
	500	3	0.02	40	0.10	0.10	0.05	0.95	1.05	0.70	10^{-4}
Noble	60	2	0.02	40	0.10	-	0.05	0.98	0.90	0.80	10^{-7}
	100	4	0.01	30	0.10	-	0.02	0.95	0.99	0.90	10^{-7}
	200	4	0.01	40	0.10	-	0.02	0.95	0.99	0.90	10^{-7}
	300	2	0.01	40	0.10	-	0.02	0.90	0.99	0.90	10^{-7}
	400	3	0.01	40	0.10	-	0.02	0.95	0.99	0.80	10^{-7}
	500	4	0.01	40	0.10	-	0.05	0.95	0.99	0.90	10^{-7}
Experimental Data	60	3	0.01	40	0.02	0.10	0.02	0.98	1.05	0.90	10^{-5}
	100	3	0.01	40	0.10	0.02	0.02	0.98	1.05	0.70	10^{-6}
	200	3	0.01	40	0.10	0.10	0.10	0.98	1.05	0.70	10^{-6}
	300	2	0.01	40	0.10	0.02	0.10	0.98	1.05	0.80	10^{-6}
	400	3	0.01	40	0.10	0.10	0.02	0.98	1.05	0.70	10^{-5}
	500	2	0.01	40	0.10	0.10	0.10	0.95	1.05	1.00	10^{-6}

* For the Noble dataset, stimulus information is not used so the input weight scale for the stimulus current is not applicable.