




RESEARCH ARTICLE | JUNE 10 2022

A machine-learning approach for long-term prediction of experimental cardiac action potential time series using an autoencoder and echo state networks F FREE

Shahrokh Shahi ; Flavio H. Fenton ; Elizabeth M. Cherry 



Chaos 32, 063117 (2022)

<https://doi.org/10.1063/5.0087812>

 CHORUS



Articles You May Be Interested In

Inverse mechano-electrical reconstruction of cardiac excitation wave patterns from mechanical deformation using deep learning

Chaos (December 2020)

A hybrid wind speed forecasting model using stacked autoencoder and LSTM

J. Renewable Sustainable Energy (April 2020)

Phase autoencoder for limit-cycle oscillators

Chaos (June 2024)



Chaos

Special Topics Open for Submissions

[Learn More](#)

A machine-learning approach for long-term prediction of experimental cardiac action potential time series using an autoencoder and echo state networks

Cite as: Chaos 32, 063117 (2022); doi: 10.1063/5.0087812

Submitted: 9 February 2022 · Accepted: 24 May 2022 ·

Published Online: 10 June 2022



View Online



Export Citation



CrossMark

Shahrokh Shahi,^{1,a)}  Flavio H. Fenton,²  and Elizabeth M. Cherry¹ 

AFFILIATIONS

¹School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA

²School of Physics, Georgia Institute of Technology, Atlanta, Georgia 30332, USA

^{a)}Author to whom correspondence should be addressed: shahi@gatech.edu

ABSTRACT

Computational modeling and experimental/clinical prediction of the complex signals during cardiac arrhythmias have the potential to lead to new approaches for prevention and treatment. Machine-learning (ML) and deep-learning approaches can be used for time-series forecasting and have recently been applied to cardiac electrophysiology. While the high spatiotemporal nonlinearity of cardiac electrical dynamics has hindered application of these approaches, the fact that cardiac voltage time series are not random suggests that reliable and efficient ML methods have the potential to predict future action potentials. This work introduces and evaluates an integrated architecture in which a long short-term memory autoencoder (AE) is integrated into the echo state network (ESN) framework. In this approach, the AE learns a compressed representation of the input nonlinear time series. Then, the trained encoder serves as a feature-extraction component, feeding the learned features into the recurrent ESN reservoir. The proposed AE-ESN approach is evaluated using synthetic and experimental voltage time series from cardiac cells, which exhibit nonlinear and chaotic behavior. Compared to the baseline and physics-informed ESN approaches, the AE-ESN yields mean absolute errors in predicted voltage 6–14 times smaller when forecasting approximately 20 future action potentials for the datasets considered. The AE-ESN also demonstrates less sensitivity to algorithmic parameter settings. Furthermore, the representation provided by the feature-extraction component removes the requirement in previous work for explicitly introducing external stimulus currents, which may not be easily extracted from real-world datasets, as additional time series, thereby making the AE-ESN easier to apply to clinical data.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0087812>

Time series of cardiac electrical signals during arrhythmias can exhibit highly nonlinear and chaotic dynamical behavior, making forecasting of their dynamics challenging. However, estimates of future system states could facilitate the development of improved arrhythmia control and termination methods by allowing opportunities for early intervention. Data-driven methods have shown promise in forecasting tasks, such as these, but finding appropriate values for the many network parameters and hyperparameters involved is a difficult and time-consuming task. Here, we aim to decrease sensitivity to parameter values and to improve performance in a cardiac application of an echo state network (ESN) by incorporating an autoencoder (AE), which automatically extracts signal features to learn a compressed representation of the time

series. We show that the integration of an AE with an ESN can produce reliable and robust predictive models for long-term forecasting of experimentally recorded complex cardiac action potential time series.

I. INTRODUCTION

Cardiac cells exhibit a range of complex nonlinear behavior, particularly in how the transmembrane potential changes over time.^{1,2} Most cardiac cells are excitable systems, with a stable rest state and production of an action potential when a super-threshold stimulus is supplied, leading to propagation of electrical waves that

activate the heart's contraction in space.³ As the period of stimulation is decreased below a critical value, often, a bifurcation occurs in a state of alternans,^{4–6} in which action potentials alternate in duration and amplitude⁷ between long and short; further bifurcations to higher-order rhythms can also occur.⁸ In space, alternans can become very complex^{9–12} and frequently degenerates into the disorganized and potentially lethal state of fibrillation, characterized by one or more reentrant spiral or scroll waves of electrical activity rotating at frequencies several times faster than the heart's natural pacemaker-driven rhythm.^{13–15} A number of methods to control cardiac alternans, and, therefore, to prevent one important route to fibrillation, have been proposed;^{16–19,19,20} however, many factors make application of such methods difficult, including the speed with which alternans can transition to fibrillation.^{7,21} Accurate and rapid forecasting²² of future cardiac voltage states could provide crucial time to intervene before fibrillation develops.^{23–25} In this paper, we investigate how machine-learning (ML) techniques developed for time-series forecasting can be employed to help predict cardiac electrical dynamics efficiently and accurately.

In general, time series are a natural way to represent sequential and temporal data that are ubiquitous in a wide range of real-world applications, including biology,²⁶ climate science,²⁷ anomaly detection in computer networks²⁸ and social networks,²⁹ finance,^{30–34} and energy.^{35–37} Accordingly, time-series forecasting has remained an interesting, yet challenging, problem for decades. In recent years, advances in deep-learning approaches have provided functional alternatives that outperform traditional techniques in many time-series forecasting applications.

Recurrent neural networks (RNNs) and, in particular, the gated architectures, such as the long short-term memory networks (LSTMs) and the gated recurrent units (GRUs), have been demonstrated to be successful for predicting time series in many applications, such as voice recognition,³⁸ natural language processing,³⁹ and analyzing and forecasting market data.⁴⁰ However, they are not always successful in long-term prediction of complex time series representing the state of nonlinear and chaotic dynamical systems.²²

The reservoir computing (RC) framework was introduced as a learning paradigm derived from RNNs in which the hidden layer is composed of a set of hidden units connected to each other randomly and the training process remains limited to the output layer.^{41,42} Despite this considerable simplification resulting in a substantial reduction in the computational costs, these networks have shown to be very successful in forecasting nonlinear and chaotic time series.^{43–46} Echo state networks (ESNs) are a commonly used realization of an RC framework, and several variants of these networks have been proposed and evaluated.⁴⁵ Nonetheless, on account of the intrinsic randomness in constructing ESNs, obtaining reliable prediction results heavily depends on finding an appropriate set of hyperparameters and network parameters when forming the network, which can be cumbersome, and the added computational expense of searching for appropriate parameter values may degrade the overall efficiency of the approach. This sensitivity of ESNs to its hyperparameters and randomly chosen untrained parameters makes it unclear whether this approach will be useful for real-world applications. Still, the low-computational cost and promising results in forecasting chaotic time series have motivated further research to improve the reliability and robustness of ESNs.^{46,47}

We previously demonstrated for cardiac voltage time series²² that ESN performance can be highly affected by the time resolution of the input time series and that resampling can considerably improve the prediction ability of the ESN while reducing its sensitivity to the initial values. We also illustrated that an adaptive resampling, which selects a subset of points in the dataset designed to maintain adequate resolution in both time and the variable of interest, can effectively improve the prediction results. However, such a resampling approach needs expert supervision and tuning efforts to guarantee that the important features of the input time series are included in the resampled version.

These observations suggest the application of a feature-extraction method to learn an implicit representation of the input time series and circumvent the need for a manual resampling step. Accordingly, this work introduces an AE-ESN, an integration of an autoencoder (AE) into the ESN framework to predict the cardiac action potential time series, wherein an LSTM AE provides a feature-extraction component. We apply the integrated approach to forecasting cardiac action potential time series obtained *in silico* and in *ex vivo* experiments that exhibit nonlinear behaviors. We demonstrate that this technique provides higher prediction accuracy compared to the baseline ESN and a physics-informed variant of ESN, also known as hybrid ESN (HESN). In particular, the proposed AE-ESN approach improves prediction accuracy, with the ESN approaches achieving the mean absolute error (MAE) values between 6 and 14 times higher for the cases considered, and can capture the dynamics of the system described by the input time series, while its performance is almost independent of the initial network parameters.

The rest of this paper is organized as follows. Section II presents a background about ESNs and AEs and highlights the main features of their design. The details of the proposed method are discussed in Sec. III, and its performance in forecasting nonlinear synthetic and experimental time series of cardiac voltage is evaluated in Sec. IV. Finally, Sec. V presents some discussion and concluding remarks.

II. BACKGROUND

Below, we briefly review the main components of the ESNs and AEs.

A. Echo state networks

Echo state networks (ESNs) were introduced as a realization of the reservoir computing (RC) paradigm in which the recurrent layer is represented as a reservoir of randomly connected hidden units, and the training process remains limited to the weights of a linear memory-less readout layer. Consequently, ESNs offer a low-computational-cost approach for forecasting time series and have been employed to model and predict nonlinear time series for the last two decades.^{41,45}

Figure 1(a) illustrates the main components of an ESN: (i) an input layer accommodating the input time series, (ii) a randomly built reservoir of sparsely connected neurons, and (iii) a readout layer. The most commonly used variant of ESNs employs the “leaky” update model,⁴⁸ where the state update reads as

$$\mathbf{h}_t = (1 - \alpha)\mathbf{h}_{t-1} + \alpha \tanh(\mathbf{W}^{in}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}). \quad (1)$$

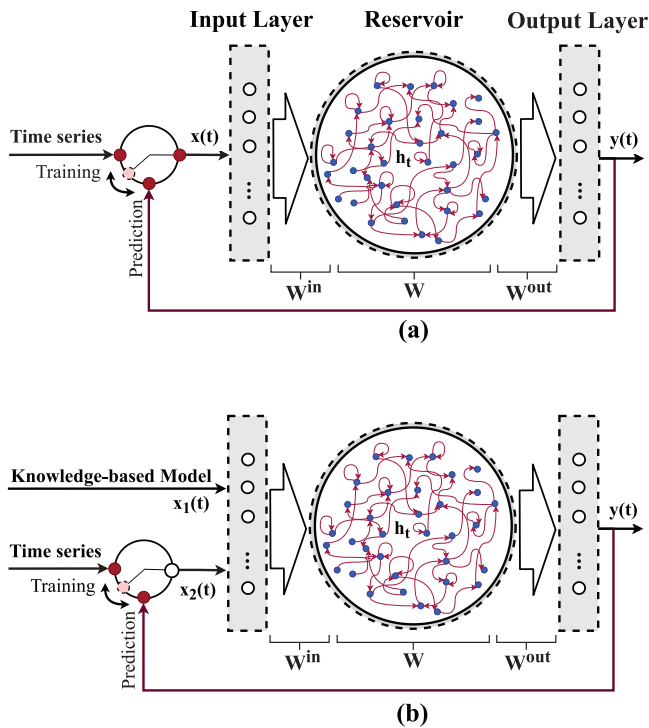


FIG. 1. The main components of (a) an ESN, including the input layer, reservoir of randomly connected neurons, and the output readout layer, and (b) an HESN with an additional input signal from the knowledge-based model.

In this formalism, W^{in} and W are randomly initialized matrices representing the input weight and reservoir weight matrices, respectively, and both remain untrained. The reservoir state at time step t is denoted by h_t , and $\alpha \in [0, 1]$ is defined as the leaking rate. The size of the input layer is determined by the number of input variables, which are denoted by x_t . Accordingly, the output of the network is calculated by the following equation:

$$y_t = f^{out}(W^{out}[x_t; h_t]), \quad (2)$$

where f^{out} denotes the output layer activation function, normally chosen here as a unity function. The output weights W^{out} are obtained by a regularized least-squares regression with Tikhonov regularization to avoid overfitting.

In the case of physical dynamical systems, domain knowledge can also be integrated into an ESN to construct a hybrid physics-informed variant of ESNs known as HESN. This approach has recently been demonstrated to be effective for forecasting nonlinear time series in a number of applications.^{22,49–51} HESNs add a knowledge-based mathematical model describing an approximation of the dynamical system as an additional input fed into the network to improve its prediction performance [see Fig. 1(b)]. Since this method has been shown to be successful for predicting cardiac action potentials,²² in this work, in addition to the baseline ESN, we also compare the obtained results with an HESN.

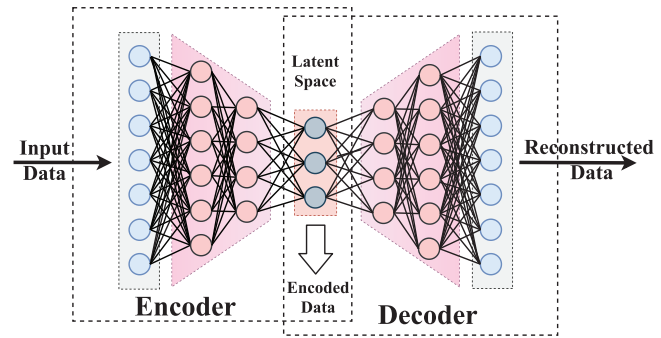


FIG. 2. The main components of an AE, including the encoder, the latent space also known as the bottleneck layer, and the decoder.

B. Autoencoder

AEs are unsupervised neural networks trained to reconstruct the input data, for which they aim to learn an approximation of the identity function by setting the target values equal to the inputs.⁵² Such architectures originally were introduced for internal representation learning and feature extraction,⁵³ but later, their applications were extended to anomaly detection,^{54,55} nonlinear dimensionality reduction,⁵⁶ and recommender systems.⁵⁷

As demonstrated in Fig. 2, an AE consists of three main components: (i) the encoder, which compresses the input into a latent space representation and provides a reduced representation of the input data; (ii) the latent space sometimes known as the bottleneck or the code layer, which represents the compressed version of the input; and (iii) the decoder, which decodes the encoded representation back to the original dimension. This purposefully designed structure enables AEs to automatically extract the high-level features of the input data by imposing certain constraints on the network, such as limiting the number of hidden units in the bottleneck layer.⁵⁸ The automatically learned features can then be used to improve the prediction ability in other ML techniques, particularly compared to cases where features are selected manually.^{59–61}

AEs normally are trained by applying the backpropagation approach to minimize the reconstruction error between the input data and the reconstructed data obtained at the output layer. This error is usually measured by a squared error or cross entropy. Accordingly, optimization techniques, such as stochastic gradient descent (SGD), can be employed to optimize the obtained loss function and train the network.

Several variants of AEs have been proposed to learn richer representations of the inputs, such as deep AEs,^{62,63} which take advantage of higher degrees of nonlinearity to extract the high-level features, and LSTM AEs, in which LSTM layers are employed for the encoding and decoding phases.^{64–66} LSTMs and gated RNNs, in general, are considered the most effective approaches for dealing with sequential data and forecasting time series,⁵² memory cells replace the usual hidden units in the network, making them successful in a wide range of practical applications.^{43,67} Accordingly, the LSTM AEs have been successfully employed in feature extraction and anomaly

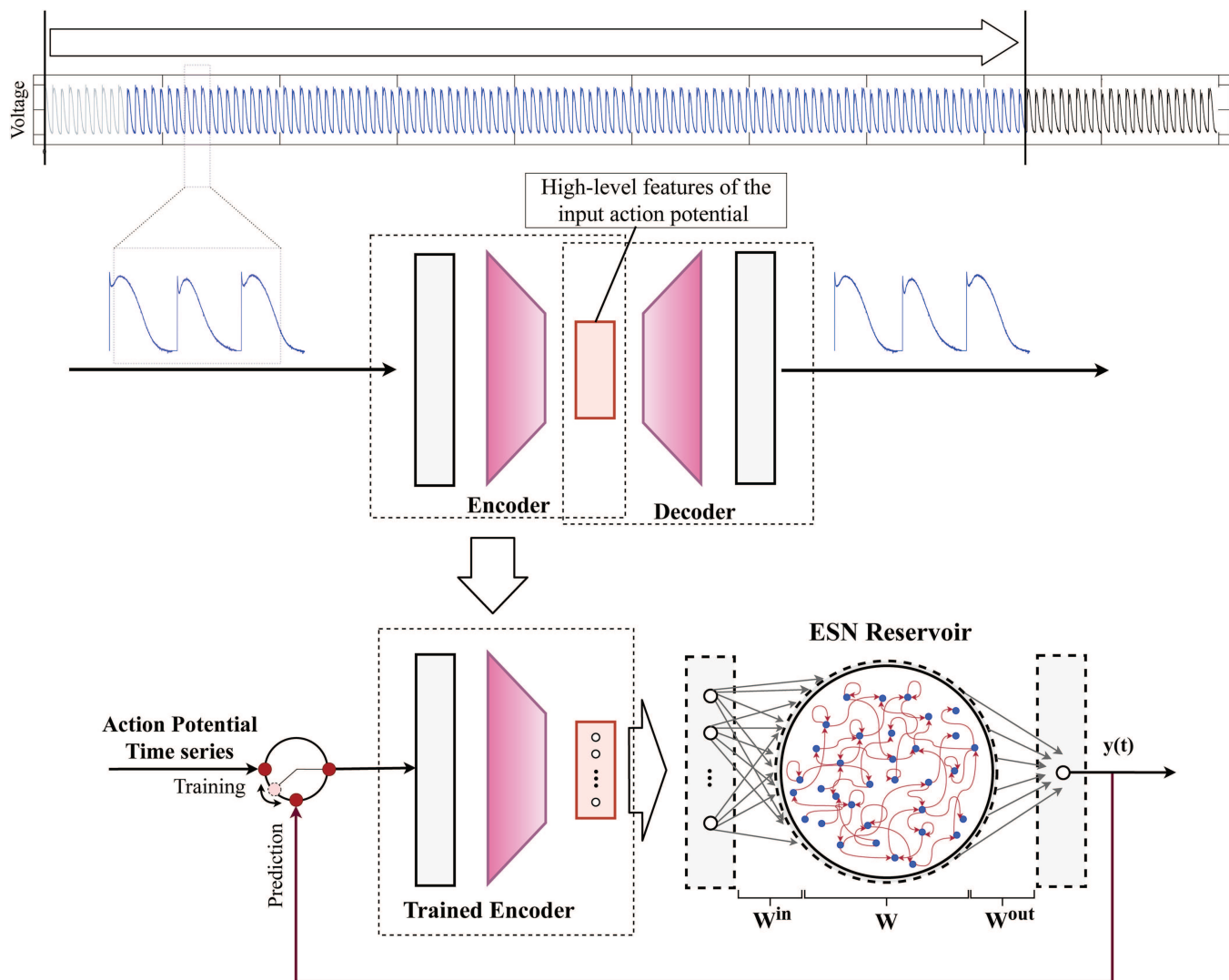


FIG. 3. Integrated architecture of the LSTM AE and ESN.

detection tasks in various real-world applications where time-series data are involved.^{68–71}

III. METHODS

A. Architecture

Conventional ESNs [see Fig. 1(a)] apply a complex nonlinear transformation to the input data mapping the given time series to a higher-dimensional space, where the reservoir of randomly connected neurons constructs a representation of the input data. Therefore, the size of the reservoir must be large enough to provide rich dynamics and to capture the behavior of the dynamical system represented by the input time series.⁷² However, the capabilities of ESNs can be obscured by the intrinsic randomness

in their construction, which makes them highly sensitive to the values of non-trainable parameters. This particularly becomes a problem when the input time series demonstrates highly nonlinear behavior.^{22,73}

To overcome these limitations and to achieve a better representation of the inputs, we introduce the AE-ESN approach in which an LSTM AE is integrated into the ESN framework. In our proposed technique, at first, an LSTM AE is trained using the training dataset to learn the higher-level features of the input data, and then the trained encoder serves as a feature-extraction layer located between the input layer and the recurrent reservoir. Thus, instead of the direct use of input data, the learned features are fed to the ESN reservoir. Figure 3 outlines the main features of the proposed approach.

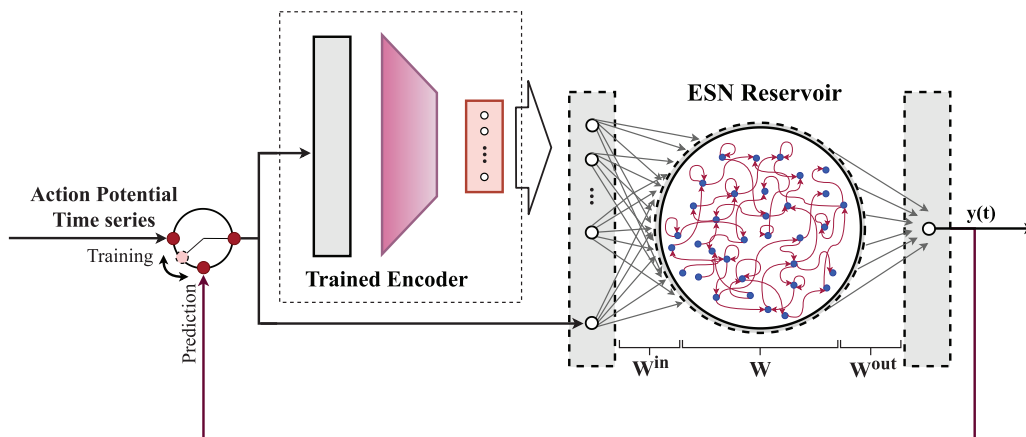


FIG. 4. Integrated architecture of the LSTM AE and ESN, where the input time series and the extracted features are input to the ESN.

It has also been demonstrated that the compressed representation learned by such unsupervised AEs can be employed as additional inputs to other supervised ML methods to improve prediction performance.^{52,74} Therefore, in this work, we also investigate augmenting the ESN inputs with the extracted features obtained in the AE latent space, as shown schematically in Fig. 4.

Network parameters for our implementations are selected through a hyperparameter optimization process, as described in Sec. III E, with specific values given in Tables S1–S3 of the supplementary material.

B. Training

The optimum values of weights and biases of the AE and the weights of the readout layer in ESN are determined during the training process, which includes two main steps. First, an AE is constructed by stacking LSTM layers in the encoder and decoder parts to form a symmetrical architecture. This model is trained by a back-propagation approach using the training data. Once the model is trained, the reconstruction part of the network can be discarded, and only the trained encoder is connected to the ESN reservoir. The output of such a trained AE at the bottleneck provides a fixed-length vector, which can be interpreted as the compressed representation of the input data and is treated as an additional input to the ESN.

Afterward, the weights of the readout layer, i.e., W^{out} , are obtained using a regularized linear regression method. For this purpose, the switch in Fig. 4 is set to the “Training” mode. Then, the training dataset is input to the network, one value at a time, and the states of the reservoir hidden units are recorded in an n -by- d state matrix H , where n is the reservoir size and d is the number of training time steps. Also, the desired output values are stored in a vector y . Accordingly, the readout weights can be obtained in one step using the Moore–Penrose generalized matrix inverse or the pseudo-inverse of the matrix H as follows:

$$W^{out} = (H^T H + \lambda I)^{-1} H^T y, \tag{3}$$

where λ is the regularization factor in the ridge regression method and I denotes the identity matrix.

One important consideration in the ESN training phase is discarding the states of the initial steps of the network. This is crucial to guarantee that the initial states of the reservoir are washed out and the state matrix only includes the state of the system describing its dynamics. In this work, we assign the beginning of each time series containing the first ten action potentials as the pretrain period, where the network states are discarded and do not contribute in the training process (see Figs. 5–7).

More information about the hyperparameter selection process is included in Sec. III E; specific values for the hyperparameters can be found in Tables S1–S3 in the supplementary material.

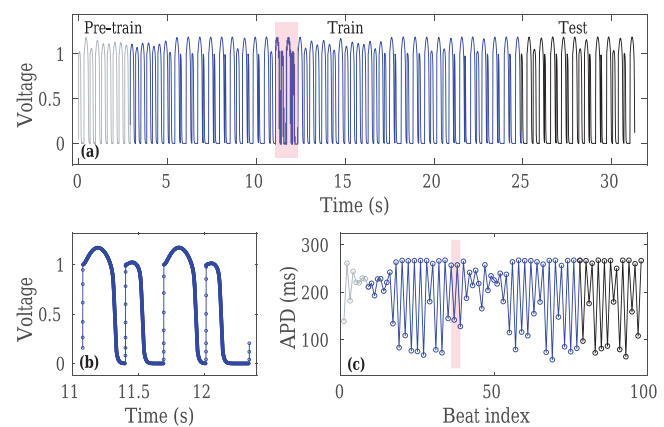


FIG. 5. Action potential time series generated by the Fenton–Karma model with the random pacing approach described in the text. (a) Voltage time series, including pre-training data (gray), training dataset (blue), and testing dataset (black). (b) Voltage trace corresponding to the shaded region in panels (a) and (c) illustrating the variation in the action potential shapes and durations. (c) Corresponding APDs following the same color scheme as panel (a).

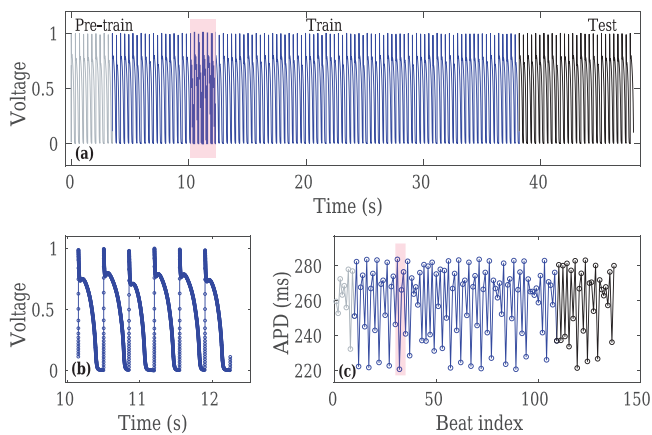


FIG. 6. Action potential time series generated by the Beeler–Reuter model. (a) Voltage time series, including pre-training data (gray), training dataset (blue), and testing dataset (black). (b) Voltage trace corresponding to the shaded region in panels (a) and (c) illustrating the variation in action potential shapes and durations. (c) Corresponding APDs following the same color scheme as panel (a).

C. Prediction

For the prediction phase, when the fully trained model is employed for forecasting the future values, the switch in Fig. 4 is set to the “Prediction” mode. In this way, once the readout layer is trained, a recursive approach performs multi-step prediction into the future, where the prediction results at each time step are provided as the input for the next time step.

D. Datasets

The effectiveness of the proposed approach is evaluated by predicting three cardiac action potential time series, the characteristics of which are described in this section.

1. Dataset 1: Fenton–Karma

The first dataset²² is a randomly timed action potential time series generated using the Fenton–Karma (FK) model,⁷⁵ including a voltage variable and two gating variables. The Beeler–Reuter fitting of the FK model, where the parameters are set to parameter set 3 in Ref. 76, is used, and the forward Euler method with a fixed time step of 0.1 ms is employed to solve the corresponding differential equations. To generate complex dynamics, the cycle lengths are drawn from a normal distribution centered at 320 ms with a standard deviation of 50 ms, where a 2-ms stimulus voltage is applied with a relative magnitude of 0.2 at the beginning of each cycle. Cycle length values both below and above the bifurcation point, beyond which alternans would be generated for a fixed cycle length, are included to allow a wide range of APD values (about 200 ms). Figure 5 demonstrates (a) the resulting voltage trace and (b) the corresponding action potential duration (APD) values. The training dataset (blue) includes nearly 80 action potentials, while the testing dataset (black) includes about 20 action potentials.

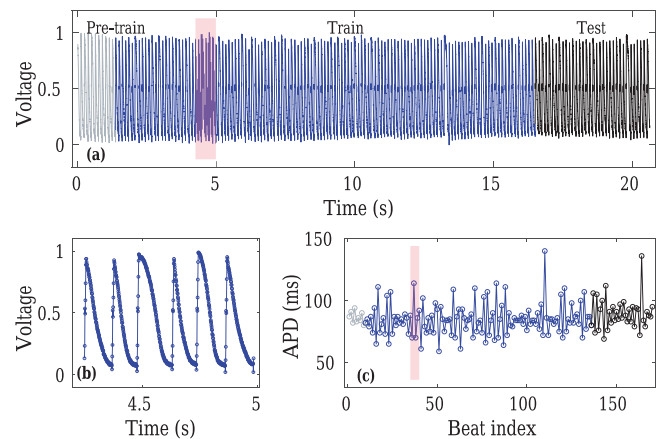


FIG. 7. Experimental action potential time series obtained as a microelectrode recording from a zebrafish heart. (a) Voltage time series, including pre-training data (gray), training dataset (blue), and testing dataset (black). (b) Voltage trace corresponding to the shaded region in panels (a) and (c) illustrating the variation in action potential shapes and durations. (c) Corresponding APDs following the same color scheme as panel (a).

2. Dataset 2: Beeler–Reuter

The second dataset is generated using a version of the Beeler–Reuter (BR) ventricular cell model⁷⁷ modified to generate chaotic signals via a nonmonotonic APD restitution curve.⁷⁸ The BR model consists of a voltage variable, six gating variables, and the intracellular calcium concentration, with nonmonotonic restitution achieved by blocking one of the potassium currents (for more detail, see Ref. 78). The cycle length is set to 350 ms, and a 2-ms stimulus voltage is applied with a relative magnitude of 0.2 at the beginning of each cycle. The corresponding differential equations are solved with a fixed time step of 0.1 ms using forward Euler for the voltage and calcium variables and backward Euler for the gating variables to improve stability. Figure 6 illustrates the generated time series consisting of about 140 beats, where the first 10 beats are used in the pre-training phase (gray), 102 beats for training the models (blue), and the remaining 28 beats for the testing dataset (black). Note that the time series is linearly scaled to be within [0, 1].

3. Dataset 3: Experimental data

The third dataset is a microelectrode recording of voltage recorded from a zebrafish heart as described in Ref. 22; it has been normalized to lie within the interval [0,1]. Note that the zebrafish heart is tiny (~ 1 mm), and we have not observed any meaningful spatiotemporal dynamics in these hearts. In addition, microelectrode recordings reflect the activity of a single cell, even though the cell is in a tissue. Thus, we believe that it is appropriate not to consider spatial effects in the zebrafish heart. Figure 7 illustrates the experimental dataset consisting of about 170 beats, where the first 10 beats are used in the pre-training phase (gray), 125 beats for training the models (blue), and the remaining 35 beats for the testing dataset (black).

E. Implementation

The proposed approach has been implemented in MATLAB (R2021a), where the AE component is built on top of the Deep Learning Toolbox. To construct the AEs, the LSTM layers are generated using the `lstmLayer` class and stacked, while dropout layers are interleaved with them to avoid overfitting issues. After training the AE in the first step of training (see Sec. III B), to connect the trained encoder to the ESN, a fully connected layer with the same size as the bottleneck layer is added at the end to transfer the extracted features to the ESN reservoir.

For all datasets, the architecture of the adopted AE-ESN model is similar to Fig. 4, with the extracted features fed to the reservoir along with the input action potential time series. In our initial experiments, we found this topology more effective compared to the other integrated architecture illustrated in Fig. 3; see the [supplementary material](#) for more details.

To obtain the optimum values of the hyperparameters required to build the AE-ESN, along with the ESN and HESN models, for each dataset, an extensive grid search was conducted, where the set of searched values and the optima found are presented in Tables S1–S3 in the [supplementary material](#); the selected optimum values are highlighted in bold.

In comparison with the standard RC techniques, training an LSTM AE is computationally expensive and involves many hyperparameters, including the number of hidden units and hidden layers; the maximum number of training epochs; and the optimizer used for training and its corresponding hyperparameters, such as the regularization factor, the learning rate, and the learning rate drop factor. Therefore, conducting a full grid search on all these parameters in the same way we did for the ESN and HESN parameters is not practically feasible. Accordingly, in this work, we limit our grid search to finding the optimum values of the most important hyperparameters, i.e., the number of hidden units and hidden layers and the learning rate (see Tables S1–S3 in the [supplementary material](#)), while the rest of the hyperparameters are set to the best found values in our initial experiments. Toward this end, we used the Adam optimizer to train the network with the corresponding default training configurations in MATLAB. The maximum number of training epochs is set to 50 with early-stopping validation patience set to 4 to avoid overfitting. We also used dropout layers with the probability of 0.2 interleaving the LSTM layers for further overfitting avoidance in the AE.

IV. RESULTS

We apply the proposed technique to predict three time series of cardiac action potentials. The results obtained using the AE-ESN method are compared with baseline ESN and HESN approaches, whose details have been discussed in our previous work;^{22,79} a summary of the configuration used is presented in the [supplementary material](#).

One important consideration about this comparison is that cardiac cells in experiments generally are stimulated exogenously so that in data-driven prediction of action potential time series, the pacing stimulus must also be introduced to the predictive model as an additional input signal (see Fig. S1 in the [supplementary material](#)). In contrast, in the proposed AE-ESN approach, the trained encoder handles the feature extraction task automatically

and provides the ESN part with the required pacing information. Therefore, the AE-ESN results are obtained without feeding explicit pacing information to the network and without manual re-sampling of the input action potential time series, as opposed to the ESN and HESN approaches.

A. Dataset 1: Fenton-Karma

Figures 8(a)–8(c) show the 19 action potentials predicted by the AE-ESN approach along with those from the ESN and HESN approaches. Compared to the ESN and HESN techniques, the voltage values predicted by the proposed AE-ESN approach are more accurate in terms of the absolute metric demonstrated in Fig. 8(d), which also shows that there is no appreciable growth in error over time (the slopes of linear fits to the prediction error values over time are 0.0029, 0.0023, and 0.0005 for ESN, HESN, and AE-ESN, respectively). Figures 8(e)–8(g) illustrate the distributions of the absolute prediction error for each method and show that the mean absolute error (MAE) for the AE-ESN (0.008) is smaller than the MAE values for the ESN (0.046) and HESN (0.055) by factors of 5.8 and 6.9, respectively. Additionally, the predicted APD values in Fig. 9 demonstrate that all three methods can accurately predict the APD values, with the MAE of the predicted APDs obtained by the AE-ESN the smallest by factors of 1.7 and 1.9 compared to the ESN and HESN, respectively.

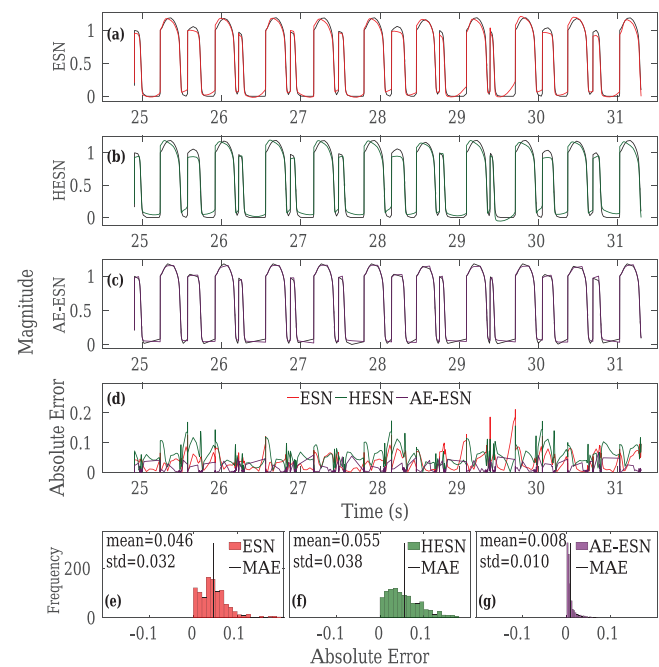


FIG. 8. FK voltage values predicted by (a) ESN, (b) HESN, and (c) AE-ESN for about 19 action potentials. Target values are plotted in black for comparison. Panel (d) shows the absolute error values at each time step for the three methods, and panels (e)–(g) show histograms of error values for each method along with the corresponding means and standard deviations.

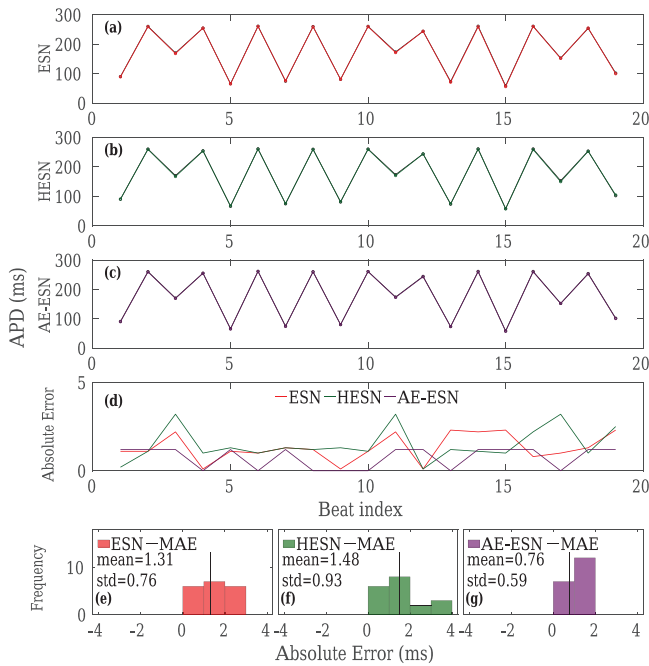


FIG. 9. APD prediction results for the FK dataset obtained by the (a) ESN, (b) HESN, and (c) AE-ESN for about 19 heart beats. Target values are plotted in black. Panel (d) shows the absolute error values for each APD, and panels (e)–(g) show histograms of error values for each method along with the corresponding means and standard deviations.

Note that the features extracted by the autoencoder can be difficult to interpret. Figure S2 in the [supplementary material](#) shows the output of the encoder for the FK dataset. Although it is not obvious why these features were selected, together, they are used to form a good approximation of the system, and basic properties, such as action potential timing and shapes, are recognizable in many of the output signals.

One major drawback with ESN techniques is the sensitivity of the prediction performance to the initial values of the model parameters and the initialization technique. Therefore, after constructing an ESN, extra effort is required to find an appropriate set of parameters, which usually consists of an extensive grid search. In contrast, AE-ESN is less sensitive to the initialization, and once the model is created, the initialization does not affect the prediction performance. To evaluate this property, each of the three compared models was run with 100 various random initializations, and the ranges of the predicted values are illustrated as shaded regions in [Fig. 10](#). In the ESN approach, the range of the predicted values obtained by different initializations is narrow for only the first three beats and start to grow over longer times. For instance, at the end of the predicted 19 beats, the prediction results differ from the ground truth by more than 300%. In contrast, although the range of the HESN prediction results started to grow after the first two beats, this range remains limited for a longer period of time, indicating that the knowledge-based model can reduce variation in the prediction values. Finally,

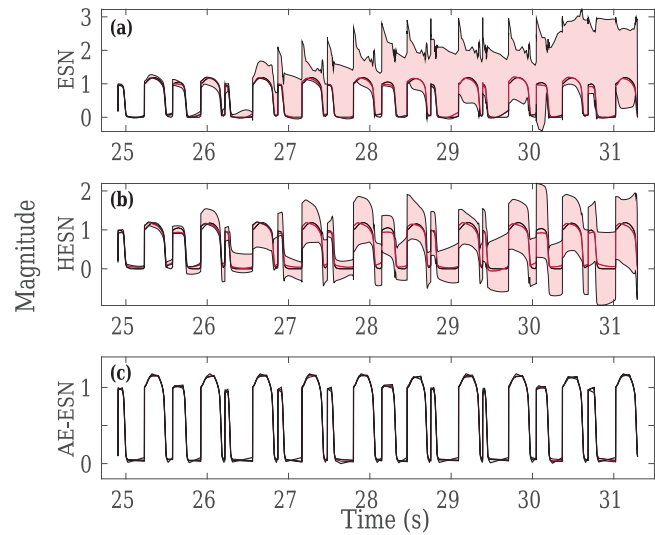


FIG. 10. Range of the predicted voltage values of the FK dataset obtained by 100 different randomizations using the (a) ESN, (b) HESN, and (c) AE-ESN approaches. Maximum and minimum voltage values obtained across the 100 trials are shown to outline the shaded regions.

the bottom panel in [Fig. 10](#) demonstrates the range of prediction results obtained by the proposed AE-ESN approach with 100 random initial values, which reveals that the range of predicted values remains narrow and close to the target values for the entire test dataset. This finding indicates that even though the reservoir and the associated input weights are randomly constructed, taking advantage of the extracted features can reduce the sensitivity of the method to the initial network parameter values.

B. Dataset 2: Beeler-Reuter

Predicting the action potentials for the BR dataset yields results similar to those found for the FK dataset. [Figures 11\(a\)–11\(c\)](#) show the voltage values across the roughly 28 action potentials of the dataset. Good fittings are obtained in all cases, and the error does not grow appreciably over time, as can be seen in [Fig. 11\(d\)](#) (linear fit slopes: 0.000 10 for ESN, 0.000 22 for HESN, and 0.000 02 for AE-ESN). Nevertheless, the AE-ESN approach again produces the highest accuracy. [Figures 11\(e\)–11\(g\)](#) show that the MAE obtained is smaller by an order of magnitude for the AE-ESN approach compared to the ESN and HESN methods (0.002 vs 0.028 and 0.026 for ESN and HESN, respectively). In addition, the AE-ESN has the smallest standard deviation, indicating that the low MAE values occur consistently across the time series. Because of the low error across the methods, the APDs can be predicted with little error for all the methods, as shown in [Fig. 12](#). Still, the MAE for APDs for the AE-ESN approach is the smallest of the errors for the three methods, with the HESN outperforming the ESN.

The encoder output for the AE-ESN approach, shown in [Fig. S3](#) of the [supplementary material](#), displays features that appear similar in general to the same type of output for the FK dataset. The signals

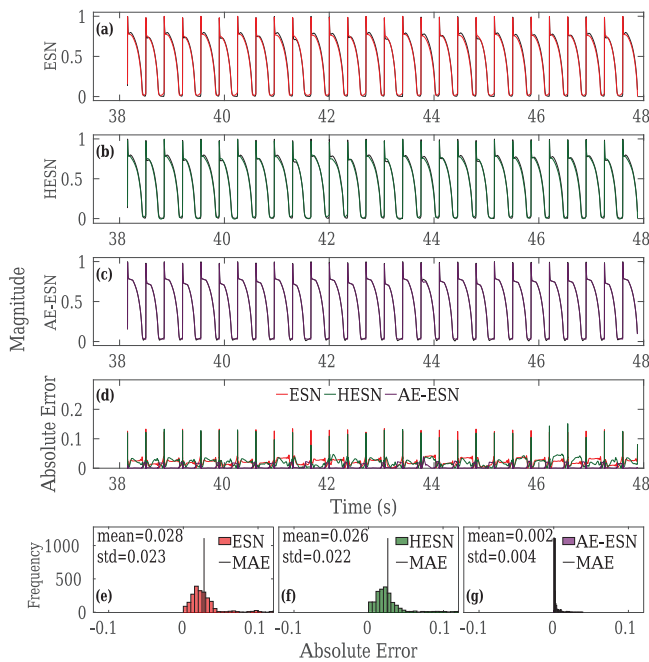


FIG. 11. BR voltage values predicted by (a) ESN, (b) HESN, and (c) AE-ESN for about 28 action potentials. Target values are plotted in black for comparison. Panel (d) shows the absolute error values at each time step, and panels (e)–(g) show histograms of error values for each method along with the corresponding means and standard deviations.

again show the timing and shapes of the action potentials in the time series. Note that the optimal hyperparameters for this dataset lead to a smaller number of layers and cells for the bottleneck layer so that only 16 signals are output.

Figure 13 demonstrates the sensitivity of the methods to the random initialization by displaying the range of output at each time point across 100 different random seed numbers. This dataset appears much less sensitive to initialization effects than the FK dataset such that all three methods have noticeably smaller variability. Nevertheless, the variability in the AE-ESN predictions is smaller than the variability for both the ESN and HESN approaches, each of which demonstrates more variability within the first few action potentials than the AE-ESN demonstrates over the entire time series.

C. Dataset 3: Experimental data

The constructed models were applied to predict the experimental dataset. As can be seen in Fig. 14, the prediction accuracy of the AE-ESN measured by the absolute error is higher (lower error) than for the ESN and HESN techniques (mean: 0.010 vs 0.059 and 0.057 for ESN and HESN, respectively). The linear fit slopes (0.0087 for ESN, -0.0003 for HESN, and 0.0006 for AE-ESN) reveal that the error does not grow over time for the predicted test set. The same pattern can be observed in the APDs shown in Fig. 15, where the absolute errors of the predicted values obtained by the AE-ESN

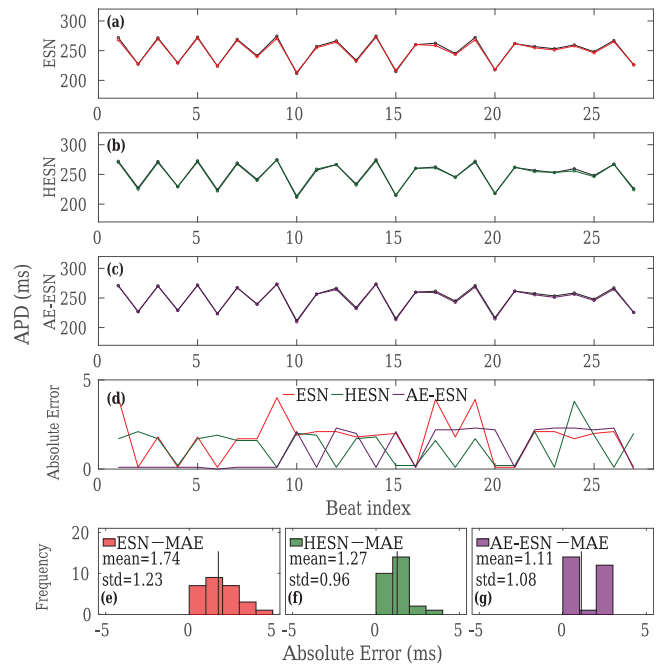


FIG. 12. APD prediction results for the BR dataset obtained by (a) ESN, (b) HESN, and (c) AE-ESN for about 28 action potentials. Target values are plotted in black. Panel (d) shows the absolute error values at each beat, and panels (e)–(g) show histograms of error values for each method along with the corresponding means and standard deviations.

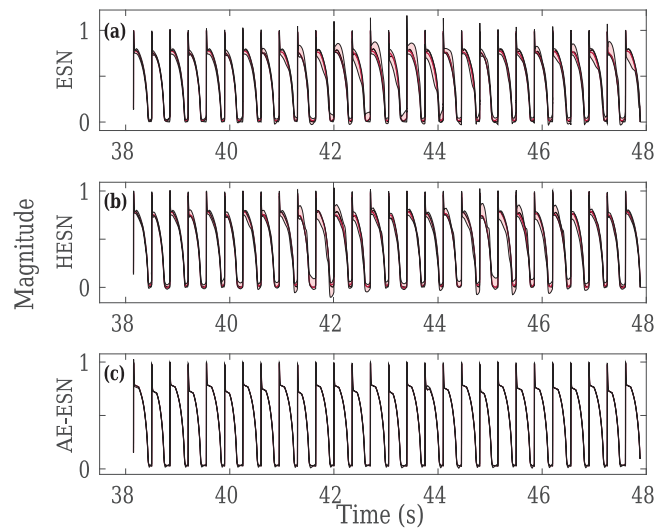


FIG. 13. Range of the predicted voltage values of the BR dataset obtained by 100 different randomizations using the (a) ESN, (b) HESN, and (c) AE-ESN approaches. Maximum and minimum voltage values obtained across the 100 trials are shown to outline the shaded regions.

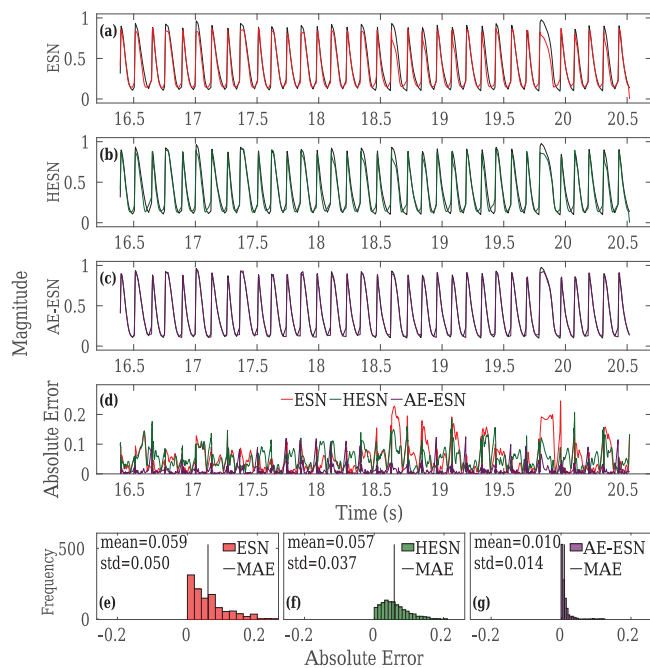


FIG. 14. Experimental voltage values predicted by (a) ESN, (b) HESN, and (c) AE-ESN for about 19 action potentials. Target values are plotted in black for comparison. Panel (d) shows the absolute error values at each time step, and panels (e)–(g) show histograms of error values for each method along with the corresponding means and standard deviations.

approach are consistently less than the absolute errors of the predicted values by the ESN and HESN for almost the entire test dataset. Figure S4 in the [supplementary material](#) shows the trained encoder output, which is similar to what was seen for the other datasets.

To compare the sensitivity of the proposed method to the random initialization, the models were trained with 100 different random seed numbers and the range of the outputs is presented as the shaded regions in [Fig. 16](#). The highest variability occurs for the ESN approach, where the variability increases further into the future, and the range of predicted values is narrow only for the first two action potentials. The ranges of the predicted values are narrower for the HESN results, reflecting the controlling effect of the knowledge-based model integrated into the network. The best results occur for the proposed AE-ESN approach, where the range of the obtained results is very narrow and close to the target values for the entire test dataset.

D. Sensitivity of results

Each of the methods discussed requires choosing a number of network settings and parameters. As is typical in machine learning, we perform an optimization procedure to determine the values of these parameters and obtain the prediction results using those values. Using different settings thus would affect the quality of the prediction. It would be desirable for the prediction method to be insensitive to key hyperparameters so that the prediction quality

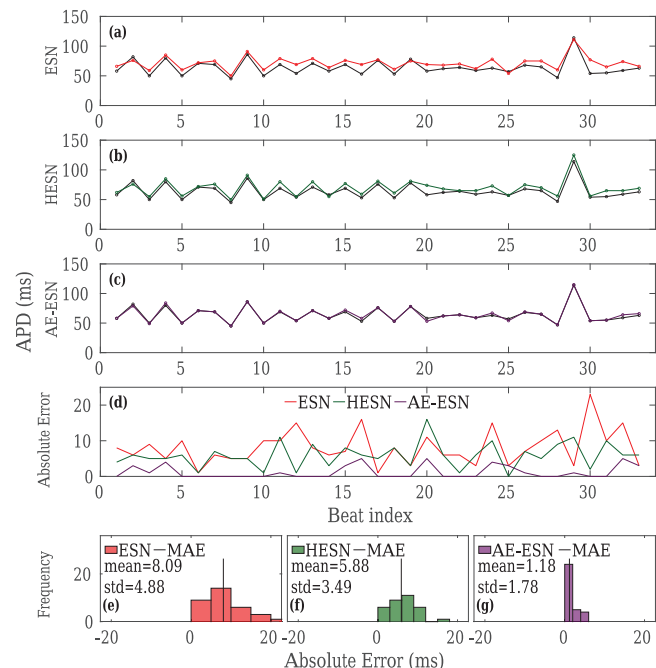


FIG. 15. APD prediction results for the experimental dataset obtained by (a) ESN, (b) HESN, and (c) AE-ESN for about 19 action potentials. Target values are plotted in black. Panel (d) shows the absolute error values at each beat, and panels (e)–(g) show histograms of error values for each method along with the corresponding means and standard deviations.

would not depend strongly on the selection of these parameters. In this section, we demonstrate the sensitivity of the predictions to several important algorithmic parameters.

Some of the main parameters important for ESN approaches are the number of reservoir hidden units, the connection probability, the leaking rate, and the spectral radius. Figures S5 and S6 in the [supplementary material](#) show the effects on the mean absolute prediction error of separately varying these four parameters up to 20% for the FK dataset by the ESN and AE-ESN, respectively. For the ESN, the predictions appear to be least sensitive to the leaking rate, but even in this case, only moderate changes of 5–10% can yield a substantial decrease in accuracy. For the other parameters, the ESN is highly sensitive and even moderate changes often lead to results with an unacceptably large error. In contrast, the results from the AE-ESN maintain low MAE values below 0.01 except in one case where it is still below 0.015.

For the BR dataset, MAE values remain higher for the ESN predictions than for the AE-ESN predictions, as shown in Figs. S7 and S8 of the [supplementary material](#). Although the MAE values for the ESN predictions are not as high as for the FK dataset, the AE-ESN achieves much smaller MAE values below 0.005 in all cases and roughly an order of magnitude smaller than the ESN MAE values. Among the four parameters considered, the ESN demonstrates the greatest sensitivity to the spectral radius. The experimental dataset shows results similar to those obtained for the BR dataset, with the

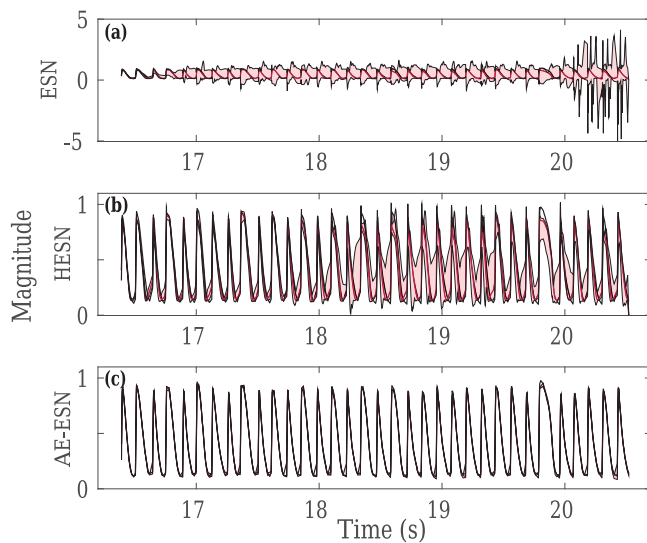


FIG. 16. Range of the predicted voltage values of the experimental dataset obtained by 100 different randomizations using the (a) ESN, (b) HESN, and (c) AE-ESN approaches. Maximum and minimum voltage values obtained across the 100 trials are shown to outline the shaded regions.

ESN most sensitive to the number of hidden units and the connection probability; see Figs. S9 and S10 in the [supplementary material](#). The MAE using the AE-ESN approach also remains below 0.015 and about an order of magnitude lower than the ESN error values. Thus, the AE-ESN, in addition to achieving greater accuracy, also reduces sensitivity to network parameters.

Another important parameter is the time resolution used in the input signals. For the ESN and HESN methods, the signals are resampled according to a parameter included during the optimization process; therefore, this discussion is focused on results using the AE-ESN, which uses a fixed temporal resolution. Figures S11 and S12 in the [supplementary material](#) show the results of changing the temporal resolution of the FK dataset from its baseline [shown in panel (b) in both cases], which includes about nine points to resolve the upstrokes; specifically, this is by increasing resolution by a factor of 2 [panel (a)] and decreasing resolution by factors of 2.5, 5, 10, and 20 [panels (c)–(f)]. The MAE increases as the resolution of the training set decreases; it scales by about a factor of -1.5 with the number of points in the training set. Visibly, the action potentials appear nearly identical even up to a temporal resolution of the dataset that is five times coarser.

V. DISCUSSION AND CONCLUSION

In this work, we have proposed a novel integrated architecture in which an LSTM AE is integrated into the ESN framework, with the trained encoder serving as a feature-extraction layer feeding the extracted feature into the recurrent reservoir in the ESN. We tested this approach against both synthetic and experimental datasets; in all cases, the predictions across about 20 action potentials

show a considerable improvement in both robustness and prediction performance when using the AE-ESN compared to the ESN or HESN approaches, as shown in Fig. S13 in the [supplementary material](#). In particular, prediction accuracy for the voltage is 6–14 times lower when using the AE-ESN in terms of MAE. Notably, the best improvement occurred for the experimental data case, with voltage prediction errors differing by factors of 14 and 13 for the ESN and HESN methods, respectively, when compared to the AE-ESN. We also evaluated the sensitivity of the proposed approach to the initial network parameter values and demonstrated that, in contrast to the ESN and HESN approaches, the results obtained by the AE-ESN technique show minimal dependence on the initial values as well as on the network parameters.

Understanding the source of the considerable improvements achieved by integrating an AE into the ESN framework requires considering whether a standalone LSTM AE without an ESN may also be capable of capturing the dynamics of the system and provide reliable predictions. However, the application of LSTM AEs for forecasting both synthetic and experimental action potential time series resulted in poor prediction performance, revealing that a single AE is incapable of forecasting such highly nonlinear time series. These poor prediction results are not presented in Sec. IV to avoid inconsistency and dramatic changes in the plot scales. This finding demonstrates that the combination of an LSTM AE with an ESN overcomes the poor prediction of complex cardiac time series of the former and the high sensitivity to initial parameters of the latter to produce a robust and effective approach for forecasting these nonlinear time series.

The requirement for training an LSTM AE increases the computational time needed for this approach, which reduces the benefit of the fast training process and low-computational cost characteristics typical of ESNs. A fair comparison of the computational time was not possible in this study since the AEs were trained using a GPU, while the rest of the computations were handled by the CPU. Still, training the AE was the most computationally expensive part of the process in the AE-ESN approach; depending on the complexity of the AE architecture and the size of the training dataset, it could take 1–2 orders of magnitude longer than the rest of the process in terms of wall-clock time for the three evaluated test cases. However, the reduced sensitivity of the AE-ESN to network parameters potentially could offset the increased computational time of the AE by allowing use of a lower-resolution grid search in comparison with baseline ESNs.

An importation limitation of the methods used in this study is that they are purely data-driven and thus are agnostic to the source of the time series. As a result, we cannot predict how well the methods will generalize to other datasets, including time series obtained under other conditions, such as different pacing protocols; synthetic datasets generated from the same model using different parameters or from another model; or experimental datasets recorded from the same experimental preparation at a different time or location, different preparations from the same species, or different species. For the same reason, the best performance is obtained by optimizing hyperparameters and performing training anew for each time series and/or change to untrained network parameters, which imposes a high computational cost at present if prediction is to be performed multiple times. It is an area of active research to understand how

optimization and training for one dataset may be used to facilitate the same tasks for a different dataset that has some similarities.^{80,81} Such transfer learning eventually may decrease computational costs, for example, by identifying how optimized hyperparameter values obtained for one dataset may be used to decrease the grid search time.

We note that the AE-ESN accuracy improvements are more modest when predicting APD rather than a more detailed prediction of voltage (and thus action potential shape), especially for the synthetic datasets. Given the increased computational costs of the AE-ESN approach, the simpler ESN or HESN may be a useful strategy when predicting only APD values, but for experimental data, the higher accuracy achieved by the AE-ESN may justify the extra cost.

In terms of parameter sensitivity across the three datasets, the main conclusion for the ESN was that the accuracy did not depend strongly on the leaking rate. For the other parameters, changes to the values resulted in poor accuracy for at least one but not all datasets considered. No clear conclusions could be drawn regarding the parameters to which the AE-ESN was most sensitive, as the error changed little within the parameter variations considered.

For the HESN, sometimes unphysiological values, including negative voltage values, can be obtained depending on the initialization. In this approach, there is no guarantee for the output to remain within the expected physical range since the domain knowledge is integrated as an additional time series fed to the network to promote the training process and does not directly impose constraints.

It should also be noted that in this study, we considered a limited number of action potential datasets; different results may be obtained for different time series representing different dynamical behavior. As such, although we would expect that the same methods could be used for datasets from hearts of different species, including human hearts, further study is needed to identify characteristics needed for successful long-term prediction. Furthermore, we considered a limited number of hyperparameters in the grid search process, and it is possible that different sets of hyperparameter values could generate different results.

SUPPLEMENTARY MATERIAL

Additional information regarding Methods and Results is available in the [supplementary material](#), as indicated in the text.

ACKNOWLEDGMENTS

This study was supported by NSF grants (Nos. CMMI-2011280 and CMMI-1762553) and the NIH grant (No. 1R01HL143450). We thank Conner J. Herndon for assistance in providing the experimental dataset.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Shahrokh Shahi: Conceptualization (equal); Methodology (equal); Software (equal); Visualization (equal); Writing – original

draft (equal). **Flavio H. Fenton:** Data curation (equal); Funding acquisition (equal); Investigation (equal); Resources (equal); Supervision (equal). **Elizabeth M. Cherry:** Conceptualization (equal); Funding acquisition (equal); Methodology (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- 1 A. Karma and R. F. Gilmour, Jr., “Nonlinear dynamics of heart rhythm disorders,” *Phys. Today* **60**(3), 51–57 (2007).
- 2 E. M. Cherry, F. H. Fenton, and R. F. Gilmour, Jr., “Mechanisms of ventricular arrhythmias: A dynamical systems-based perspective,” *Am. J. Physiol. Heart Circ. Physiol.* **302**, H2451–H2463 (2012).
- 3 V. Kappadan, S. Telele, I. Uzelac, F. Fenton, U. Parlitz, S. Luther, and J. Christoph, “High-resolution optical measurement of cardiac restitution, contraction, and fibrillation dynamics in beating vs blebbistatin-uncoupled isolated rabbit hearts,” *Front. Physiol.* **11**, 464 (2020).
- 4 J. B. Nolasco and R. W. Dahlen, “A graphic method for the study of alternation in cardiac action potentials,” *J. Appl. Physiol.* **25**, 191–196 (1968).
- 5 M. R. Guevara, G. Ward, A. Shrier, and L. Glass, “Electrical alternans and period-doubling bifurcations,” *Comput. Cardiol.* **11**, 167–170 (1984).
- 6 A. Karma, “Electrical alternans and spiral wave breakup in cardiac tissue,” *Chaos* **4**, 461–472 (1994).
- 7 D. D. Chen, R. A. Gray, I. Uzelac, C. Herndon, and F. H. Fenton, “Mechanism for amplitude alternans in electrocardiograms and the initiation of spatiotemporal chaos,” *Phys. Rev. Lett.* **118**, 168101 (2017).
- 8 A. Gizzi, E. M. Cherry, R. F. Gilmour, Jr., S. Luther, S. Filippi, and F. H. Fenton, “Effects of pacing site and stimulation history on alternans dynamics and the development of complex spatiotemporal patterns in cardiac tissue,” *Front. Card. Electrophysiol.* **4**, 71 (2013).
- 9 J. M. Pastore, S. D. Girouard, K. R. Laurita, F. G. Akar, and D. S. Rosenbaum, “Mechanism linking T-wave alternans to the genesis of cardiac fibrillation,” *Circulation* **99**, 1385–1394 (1999).
- 10 M. A. Watanabe, F. H. Fenton, S. J. Evans, H. M. Hastings, and A. Karma, “Mechanisms for discordant alternans,” *J. Cardiovasc. Electrophysiol.* **12**, 196–206 (2001).
- 11 E. M. Cherry and F. H. Fenton, “Suppression of alternans and conduction blocks despite steep APD restitution: Electrotonic, memory, and conduction velocity restitution effects,” *Am. J. Physiol. Heart Circ. Physiol.* **286**, H2332–H2341 (2004).
- 12 I. Uzelac, Y. C. Ji, D. Hornung, J. Schröder-Scheteling, S. Luther, R. A. Gray, E. M. Cherry, and F. H. Fenton, “Simultaneous quantification of spatially discordant alternans in voltage and intracellular calcium in Langendorff-perfused rabbit hearts and inconsistencies with models of cardiac action potentials and Ca transients,” *Front. Physiol.* **8**, 819 (2017).
- 13 F. X. Witkowski, L. J. Leon, P. A. Penkoske, W. R. Giles, M. L. Spano, W. L. Ditto, and A. T. Winfree, “Spatiotemporal evolution of ventricular fibrillation,” *Nature* **392**, 78–82 (1998).
- 14 R. A. Gray, A. M. Pertsov, and J. Jalife, “Spatial and temporal organization during cardiac fibrillation,” *Nature* **392**, 75–78 (1998).
- 15 J. Christoph, M. Chebbok, C. Richter, J. Schröder-Schetelig, P. Bittihn, S. Stein, I. Uzelac, F. H. Fenton, G. Hasenfuß, R. Gilmour, Jr., and S. Luther, “Electromechanical vortex filaments during cardiac fibrillation,” *Nature* **555**, 667–672 (2018).
- 16 W. J. Rappel, F. Fenton, and A. Karma, “Spatiotemporal control of wave instabilities in cardiac tissue,” *Phys. Rev. Lett.* **83**, 456–459 (1999).
- 17 D. J. Christini, M. L. Riccio, C. A. Cuiianu, J. J. Fox, A. Karma, and R. F. Gilmour, Jr., “Control of electrical alternans in canine cardiac Purkinje fibers,” *Phys. Rev. Lett.* **96**, 104101 (2006).

- ¹⁸C. M. Berger, J. W. Cain, J. E. S. Socolar, and D. J. Gauthier, "Control of electrical alternans in simulations of paced myocardium using extended time-delay autosynchronization," *Phys. Rev. E* **76**, 041917 (2007).
- ¹⁹A. Garzón, R. O. Grigoriev, and F. H. Fenton, "Continuous-time control of alternans in long Purkinje fibers," *Chaos* **24**, 033124 (2014).
- ²⁰K. Kulkarni, S. W. Lee, R. Kluck, and E. G. Tolkacheva, "Real-time closed loop diastolic interval control prevents cardiac alternans in isolated whole rabbit hearts," *Ann. Biomed. Eng.* **46**, 555–566 (2018).
- ²¹B.-R. Choi, W. Jang, and G. Salama, "Spatially discordant voltage alternans cause wavebreaks in ventricular fibrillation," *Heart Rhythm* **4**, 1057–1068 (2007).
- ²²S. Shahi, C. D. Marcotte, C. J. Herndon, F. H. Fenton, Y. Shiferaw, and E. Cherry, "Long-time prediction of arrhythmic cardiac action potentials using recurrent neural networks and reservoir computing," *Front. Physiol.* **12**, 1585 (2021).
- ²³N. Ibtihaz, M. S. Rahman, and M. S. Rahman, "VFPred: A fusion of signal processing and machine learning techniques in detecting ventricular fibrillation from ECG signals," *Biomed. Signal Process. Control* **49**, 349–359 (2019).
- ²⁴G. T. Taye, E. B. Shim, H.-J. Hwang, and K. M. Lim, "Machine learning approach to predict ventricular fibrillation based on QRS complex shape," *Front. Physiol.* **10**, 1193 (2019).
- ²⁵M. Murugappan, L. Murugesan, S. Jerritta, and H. Adeli, "Sudden cardiac arrest (SCA) prediction using ECG morphological features," *Arab. J. Sci. Eng.* **46**, 947–961 (2021).
- ²⁶Z. Bar-Joseph, G. K. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon, "Continuous representations of time-series gene expression data," *J. Comput. Biol.* **10**, 341–356 (2003).
- ²⁷M. Ghil and R. Vautard, "Interdecadal oscillations and the warming trend in global temperature time series," *Nature* **350**, 324–327 (1991).
- ²⁸K. Limthong, "Real-time computer network anomaly detection using machine learning techniques," *J. Adv. Comput. Netw.* **1**, 126–133 (2013).
- ²⁹Q. Gong, Y. Chen, X. He, Z. Zhuang, T. Wang, H. Huang, X. Wang, and X. Fu, "DeepScan: Exploiting deep learning for malicious account detection in location-based social networks," *IEEE Commun. Mag.* **56**, 21–27 (2018).
- ³⁰V. Plagianakos and E. Tzanaki, "Chaotic analysis of seismic time series and short term forecasting using neural networks," in *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)* (IEEE, 2001), Vol. 3, pp. 1598–1602.
- ³¹R. S. Tsay, *Analysis of Financial Time Series* (John Wiley & Sons, 2005), Vol. 543.
- ³²A. Dingli and K. S. Fournier, "Financial time series forecasting—A deep learning approach," *Int. J. Mach. Learn. Comput.* **7**, 118–122 (2017).
- ³³H. Zhao, "A chaotic time series prediction based on neural network: Evidence from the Shanghai Composite Index in China," in *2009 International Conference on Test and Measurement* (IEEE, 2009), Vol. 2, pp. 382–385.
- ³⁴S. Takahashi, Y. Chen, and K. Tanaka-Ishii, "Modeling financial time-series with generative adversarial networks," *Phys. A* **527**, 121261 (2019).
- ³⁵R. Billinton, H. Chen, and R. Ghajar, "Time-series models for reliability evaluation of power systems including wind energy," *Microelectron. Reliab.* **36**, 1253–1261 (1996).
- ³⁶D. W. Bunn, "Forecasting loads and prices in competitive power markets," *Proc. IEEE* **88**, 163–169 (2000).
- ³⁷A. Deihimi and H. Showkati, "Application of echo state networks in short-term electric load forecasting," *Energy* **39**, 327–340 (2012).
- ³⁸R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access* **8**, 67366–67376 (2020).
- ³⁹Y. Zhang, P. Tiwari, D. Song, X. Mao, P. Wang, X. Li, and H. M. Pandey, "Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis," *Neural Netw.* **133**, 40–56 (2021).
- ⁴⁰Y. Chen, J. Wu, and H. Bu, "Stock market embedding and prediction: A deep learning method," in *2018 15th International Conference on Service Systems and Service Management (ICSSSM)* (IEEE, 2018), pp. 1–6.
- ⁴¹H. Jaeger, *Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the "Echo State Network" Approach* (GMD-Forschungszentrum Informationstechnik Bonn, 2002), Vol. 5.
- ⁴²M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.* **3**, 127–149 (2009).
- ⁴³F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, "Other recurrent neural networks models," in *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis* (Springer International Publishing, Cham, 2017), pp. 31–39.
- ⁴⁴Z. Han, J. Zhao, H. Leung, K. F. Ma, and W. Wang, "A review of deep learning models for time series prediction," *IEEE Sens. J.* **21**, 7833–7848 (2021).
- ⁴⁵C. Sun, M. Song, S. Hong, and H. Li, "A review of designs and applications of echo state networks," *arXiv:2012.02974* (2020).
- ⁴⁶E. Bollt, "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 (2021).
- ⁴⁷D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, "Next generation reservoir computing," *Nat. Commun.* **12**, 5564 (2021).
- ⁴⁸H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Netw.* **20**, 335–352 (2007).
- ⁴⁹J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model," *Chaos* **28**, 041101 (2018).
- ⁵⁰N. A. K. Doan, W. Polifke, and L. Magri, "Physics-informed echo state networks," *J. Comput. Sci.* **47**, 101237 (2020).
- ⁵¹D. K. Oh, "Toward the fully physics-informed echo state network—An ODE approximator based on recurrent artificial neurons," *arXiv:2011.06769* (2020).
- ⁵²I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning* (MIT Press, 2016), Vol. 1.
- ⁵³D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," Technical Report, Institute for Cognitive Science, University of California, San Diego, La Jolla, CA, 1985.
- ⁵⁴N. Japkowicz, C. Myers, M. Gluck *et al.*, "A novelty detection approach to classification," in *IJCAI'95: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI, 1995)*, Vol. 1, pp. 518–523.
- ⁵⁵B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Comput. Ind.* **131**, 103498 (2021).
- ⁵⁶G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science* **313**, 504–507 (2006).
- ⁵⁷X. J. Guizuan Zhang and Y. Liu, "A survey of autoencoder-based recommender systems," *Front. Comput. Sci.* **14**, 430 (2020).
- ⁵⁸M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," *arXiv:1206.4683* (2012).
- ⁵⁹G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.* **18**, 1527–1554 (2006).
- ⁶⁰G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.* **29**, 82–97 (2012).
- ⁶¹I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning (PMLR, 2013)*, pp. 1139–1147.
- ⁶²P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.* **11**, 3371–3408 (2010).
- ⁶³C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, 2017), pp. 665–674.
- ⁶⁴J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.* **2**, 4507–4518 (2020).
- ⁶⁵S. Maleki, S. Maleki, and N. R. Jennings, "Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering," *Appl. Soft Comput.* **108**, 107443 (2021).

- ⁶⁶P. Liu, X. Sun, Y. Han, Z. He, W. Zhang, and C. Wu, "Arrhythmia classification of LSTM autoencoder based on time series anomaly detection," *Biomed. Signal Process. Control* **71**, 103228 (2022).
- ⁶⁷R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *IEEE Access* **9**, 83105–83123 (2021).
- ⁶⁸A. H. Mirza and S. Cosan, "Computer network intrusion detection using sequential LSTM neural networks autoencoders," in *2018 26th Signal Processing and Communications Applications Conference (SIU)* (IEEE, 2018), pp. 1–4.
- ⁶⁹T. Tang, J. Jia, and H. Mao, "Dance with melody: An LSTM-autoencoder approach to music-oriented dance synthesis," in *Proceedings of the 26th ACM International Conference on Multimedia* (Association for Computing Machinery, 2018), pp. 1598–1606.
- ⁷⁰B. Hou, J. Yang, P. Wang, and R. Yan, "LSTM-based auto-encoder model for ECG arrhythmias classification," *IEEE Trans. Instrum. Meas.* **69**, 1232–1240 (2019).
- ⁷¹K. Jun, D.-W. Lee, K. Lee, S. Lee, and M. S. Kim, "Feature extraction using an RNN autoencoder for skeleton-based abnormal gait recognition," *IEEE Access* **8**, 19196–19207 (2020).
- ⁷²M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade* (Springer, 2012), pp. 659–686.
- ⁷³P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," [arXiv:2103.12057](https://arxiv.org/abs/2103.12057) (2021).
- ⁷⁴M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *2017 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2017), pp. 3854–3861.
- ⁷⁵F. Fenton and A. Karma, "Vortex dynamics in three-dimensional continuous myocardium with fiber rotation: Filament instability and fibrillation," *Chaos* **8**, 20–47 (1998).
- ⁷⁶F. H. Fenton, E. M. Cherry, H. M. Hastings, and S. J. Evans, "Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity," *Chaos* **12**, 852–892 (2002).
- ⁷⁷G. W. Beeler and H. Reuter, "Reconstruction of the action potential of ventricular myocardial fibres," *J. Physiol.* **268**, 177–210 (1977).
- ⁷⁸Z. Qu, J. N. Weiss, and A. Garfinkel, "Spatiotemporal chaos in a simulated ring of cardiac cells," *Phys. Rev. Lett.* **78**, 1387 (1997).
- ⁷⁹S. Shahi, F. H. Fenton, and E. M. Cherry, "Prediction of chaotic time series using recurrent neural networks and reservoir computing techniques: A comparative study," *Mach. Learn. Appl.* **8**, 100300 (2022).
- ⁸⁰S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.* **22**, 1345–1359 (2009).
- ⁸¹K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data* **3**, 9 (2016).