Neural Processing Letters
https://doi.org/10.1007/s11063-022-10950-2

Cyclic Gate Recurrent Neural Networks for Time Series Data with Missing Values

Philip B. Weerakody[1] · **Kok Wai Wong**[1] · **Guanjin Wang**[1]

Accepted: 27 June 2022

Abstract
Gated Recurrent Neural Networks (RNNs) such as LSTM and GRU have been highly effective in handling sequential time series data in recent years. Although Gated RNNs have an inherent ability to learn complex temporal dynamics, there is potential for further enhancement by enabling these deep learning networks to directly use time information to recognise time-dependent patterns in data and identify important segments of time. Synonymous with time series data in real-world applications are missing values, which often reduce a model's ability to perform predictive tasks. Historically, missing values have been handled by simple or complex imputation techniques as well as machine learning models, which manage the missing values in the prediction layers. However, these methods do not attempt to identify the significance of data segments and therefore are susceptible to poor imputation values or model degradation from high missing value rates. This paper develops Cyclic Gate enhanced recurrent neural networks with learnt waveform parameters to automatically identify important data segments within a time series and neglect unimportant segments. By using the proposed networks, the negative impact of missing data on model performance is mitigated through the addition of customised cyclic opening and closing gate operations. Cyclic Gate Recurrent Neural Networks are tested on several sequential time series datasets for classification performance. For long sequence datasets with high rates of missing values, Cyclic Gate enhanced RNN models achieve higher performance metrics than standard gated recurrent neural network models, conventional non-neural network machine learning algorithms and current state of the art RNN cell variants.

Keywords Time series · Missing values · Recurrent neural network · GRU · LSTM · RNN

1 Introduction

Due to the numerous types of sensing devices or recording practices that generate data, it is rare for raw time series data to have all input features sampled at a constant rate with common timestamps and consistency across multiple features [1]. Missing observations are common

✉ Philip B. Weerakody
p.weerakody@murdoch.edu.au

[1] Discipline of Information Technology, Murdoch University, Perth, WA, Australia

Published online: 24 July 2022

in univariate and multivariate datasets. Missing data occur in univariate data generated from a single feature variable measured at a sampling rate without a consistent interval between observations due to unstructured manual processes, event-driven monitoring, device or signal failure, and intentional omissions based on cost or importance. For multivariate datasets derived from numerous measuring techniques and instruments, the frequency at which each variable is sampled will often be different and result in missing values for one or more features at any given timestamp.

The impact of missing values on data modelling often results in performance degradation in forecasting and classification tasks [2]. Therefore, dealing with missing values is an important and often overlooked part of building an effective model. There are two common approaches for handling missing values in time series data: missing value imputation at the data pre-processing stage [3–6] and modification of algorithms to directly handle missing values in the learning process [7, 8]. Imputation based methods estimate missing values and reconstruct a complete time series which is subsequently fed into prediction layers. Methods that rely on algorithms within the prediction model to handle missing values during the learning process, do not aim to develop the most accurate estimation of missing values but rather optimise the final prediction capability taking into account the missing values.

Over the past few decades, most approaches to tackling missing values in datasets have focused on imputation techniques, which range from simple statistical methods such as mean, moving average and simple regression to complex imputation methods involving machine learning (ML) to predict accurate missing values. Simple statistical imputation methods can often introduce a loss of accuracy or bias to models, while complex imputation models are computationally expensive [9, 10]. When applied to time series data, most imputation techniques fail to capture the temporal dependencies between observations in univariate or multivariate data. Additionally, missing patterns, which can be time-dependent, are hidden by imputation techniques and not effectively explored in the prediction layers, resulting in sub-optimal models. Utilising time information in addition to feature variable data has been shown to improve a number of machine learning models for time series prediction tasks with missing or irregular data [8]. Augmentation of model inputs with time values or time intervals can be applied in several ways, including their use in learnt decay functions to impute missing values [11] or direct input of time values as a feature variable for the prediction layers [12].

Standard gated Recurrent Neural Network (RNN) models such as LSTM and GRU and their associated variants provide an ideal starting point for handling time sequential data due to their success in providing state-of-the-art performance in sequential data modelling tasks. Their successful applications have included machine translation, speech recognition and other natural language processing (NLP) applications that require learning temporal dependencies within a sequence of text [13]. Gated recurrent neural networks are capable of utilising time inputs within the model and learning temporal patterns in sequential time series data [14]. Their architectures have the flexibility to allow for modifications to address specific data issues, including irregular time series sequences [10]. Traditional machine learning techniques for handling sequential data not based on Neural Network (NN) architectures have included examples such as Naive Bayes, k-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest (RF), which predominantly rely on feature extraction prior to inference and therefore fail to utilise the rich information associated with the raw time sequence. Time information has been successfully utilised as part of structural modification of conventional recurrent neural networks, such as the LSTM and GRU, by modification of their gate operations [1, 8]. These modifications allow for better prediction of sequences that can be long, noisy or sparse, by enabling the model to be aware of patterns in time that identify more and less significant segments of data. The time-aware concepts behind these

structural changes to recurrent neural networks can similarly be applied to missing data so that the prediction model can be less susceptible to poor imputation values. They can also identify patterns in the missing data or input values, allowing only essential data segments to be input or remembered by the model's current state.

However, handling missing time series data as part of the learning process with a time-aware architecture is still a relatively undeveloped field given the limited body of research in this area. Many current state-of-the-art models continue to be significantly impacted by the occurrences of high levels of missing data [15]. Recent cutting edge RNN based models that address the missing value problem have applied time intervals to generate time decay functions used by gated RNN models [11, 16, 17] for regulating imputation values with time. Innovative models with modified recurrent cell architectures with time-aware units [7, 8] have not been applied to missing values data with real-valued feature variables. None of these current works focus on identifying patterns within the data to distinguish important segments of data with respect to time in order to manage missing values.

Solving the challenge of missing values is often dependent on the type of data in which the missing values occurs and how the missing data occur [18]. Both these characteristics provide valuable information which can be learnt and used to mitigate the negative effects of missing values on a model's performance. Time awareness in data can refer to the periodicity of occurrences of important or unimportant data segments, which may occur as a well-defined repeating pattern or within consistently positioned timesteps within a sequence. Supplementing existing gated RNN architectures with additional gates defined by appropriate waveforms provides a technique for learning the patterns associated with significant sections of data within a sequence.

We draw upon the successful development of models in different but related areas, in this case, long term sequential data modelling, to provide methods for resolving the missing value problem. Our proposed models extend concepts introduced by Neil et al. [1] by introducing alternate waveforms for customisation of gate control on the LSTM and GRU models to better identify important data segments for datasets with missing data. The Phased LSTM's gate control is limited to a piecewise linear function, analogous to a Relu, which aims to learn long sequences and accelerate training. The model does not specifically address missing values but does consider asynchronous sampling rates, which pose similar issues to missing values and therefore provide a valid application for missing values in time series.

This paper develops a novel deep learning cell architecture called Cyclic Gate Recurrent Neural Networks to exploit informative input and missingness patterns based on time information. The models further develop the concept of discriminating rhythmic signals by using simple and Fourier Series waveforms. Cyclic Gate Recurrent Neural Networks use time information to generate periodic waveforms that control the gates' opening and closing characteristics in the recurrent cells. Instead of memory and output updates on every fixed time step, the use of learnt waveforms allows updates only when necessary, which allow for more fine-tuned updates of the recurrent cell and assists in reducing the effect of missing values. The proposed models aim to enhance the performance of model predictions against the original gated RNNs and recent variants.

Empirical experiments on several real-world time series datasets with simulated missing values of different types, MCAR, MAR and MNAR, demonstrate that our proposed model outperforms the baselines RNN models and a number of recent gated RNN variants. The experiments show that our method is suitable for a range of time series classification problems with high rates of missing data, and in particular, is highly suited to predictive tasks on datasets that have periodic behaviour. The main contributions of the paper are:

(1) Providing a modified recurrent neural network architecture to be applied to time series data with high rates of missing data that can capture time-based patterns of input features and improve classification performance. The modifications include the use of differentiable Fourier Series waveforms for driving gate activation to learn cyclic patterns within time series sequences.
(2) Modelling long sequence time series data with missing values, while avoiding vanishing gradients by using shortened back-propagation paths.
(3) Handling missing values by considering the significance of segments of data while utilising a minimal missing value imputation strategy to reduce the potential inaccuracies associated with imputed data.
(4) Performing model modifications at the cell level instead of the more commonly applied network level, which allows for subsequent incorporation of cells into higher level network architectures.

2 Background

Recurrent Neural Networks (RNN) are a leading machine learning technique for dealing with sequential data [19] and tasks requiring memory of past events [20]. In particular, Long Short-Term Memory (LSTM) [21], as well as Gated Recurrent Units (GRU) [22] have emerged as two of the most effective models for sequential data modelling due to their ability to handle long sequences while limiting the effect of vanishing gradients.

One of the most prevalent types of sequential data, alongside language data, is time series data which arises wherever collected data is indexed and ordered by time. Gated RNN models are well suited to modelling time series data with long-term dependencies [21, 23] due to their internal memory mechanism that allows access to the history of previous time series values. In the medical diagnostics field, Lipton et al. [12] use LSTMS for diagnoses classification of critical care patients by recognising patterns in multivariate time series data, while Malhotra et al. [24] uses stacked LSTM networks for anomaly detection in ECG time series data. Hsu et al. [25] present a model which combines LSTM and Auto-encoder (AE) architectures to capture long-term dependencies across data points. Malhotra et al.'s Timenet [26] demonstrate the strength of pre-trained deep RNNs for time series classification, using a Sequence Auto-Encoder (SAE) to learn latent representations of time series data. Qin et al. [27] successfully combine a LSTM encoder-decoder architecture with dual Attention, to extract relevant driving RNNS for time series with missing values. Shukla et al. [28] utilise Multi-Time Attention Networks to learn embedding of continuous-time values by applying bidirectional RNNs and an attention mechanism to handle sparse and irregular data. Wang et al. [29] develop an ensemble architecture with a CNN combined with a Sequence-to-Sequence Attention mechanism in the hidden state of the LSTM for long time series forecasting.

In recent years there has been an increasing interest by researchers in designing modified recurrent cell architectures of gated RNNs to increase the efficiency and accuracy of the models for various data types. In order to reduce the number of cell parameters and provide for faster training, Zhou et al. [30] developed the Minimal Gated Unit (MGU). Nina and Rodriguez [31] and Hu [32] also simplify the LSTM cell by coupling the forget gate and input gate into one gate for greater accuracy in image descriptions and long time series, respectively. Jozefowicz et al. [33] conducted a comprehensive evaluation of RNN architectures, involving a review of over 10,000 different architectures, which identified three specific models that

outperformed the LSTM and GRU on a selection of tasks. Modification of internal RNN architectures has also involved changes beyond increasing and decreasing gate functions, with Rahman, Mohammed, and Azad [34] presenting a biologically inspired variant of the LSTM, which changes the update mechanism of the LSTM cell state to enhance cell capacity and improve sentiment analysis of textual data. An LSTM with working memory was introduced by Pulver and Lyu [35], and Mirza [36] modified the GRU model by performing linear calculations in the frequency domain prior to performing non-gating functions. In both these cases, the focus was on providing a gated RNN variant for improving prediction performance.

Of the research that has been conducted on RNN internal cell modifications, several have included optimisation for handling missing or irregular data, as well as the direct utilisation of time information. Che et al. [11] investigate the application of Gated Recurrent Units with trainable decays on multivariate time series with missing data. Pham et al. [7] modifies an LSTM model to generate predictions based on healthcare observations using time parameterisations to handle irregular timed events, moderating the forget operation and consolidation of memory cells. Baytas et al. [8] further develop irregular data research through a proposed time-aware LSTM (T-LSTM) model to handle irregular time intervals in longitudinal patient records by using the elapsed time between consecutive elements to adjust the memory content of the LSTM unit. Similar to the T-LSTM, Tan et al. [16] use a time-aware GRU structure to provide an end-to-end dual-attention time-aware gated recurrent unit (DATA-GRU) to predict patients' mortality risk on multivariate data.

Neil et al. [1] introduce a Phased LSTM model which modifies the base LSTM unit by adding a new timing gate that is of an oscillatory nature, which updates the memory cell only during a fraction of its cycle and therefore handles asynchronous input data from sensors with irregular sampling rates. The model differs from the standard LSTM by requiring the input of event times. An independent periodic function based on a piecewise linear function analogous to the Relu function determines the opening and closing of a new time gate. The paper focuses on accelerated learning of long sequences and discriminating rhythmic signals. The Phased LSTM has been applied in several subsequent research papers [37], which demonstrate its effectiveness in handling long, sparse time series. The Phased LSTM has limited benefit for short sequences, and its proposed piecewise linear gating function is unlikely to be flexible enough to assist in discriminating certain data signals or patterns due to its short fully-open period. Skip RNNs [38, 39] also focus on accelerated learning of long sequences, in this case, through the use of gate structures with skip connections to manage time scales with learned gate parameters. The MS-LMN model [40] considers the importance of different frequencies in long sequence data by separating hidden states in the simple RNN into different modules with different sampling rates, using an incremental training algorithm to target multiscale learning.

In our proposed models, we extend the concept of oscillatory learned gates and discriminatory rhythmic signals [1] in order to handle missing data from univariate and multivariate time series data. The handling of sparse updates is extended to missing values as a method for identifying patterns in time series data through time augmentation and waveform-controlled gate activations. We generate several simple and sophisticated waveforms with learned parameters and apply them to gated recurrent neural networks to test their performance against baseline recurrent neural networks, conventional non-neural network ML algorithms and several recent state of the art gated RNNs.

3 The Cyclic Gate RNN Model

This section describes the recurrent neural network cells with modified gates, whereby cyclic waveforms control their opening and closing operation. The resulting recurrent cell aims to make sparser updates to units, thereby being more selective on when input data updates the memory and when updates to the hidden state occur. We begin by providing notation for time series data, followed by the equations for the recurrent cells.

Assume a multivariate time series $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\} \in \mathbb{R}^{T \times D}$ with time series length T where the sample $\boldsymbol{x}_t = \left(x_t^1, .., x_t^d, \ldots, x_t^D\right)$ has D multivariate dimensions. Missing indicators $m_t^d = 1 if x_t^d$ is observed and $m_t^d = 0 if x_t^d$ is missing. In a labelled dataset with N samples, we have a set of features and labels $\left\{\left(\boldsymbol{X}^{(n)}, \boldsymbol{y}^{(n)}\right)\right\}_{n=1}^{N}$, where each timeseries $\boldsymbol{X}^{(n)}$ has an associated label $\boldsymbol{y}^{(n)}$.

3.1 Standard Gated Recurrent Model

Modifications are made to standard LSTM and GRU cells, resulting in cell models termed the Cyclic Gate LSTM and Cyclic Gate GRU.

LSTMs are defined by the following equations [21].

$$f_t = \sigma\left(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f\right) \tag{1}$$

$$i_t = \sigma(W_{xi}.x_t + W_{hi}.h_{t-1} + b_i) \tag{2}$$

$$o_t = \sigma(W_{xo}.x_t + W_{ho}.h_{t-1} + b_o) \tag{3}$$

$$c_t' = tanh\left(W_{xg}.x_t + W_{hg}.h_{t-1} + b_g\right) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot c_t' \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where x_t is the input at time step t; W are weight parameter matrices; b are bias vectors; c_t is the cell state at time step t; h_t is the hidden state at time step t; "." is an inner product for standard matrix multiplication; $\odot$ is the elementwise (Hadamard) product; and σ is the Sigmoid function. Both weights and biases are shared through all time steps. Three gates, consisting of the input gate i, forget gate f, and output gate o, modulate the flow of information inside the cell by generating values in the range [0, 1] to write the input to the internal memory c_t, reset the memory, or read from memory, respectively.

GRUs are defined by the following equations [22].

$$r_t = \sigma(W_{xr}.x_t + W_{hr}.h_{t-1} + b_r) \tag{7}$$

$$z_t = \sigma(W_{xz}.x_t + W_{hz}.h_{t-1} + b_z) \tag{8}$$

$$h_t' = tanh(W_{xh'}.x_t + W_{hh'}.(r_t \odot h_{t-1}) + b_c) \tag{9}$$

$$h_t = (z_t) \odot h_t' + (1 - z_t) \odot h_{t-1} \tag{10}$$

where x_t is the input at time step t; W are weight parameter matrices; b are bias vectors; h_t is the hidden state at time step t; "." is an inner product for standard matrix multiplication;

$\odot$ is the elementwise (Hadamard) product; and σ is the Sigmoid function. Both weights and biases are shared through all time steps. Two gates, consisting of the reset gate r and update gate z, modulate the flow of information inside the cell by generating values in the range [0, 1] to determine how much past information to reset or forget and how much of the current input and previous state to output to the new hidden state.

3.2 Cyclic Gated Recurrent Model

The Cyclic Gate LSTM uses two new gates, one for controlling the input to the cell state memory and one for controlling the output to the hidden state. The GRU does not have a distinct cell state like the LSTM and only propagates the hidden state h_t through time; therefore, the Cyclic Gate GRU cell has a single new gate that controls output to the hidden state.

A time-dependent waveform controls the operation of the Cyclic Gates. Four learned parameters define the waveform.

T—Period (wavelength).

S—Phase Shift.

R—Ratio On–Off (duration of the "open" state to the duration of the full period T).

A – Amplitude.

Unlike conventional gated RNNs, the Cyclic Gate models do not require the RNN to update at every regular time timestep but can alternatively update at sparse points in time and neglect time periods in between. This functionality also allows for different update rates for features with different sampling periods. For a particular RNN cell, the update time between observations of a feature can occur with sparse and irregular intervals between updates. As an example of sparse timing of a feature variable, for a time window of 10 s, the time sequence t' may differ from regular sampling sequence t, as shown below.

Timestamp (sec)	1	2	3	4	5	6	7	8	9	10
t	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
t'	t'_1				t'_2					t'_3

The waveform equation which drives the cyclic gates is represented by the notation $g_{t'}$. This wave function $g_{t'}$ will be described in detail in the next section.

The internal architecture of a single Cyclic Gate LSTM cell is shown in Fig. 1a. The additional Cyclic Gate LSTM equations are:

$$\tilde{c}_{t'} = f_{t'} \odot c_{t'-1} + i_t \odot c'_{t'} \tag{11}$$

$$c_{t'} = g_{t'} \odot \tilde{c}_{t'} + (1 - g_{t'}) \odot c_{t'-1} \tag{12}$$

$$\tilde{h}_{t'} = o_{t'} \odot \tanh(\tilde{c}_{t'}) \tag{13}$$

$$h_{t'} = g_{t'} \odot \tilde{h}_{t'} + (1 - g_{t'}) \odot h_{t'-1} \tag{14}$$

The internal architecture of a single Cyclic Gate GRU cell is shown in Fig. 1b. The additional Cyclic Gate GRU equations are:

$$\tilde{h}_{t'} = z_{t'} \odot h'_{t'} + (1 - z_{t'}) \odot h_{t'-1} \tag{15}$$

$$h_{t'} = g_{t'} \odot \tilde{h}_{t'} + (1 - g_{t'}) \odot h_{t'-1} \tag{16}$$

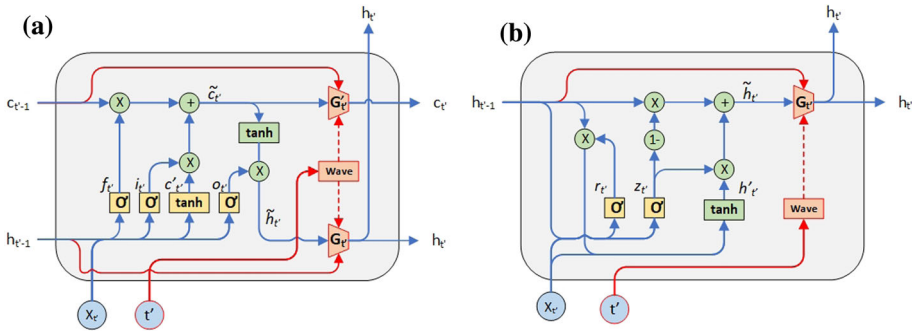


Fig. 1 Cyclic Gate Models with modification in red **a** Cyclic Gate LSTM, **b** Cyclic Gate GRU

To simplify the architecture drawings in Fig. 1, we use summary notations $G^{'}_{t'}$ and $G_{t'}$ to represent the equations applying $g_{t'}$ as follows.

For Fig. 1a, $G^{'}_{t'} = g_{t'} \odot \tilde{c}_{t'} + (1 - g_{t'}) \odot c_{t'-1}$, which refers to Eq. (12) and $G_{t'} = g_{t'} \odot \tilde{h}_{t'} + (1 - g_{t'}) \odot h_{t'-1}$, which refers to Eq. (14).

For Fig. 1b, $G_{t'} = g_{t'} \odot \tilde{h}_{t'} + (1 - g_{t'}) \odot h_{t'-1}$, which refers to Eq. (16).

3.3 Waveform Equations

We generate a number of periodic waveforms in order to determine which wave types better capture patterns in the input features, taking into account missing values. The wave types include simple and complex forms of square, triangular and sawtooth waves. The new gate controls supplement the current gate activations [1, 21, 22] and therefore enable an immediate opening and closing action (square wave), a ramped opening and closing action (triangular wave) or a ramped opening action with an immediate closing action (sawtooth). The waveforms are generated as pulsed trains, with only positive amplitude, flat intervals between pulses, and a delay between the start of the wave period and the start of the pulse. The square, triangle or sawtooth shapes can represent the level of fine-tuning required for opening the gate for small or large sections of important data and whether a gradual (ramped) input of data is required on either side of the fully open gate value. This can alternatively be considered as a method of noise filtering or dropout; for example, square waves have complete filtering of unwanted data, triangular waves have graduated filtering, and sawtooth waves have graduated filtering only during gate opening and complete filtering afterwards.

To accurately develop a differentiable periodic function for our pulse trains, we use finite Fourier series approximations. The basic concept of the Fourier series is that any periodic waveform can be represented by the sum of harmonically related sinusoidal functions. The sinusoids that make up the resulting waveform have frequencies that are integer multiples of a fundamental frequency, also known as harmonics of the fundamental frequency. A Fourier series expansion of a periodic function is represented as:

$$f(t) = a_0 + a_1 \cos kt + b_1 \sin kt + a_2 \cos 2kt + b_2 \sin 2kt + a_3 \cos 3kt \ldots \tag{17}$$

or,

$$f(t) = \sum_{n=0}^{\infty} a_n . \cos nkt + \sum_{n=0}^{\infty} b_n . \sin nkt \tag{18}$$

where $k = 2\pi/T$ and a_0, a_n, b_n are Fourier Coefficients and $n = 0,1,2,3..\infty$ is considered the harmonic number. The Fourier Coefficients can be given for a T-period signal by the following equations.

$$a_0 = \frac{1}{T}\int_0^T f(t)dt \tag{19}$$

$$a_n = \frac{2}{T}\int_0^T f(t)\cos nktdt \tag{20}$$

$$b_n = \frac{2}{T}\int_0^T f(t)\sin nktdt \tag{21}$$

The Fourier series can be approximated by a finite number of harmonics for each type of periodic function, where $n = 0,1,2,3,..,m$. The sine or cosine functions can be simplified based on trigonometric identities, integrals and whether the function is odd or even. Where *f(t)* is even, the b_n coefficients will be zero, while if *f(t)* is odd, the a_n coefficients will be zero. The resulting Fourier Series representation of continuous time periodic signals is a weighted sum of sinusoidal signals. For the purposes of gate opening operations, our resulting Fourier Series equations are required to generate a train of pulses, which are dependent on time and where the pulse is only active for a fraction of the total period of each cycle.

In addition to Fourier series waveforms, we also use simpler equations for each of the waveforms to compare their performance, taking into account the high computation complexity associated with Fourier series waveforms. Figure 2 shows the resulting waveforms, and the associated equations are presented in the formulas for $g(t)$ below.

$$\text{Fourier Square} \quad g(t) = \frac{T_P}{T} + \sum_{n=1}^{m}\left(\frac{2}{n\pi}\right)\sin\left(\frac{n\pi T_P}{T}\right).\cos\left(\frac{2\pi nt}{T}\right) \tag{22}$$

$$\text{Fourier Triangular} \quad g(t) = \frac{T_P}{2T} + \sum_{n=1}^{m}\left(\frac{4T}{T_P(n\pi)^2}\right)\sin^2\left(\frac{n\pi T_P}{2T}\right).\cos\left(\frac{2\pi nt}{T}\right) \tag{23}$$

$$\text{Fourier sawtooth} \quad g(t) = \frac{T_P}{4T} + \sum_{n=1}^{m}\left(\frac{T}{4T_P(n\pi)^2}\right)\left(\left(\frac{jn2\pi T_P}{T}+1\right).e^{\frac{-jn2\pi T_P}{T}} - 1\right).e^{\frac{jn2\pi t}{T}} \tag{24}$$

$$\begin{aligned}\text{Simple Square} \quad g(t) &= 0 if T.floor\left(\frac{t-s}{T}\right) > t > T.floor\left(\frac{t-s}{T}\right) + T_P \\ &= 1 if T.floor\left(\frac{t-s}{T}\right) < t < T.floor\left(\frac{t-s}{T}\right) + T_P\end{aligned} \tag{25}$$

$$\begin{aligned}\text{Simple Triangular} \quad g(t) &= \frac{2\varphi(t)}{R} if \varphi(t) < \frac{R}{2} \\ &= 2 - \frac{2\varphi(t)}{R} if \frac{R}{2} \le \varphi(t) \le R \\ &= \alpha.\varphi(t) if R < \varphi(t) \\ \text{where,} \quad \varphi(t) &= \frac{\text{mod}(t-s,T)}{T}\end{aligned} \tag{26}$$

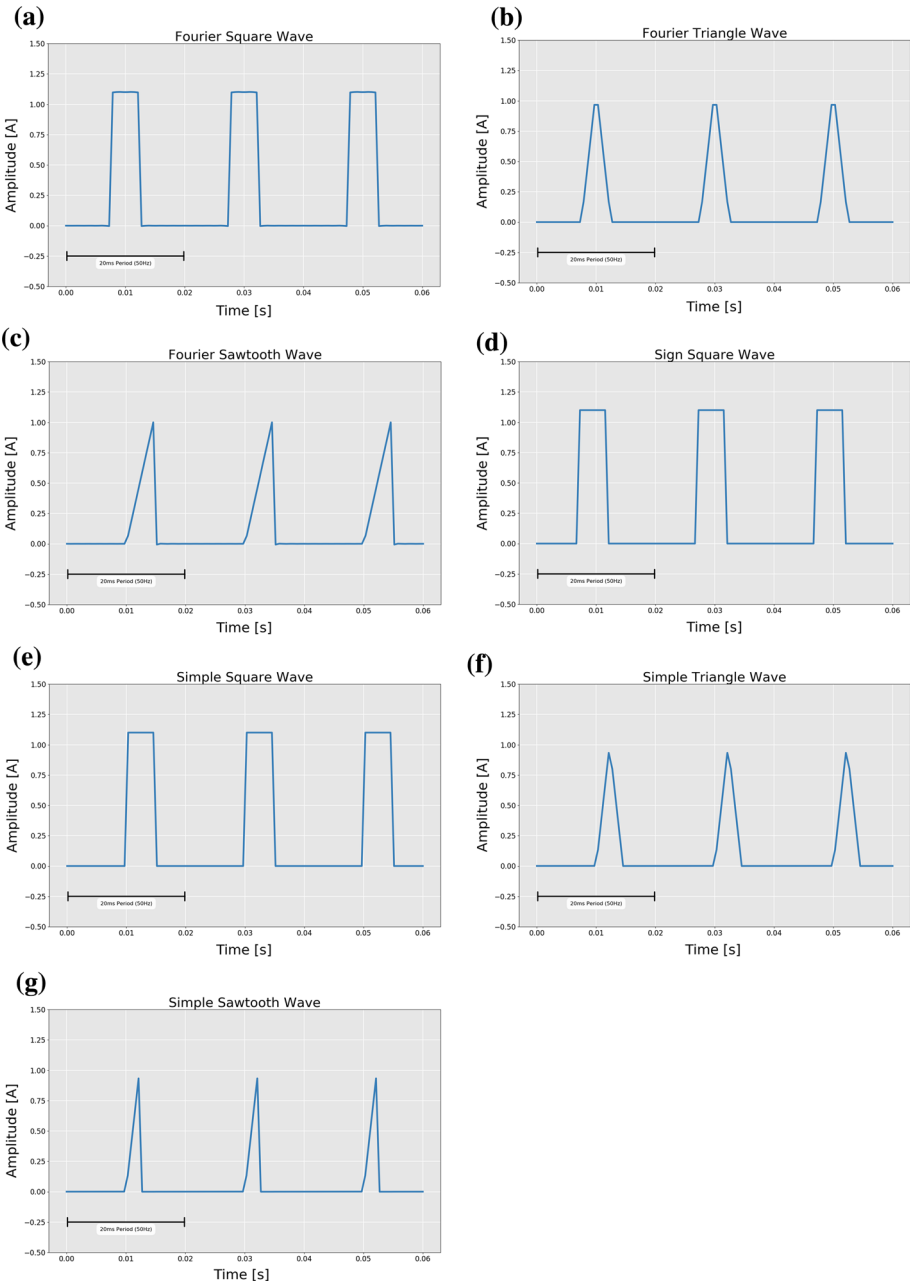


Fig. 2 Waveforms **a** Fourier square wave **b** Fourier triangle wave **c** Fourier sawtooth wave **d** Sign square wave **e** Simple square wave **f** Simple triangle wave **g** Simple sawtooth wave

$$\begin{aligned} \text{Simple Sawtooth} \quad g(t) &= \frac{2\varphi(t)}{R} if \varphi(t) < \frac{R}{2} \\ &= 0 if \frac{R}{2} \leq \varphi(t) \leq R \\ &= \alpha.\varphi(t) if R < \varphi(t) \\ \text{where,} \qquad \varphi(t) &= \frac{\text{mod}(t-s, T)}{T} \end{aligned} \tag{27}$$

$$\text{Sign Square} \quad g(t) = 1 + sign(\text{mod}((t-s), T) - (T - T_P)) \tag{28}$$

where t = time, s = phase shift, T = Period (or cycle), T_p = Pulse width, φ = phase inside a cycle, R = ratio of ON period to total period, α = leak rate, m = finite number of terms for the series.

We note that finite Fourier series representations are smooth waveforms, which are continuous and have well defined derivatives for back-propagation. The non-Fourier waveforms are piecewise continuous functions which are not differentiable at every point due to discontinuity in their derivatives. Although discontinuity in the derivatives does not make back-propagation impossible, due to the availability of sub-gradients, the presence of multiple points of discontinuity may have some impact on the back-propagation process [41, 42].

3.4 Computational Complexity

Considering only Time Complexity, which is the quantity of time taken by an algorithm to run, as a function of the length of the input, as denoted using the big-O notation. The standard LSTM is known to be local in time complexity, meaning that its time complexity per time step and weight is *O(1)* constant [21, 43]. The LSTM architecture contains an input layer, a recurrent layer and an output layer. The total number of weight parameters W, with one cell in each memory block and ignoring bias, is shown below.

$$W = (N_c \times N_c \times 4) + (N_i \times N_c \times 4) + (N_c \times N_o) + (N_c \times 3)$$

where N_c is the number of memory cells, N_i is the number of input units, and N_o is the number of output units. Given that the computational complexity of LSTM models per weight and time step when using stochastic gradient descent (SGD) optimisation is *O(1)*, the complexity per time step is *O(W)* [43]. Similarly, for other RNN variants, the time complexity is *O(W)* and linear.

Within the predictive models, augmentation of input values with missing indicators or time information increases complexity from standard gated RNN models to a small degree due to larger weight matrices associated with a larger number of input features. RNN modified gate structures introduce new weight parameters with each additional gate, increasing the complexity of the standard LSTM or GRU model.

A Fourier trigonometric series has a time complexity of $O(n^2)$ [44]. The complexity is large because the sine, cosine or multiply operations need to be done for each time step and each harmonic frequency. This results in high time complexity of $O(n^2)$, meaning that as the number of points of the time series increases, the time to calculate the series will increase by the square of the number of points.

4 Experiments and Results

In this section, we will present the implementation settings, the different datasets and metrics, and the experimental results to evaluate the performance of the Cyclic Gated recurrent network models.

4.1 Implementation Details

We implement Cyclic Gate RNN networks in Tensorflow (Python) using a single RNN layer followed by a single fully connected layer. For consistent comparisons, the number of hidden units which reflect the hidden state vector is set at 32 for all the recurrent network models. A single fully-connected layer (perceptron) serves as a classifier on top of the recurrent network layer to map the final state h_t to a class probability. The model is trained with the Adam optimiser [45] for classification prediction, with mini-batch sizing varying with each dataset. The learning rate is initially set to 0.0025 and decay rate of 0.9 (first moment) and 0.999 (second moment) after each batch iteration. Each dataset is trained for at least 300 epochs.

4.2 Datasets

We evaluate the proposed models on four real-world time series datasets, which are sequential and therefore characterised by relationships between past and future data points. Both univariate and multivariate datasets are included in the experiments. The four datasets have been selected for experimentation to provide a variety of periodicity levels, ranging from very low periodicity to high periodicity across the datasets. We simulate a high missing value rate of 50% for each dataset as a percentage of total data. Recurrent network cell models in Python Tensorflow cannot operate with missing values represented by NaN (Not a Number) input values so very simple imputation of NaN observations are performed using either last observation caried forward (LOCF) or zero value replacement. We include experimentation with three different missing data generation mechanisms as outlined in the section below.

4.2.1 Missing Data Mechanism

Rubin et al. [9] classified missing data problems into three categories based on the missing data mechanism; 'Missing completely at random' (MCAR), 'Missing at random' (MAR), 'Missing not at random' (MNAR).

Missing Completely at Random (MCAR) In this category, missing data points occur completely at random, so there is no systematic mechanism for the cause of the missing data. The probability of a specific observation being missing is independent of the observed data, including time variables, and is also independent of the unobserved (missing) data. For example, random transmission failure of a wireless sensor sending observation from the field to a back-end system would result in MCAR data.

Missing at Random (MAR) For MAR data, the probability of a specific observation being missing is independent of the unobserved data but is dependent on the values of the observed data. As an example, missing data from sensors data may be dependent on time, such that the

probability of missing data is high on the weekend, where maintenance on a faulty transmitter is unlikely to occur.

Not Missing at Random (MNAR) A missing observation in this group is dependent on the unobserved data and may also be dependent on the observed data. For example, a sensor's value will be missing if it is outside the range of the sensors calibrated limits.

4.2.2 Datasets Descriptions

Descriptions of each dataset are presented in this section. The original datasets do not contain any missing values, and therefore all missing values are generated using simulation mechanisms based on the missing data categories described in the previous section.

Non-Invasive Fetal ECG Thorax (ECG) This dataset contains data from non-invasive fetal electrocardiographic (NIFECG) monitoring using electrodes placed on a maternal abdomen, which is used in determining the level of fetal distress based on deceleration of fetal heart rate [46, 47]. The multivariate dataset contains two features corresponding to ECG recordings from the left and right thorax. There are 42 labelled classes and a fixed sequence length of 750 timesteps. The training sample size is 1800, and the test sample size is 1965.

Earthquakes (EQ) The dataset is from Northern California Earthquake Data Center, and each data point is an averaged seismograph reading for one hour, with readings over 36 years [47]. It is a univariate dataset with two labelled classes, representing a major earthquake event, with a Richter scale value above 5 or not a major event, with a Richer scale value below 4. The Sequence length is 512, and the training and test sample sizes are 322 and 139, respectively. There is no overlap in time for each sequence, as segmentation is used instead of a sliding window.

Internal Bleeding (IB) The dataset is generated from three vital signs monitored on 52 pigs before and after an induced injury. The three features monitored were Airway Pressure (airway pressure), Art Pressure (arterial blood pressure) and CVP (central venous pressure) [48]. The time series sequence length is 2000, and there are 52 labelled classes representing each subject animal monitored. There are 104 training samples and 208 test samples.

EOG The source data is from an electrooculography signal (EOG), which measure the electrical potential between electrodes close to the eyes of human subjects [49]. Horizontal and vertical channel signals were measured by taking readings between two electrodes on the left and right of the subject's eye and between the top and bottom of the subject's eye. The two channels represent the two features of the multivariate time series, and there are 12 resulting classes representing stroke identification. The time series sequence length is 1250, and the samples sizes are 362 for training and 362 for testing.

4.2.3 Data Sequence Characteristics

Analysis of the datasets used for experimentation involved the generation of the Discrete Fourier Transform (DFT) of each sequence in a dataset. The Fast Fourier Transform (FFT) is an implementation of the DFT, so the single-sideband spectrum of the FFT was generated for each sequence to identify the dominant DFT coefficients and their corresponding frequencies. For each feature variable within a dataset, the mean value of the two largest DFT coefficients

Table 1 Dominant frequencies and periods of dataset features

Dataset	DFT 1	Freq. 1	Period 1	DFT 2	Freq. 2	Period 2
Non-Invasive Fetal ECG						
Feature 1	250.5282	0.001745	572.7620	142.4506	0.007435	134.4889
Feature 2	256.2887	0.001608	621.8332	144.3580	0.007330	136.4188
Earthquakes						
Feature 1	57.37488	0.183642	5.445369	51.78263	0.200055	4.998605
Internal Bleeding						
Feature 1	2881.707	0.007730	129.3532	1048.514	0.012754	78.40181
Feature 2	11,064.22	0.006000	166.6666	4286.687	0.012471	80.18504
Feature 3	1057.144	0.004432	225.5965	703.2799	0.008082	123.7358
EOG						
Feature 1	30,360.99	0.003292	303.6912	16,369.36	0.006082	164.4258
Feature 2	26,064.80	0.003207	311.8538	14,015.53	0.006039	165.5689

and their frequencies were recorded. Table 1 summarises the DFT dominant coefficient values, frequencies and periods for the datasets used in experiments.

The frequency, period and DFT coefficient values provided in Table 1 are unitless values. All the original datasets contained regular sampling rates, and each sampling event was allocated an incremental integer number in the sequence (e.g., 0,1,2,..,N = Sequence Length − 1), which was input into the FFT.

4.3 Baselines Models

We compare our model to several baseline recurrent neural networks, RNN, LSTM and GRU. LSTM and GRU models with feature augmentation of missing value indicators is included in the set of comparative models. We also compare our model to more recent variations of recurrent cell architectures, the T-LSTM [8] and GRU-D [11].

We provide an additional comparison of our proposed models to several non-neural network-based models to give context to our results against a broader set of ML methods. Support Vector Machines (SVM), k-Nearest Neighbor (KNN) and Random Forest(RF) are adopted to represent a set of frequently applied algorithms for the classification of sequential and time series data [50]. These methods generally rely on feature extraction or generation prior to classification of time series data, without inputting the complete time series directly into the classifier [51]. Therefore, two of our baseline non-NN models, SVM and RF, use time series feature extraction. For the KNN algorithm, we implement a model named KNN-FE with feature extraction and also a model named KNN-TS, which directly inputs the complete time series sequence for classification. Feature extraction includes features from the frequency domain, using the DFT of the time series. The non-NN models are tuned with grid-search hyper-parameter optimisation.

In our experimentation we intentionally exclude high-level network architectures that use RNN layers such as sequence-to-sequence models [19] or ensemble models like CNN-LSTM [52] or ConvLSTM [53], as our focus is on improving the cell architecture and not the high-level network architecture, which these ensemble models are characterised by. The cell

architectures proposed in this paper may be implemented within these high-level networks as part of future works.

4.4 Metrics

Our experiments have been confined to binary or multiclass classification problems on time series datasets, where each sample has a single mutually exclusive class label value. Our evaluation metrics include recall (sensitivity), precision (positive predictive value), F1 score, and area under the curve (AUC) for the Receiver operating characteristic. Accuracy, F1-score and AUC are widely used to measure the classification prediction performance for machine learning approaches. We focus on the AUC and F1-score (harmonic mean between precision and recall), as both the AUC and F1-score are less influenced by imbalanced data with varying class frequencies. In contrast, overall accuracy tends to be biased towards the most dominant classes. Datasets in experiments were not altered to improve class balance. Therefore, a review of results associated with the accuracy metric confirmed values that primarily reflected predictions biased to the majority class and which did not represent a suitable measure of model performance. For this reason, accuracy was omitted from our set of performance evaluation metrics due to its likelihood of leading to erroneous conclusions.

The metrics set also includes an additional AUC metric variation for one of the datasets, named AUC_predicted, where the standard AUC produces values with minimal variation. This metric is provided to distinguish AUC results between models for the Fetal ECG dataset in which the standard AUC metric does not allow for adequate AUC value comparisons. Given that a ROC curve can be generated using any measure of confidence, not just predicted probabilities, we use a predicted decision $= f(X) \in [0,1]$ instead of predicted probability $= f(X) \in \{0,1\}$. The predicted decision has a lower ROCAUC, as it effectively rounds up or rounds down the predicted probability to 1 or 0 depending on if it is above or below the threshold and results in a larger error. In addition to comparatively viewing the AUC between different models, we also view the change in AUC values between a complete dataset and the same dataset with missing values. This value is presented in the results tables as the AUC Δ, representing the change in the AUC values caused by the missing values.

4.5 Results

In this section, we present the experimental results for a variety of binary and multiclass classification scenarios for testing the Cyclic Gate RNNs. We consider the RNN, LSTM and GRU as the baseline models for comparison, followed by the LSTM and GRU with missing indicators augmented into the input feature set and then finally comparison with the more recent T-LSTM and GRU-D. We also compare the different types of waveforms used by the proposed model to identify any relationships associated with the model's waveform and the characteristics of the data sequence being modelled. The metrics are presented in the tables in this section with graduated cell colouring, with the darkest cells representing the best values for each metric. The model with the highest AUC value is also formatted in bold text for easy identification in each table.

The Non-invasive Fetal ECG dataset is characterised by a sequence length per sample which broadly represents the cycle length of a repeating pattern; therefore, there is only one cycle within the sequence length of each sample. The volatility of the sequence is low, and therefore in a 750 timestep sequence containing a single cycle, missing values will have a lower impact on modelling outcomes. From the values in Table 2, the baseline LSTM and GRU

Table 2 Non-invasive Fetal ECG dataset results

Model	Precision	Recall	F1_Score	AUC*	AUC Δ*
RNN	0.4539	0.4382	0.4268	0.7124	13.13%
LSTM	0.6464	0.6188	0.6117	0.8049	6.17%
GRU	0.6898	0.683	0.6779	0.8377	5.95%
LSTM Missing Indicators	0.6575	0.6514	0.6388	0.8216	5.70%
GRU Missing Indicators	0.769	0.7537	0.7517	0.874	3.46%
T-LSTM	0.5069	0.4916	0.4833	0.7397	14.02%
GRU-D	0.3765	0.371	0.3553	0.668	13.20%
Phased-LSTM	0.8046	0.7919	0.7872	0.8934	2.30%
Phased-GRU	0.8005	0.7903	0.7891	0.8927	4.40%
Cyclic Gate LSTM – Square Wave	0.7876	0.7746	0.774	0.8846	2.42%
Cyclic Gate LSTM - Triangle Wave	0.8184	0.8092	0.804	0.9023	2.16%
Cyclic Gate LSTM - Sawtooth Wave	0.8008	0.7944	0.7927	0.8948	3.65%
Cyclic Gate LSTM - Square Sign Wave	0.7334	0.7059	0.6936	0.8495	0.13%
Cyclic Gate LSTM - Simple Square Wave	0.7121	0.6972	0.6884	0.845	5.36%
Cyclic Gate LSTM - Simple Sawtooth	0.5658	0.4836	0.4328	0.7294	1.19%
Cyclic Gate GRU – Square Wave	0.8086	0.8	0.8007	0.8976	1.28%
Cyclic Gate GRU - Triangle Wave	0.792	0.7852	0.7841	0.8901	2.66%
Cyclic Gate GRU - Simple Square Wave	0.8161	0.7954	0.7883	0.8954	2.84%

*AUC_predicted applied instead of AUC based on the predicted probability

have AUC* values averaging 0.821, the T-LSTM and GRU-D have AUC* values averaging 0.704, missing value indicator models average 0.848, while the top 3 Cyclic Gate models average 0.898. The Cyclic Gate LSTM model with triangular waveform provides superior metrics in almost all categories. The Cyclic Gate models have AUC* value improvements over the baseline LSTM and GRU models ranging from 6 to 9%.

The Earthquakes dataset is characterised by a 512-length sequence length per sample, which contains up to 100 repetitions of a cycle of approximate length 5. The volatility of the sequence is high, and the periodicity is also high, although there is some inconsistency in the cycle length over the full sequence. Based on the sequence characteristics, the impact of missing values in a sample will be high. From the values in Table 3, the baseline LSTM and GRU have AUC values averaging 0.613, the T-LSTM and GRU-D have AUC values averaging 0.674, missing value indicator models average 0.644, while the top 3 Cyclic Gate models average 0.738. The best performing model on this dataset is the Cyclic Gate LSTM simple square waveform, which achieves the highest AUC and recall values. The original Phased-LSTM, which is effectively an example of a Cyclic Gate with a simple triangular waveform, achieves the highest F1_score for this dataset.

The EOG dataset does not represent a repeating pattern within each sample sequence or a set of connected sequences from consecutive samples. However, there are important segments of data that are significant to the classification task and other sections which would be considered unimportant, which are consistent across all samples. This dataset is considered an example of a time series dataset with very low periodicity. The sequence has low volatility and a long sequence length of 1250 timesteps. From the values in Table 4, the baseline LSTM and GRU have AUC values averaging 0.814, the T-LSTM and GRU-D have AUC values averaging 0.843, missing value indicator models average 0.839, while the top 3 Cyclic

Table 3 Earthquakes dataset results

Model	Precision	Recall	F1_Score	AUC	AUC Δ
RNN	0.6739	0.7266	0.6839	0.5424	9.08%
LSTM	0.6539	0.6763	0.6635	0.5709	17.99%
GRU	0.7335	0.7626	0.7335	0.656	7.94%
LSTM Missing Indicators	0.7317	0.7626	0.7283	0.697	3.38%
GRU Missing Indicators	0.6627	0.6691	0.6658	0.5909	17.53%
T-LSTM	0.6416	0.7122	0.66	0.6412	11.35%
GRU-D	0.8157	0.7554	0.6571	0.7066	7.01%
Phased-LSTM	0.7347	0.7554	0.7406	0.7547	6.87%
Phased-GRU	0.6605	0.6763	0.6676	0.614	16.13%
Cyclic Gate LSTM – Square Wave	0.7122	0.7122	0.7122	0.6882	–2.75%
Cyclic Gate LSTM - Triangle Wave	0.7417	0.7626	0.6932	0.6838	8.90%
Cyclic Gate LSTM - Sawtooth Wave	0.6439	0.6691	0.6491	0.6	2.14%
Cyclic Gate LSTM - Square Sign Wave	0.688	0.7338	0.6954	0.6712	14.15%
Cyclic Gate LSTM - Simple Square Wave	0.7428	0.7698	0.728	0.7703	4.64%
Cyclic Gate LSTM - Simple Sawtooth	0.7033	0.7482	0.6991	0.6898	5.47%
Cyclic Gate GRU - Square Wave	0.7096	0.741	0.7168	0.6797	0.55%
Cyclic Gate GRU - Triangle Wave	0.7417	0.7626	0.6932	0.6728	1.92%
Cyclic Gate GRU - Simple Square Wave	0.7305	0.7626	0.7226	0.6739	14.01%

Table 4 EOG dataset results

Model	Precision	Recall	F1_Score	AUC	AUC Δ
RNN	0.2295	0.2818	0.2459	0.7481	0.70%
LSTM	0.4708	0.4448	0.4469	0.835	3.15%
GRU	0.3378	0.3508	0.3398	0.7924	4.25%
LSTM Missing Indicators	0.4382	0.4475	0.4321	0.8389	1.50%
GRU Missing Indicators	0.4813	0.4669	0.4676	0.839	1.63%
T-LSTM	0.4118	0.3978	0.3952	0.8172	–3.91%
GRU-D	0.4558	0.4558	0.449	0.8696	1.95%
Phased-LSTM	0.506	0.4834	0.4855	0.8575	1.17%
Phased-GRU	0.3428	0.337	0.337	0.7895	8.28%
Cyclic Gate LSTM – Square Wave	0.4705	0.442	0.4404	0.8637	–1.69%
Cyclic Gate LSTM - Triangle Wave	0.5007	0.4862	0.476	0.8575	–1.49%
Cyclic Gate LSTM - Sawtooth Wave	0.4851	0.4392	0.4419	0.8517	–0.52%
Cyclic Gate LSTM - Square Sign Wave	0.5368	0.5166	0.5214	0.8605	1.47%
Cyclic Gate LSTM - Simple Square Wave	0.5524	0.5359	0.5277	0.8877	–2.93%
Cyclic Gate LSTM - Simple Sawtooth	0.5108	0.4917	0.492	0.8791	–0.70%
Cyclic Gate GRU – Square Wave	0.3901	0.395	0.3821	0.8321	–0.04%
Cyclic Gate GRU - Triangle Wave	0.4593	0.4503	0.451	0.8509	1.19%
Cyclic Gate GRU - Simple Square Wave	0.5893	0.5829	0.5741	0.9095	–5.70%

Table 5 Internal bleeding dataset results

Model	Precision	Recall	F1_Score	AUC	AUC Δ
RNN	0.003	0.1429	0.0058	0.4764	0.0455
LSTM	0.0629	0.1974	0.092	0.7258	0.1051
GRU	0.2075	0.375	0.2035	0.9029	–0.0777
LSTM Missing Indicators	0.1637	0.1707	0.1351	0.7655	–0.004
GRU Missing Indicators	0.4511	0.5122	0.422	0.9481	–0.125
T-LSTM	0.0998	0.0781	0.0794	0.7103	0.0594
GRU-D	0.0605	0.0637	0.0595	0.715	0.1402
Phased-LSTM	0.2176	0.25	0.2019	0.7884	0.0747
Phased-GRU	0.0408	0.1875	0.0601	0.6657	0.1355
Cyclic Gate LSTM – Square Wave	0.149	0.2071	0.1591	0.7643	0.0136
Cyclic Gate LSTM - Triangle Wave	0.1271	0.1786	0.1244	0.8394	–0.1444
Cyclic Gate LSTM - Sawtooth Wave	0.0977	0.125	0.0987	0.7484	–0.0758
Cyclic Gate LSTM - Square Sign Wave	0.0859	0.1364	0.0774	0.7586	–0.0161
Cyclic Gate LSTM - Simple Square Wave	0.1193	0.2059	0.1307	0.8698	–0.0498
Cyclic Gate LSTM - Simple Sawtooth	0.1593	0.1742	0.1311	0.8388	–0.0382
Cyclic Gate GRU – Square Wave	0.5277	0.5585	0.5145	0.9594	–0.2274
Cyclic Gate GRU - Triangle Wave	0.2527	0.2981	0.1781	0.8768	0.0735
Cyclic Gate GRU - Simple Square Wave	0.2372	0.42	0.2632	0.8655	0.0747

Gate models average 0.892. The experimental results for this dataset show that for a non-periodic waveform, there is still a performance improvement from the Cyclic Gate RNNs over the baselines models and other variations of recurrent models.

The internal bleeding dataset includes a very long sequence length of 2000 timesteps per sample, containing periodic patterns within each sample, ranging from two to thirteen cycles. The sequence is relatively volatile and highly periodic. From the values in Table 5, the baseline LSTM and GRU have AUC values averaging 0.811, the T-LSTM and GRU-D have AUC values averaging 0.713, missing value indicator models average 0.857, while the top 3 Cyclic Gate models average 0.902. The best performing model on this dataset is the Cyclic Gate GRU with a Fourier square waveform, achieving the highest values on all metrics. The GRU with missing indicators also performs well on this dataset, achieving the next best set of performance metrics.

Reviewing the AUC results of each model across all datasets shows that the model with the highest average AUC is the Cyclic Gate LSTM with simple square waveform, followed by the Cyclic Gate GRU with simple square waveform and the Cyclic Gate GRU with Fourier square waveform. For the F1_score across all datasets, the model with the highest average value is the Cyclic Gate GRU with Fourier square waveform, followed by the Cyclic Gate GRU with simple square waveform and then the GRU with missing indicators. Of the 18 models experimented with across all datasets, the top 6 models with respect to the average AUC change between a complete dataset and a dataset with missing values were all Cyclic Gate models. The top three Cyclic Gate models for each dataset also had the lowest average negative impact on AUC values between a complete dataset and a dataset with missing values. This indicates that the Cyclic Gate models predominantly resulted in lower AUC drops on the introduction of high rates of missing values compared to alternate models tested.

The top three Cyclic Gate models identified by the average AUC across all the datasets were compared against conventional machine learning algorithms outside neural networks. KNN, SVM and RF algorithms were adopted for comparison against these Cyclic Gate RNN algorithms. The average AUC results across the four datasets are provided below in tabular and graphical representations in Table 6 and Fig. 3, respectively.

The comparison of the three leading Cyclic Gate RNN models with the non-NN models, as shown in Fig. 3, demonstrates a margin of higher AUC performance from the Cyclic Gate RNNs. The Random Forest algorithm performs marginally lower than the Cyclic Gate models, followed by the SVM and KNN models. The individual results in Table 6 show a general consistency in performance across each dataset, while the non-NN models show a higher variance in results across the datasets. For instance, the Random Forest algorithm performed best on the Internal Bleeding dataset, approaching values close to 100%; however, its relative performance in the other datasets was only moderate.

Table 6 Average AUC results across all the datasets using NN and non-NN models

Model	Datasets				
	ECG	EQ	EOG	IB	Average
Cyclic Gate LSTM—Simple Square	0.8450	0.7703	0.8877	0.8698	0.8432
Cyclic Gate GRU—Square	0.8976	0.6797	0.8321	0.9594	0.8422
Cyclic Gate GRU—Simple Square	0.8954	0.6739	0.9095	0.8655	0.8361
k-Nearest Neighbor—TS (KNN-TS)	0.9078	0.5045	0.7831	0.6275	0.7057
k-Nearest Neighbor—FE (KNN-FE)	0.7734	0.6411	0.6609	0.8333	0.7272
Support Vector Machine (SVM)	0.8857	0.5434	0.8697	0.7167	0.7539
Random Forest (RF)	0.8341	0.5651	0.8571	0.9896	0.8115

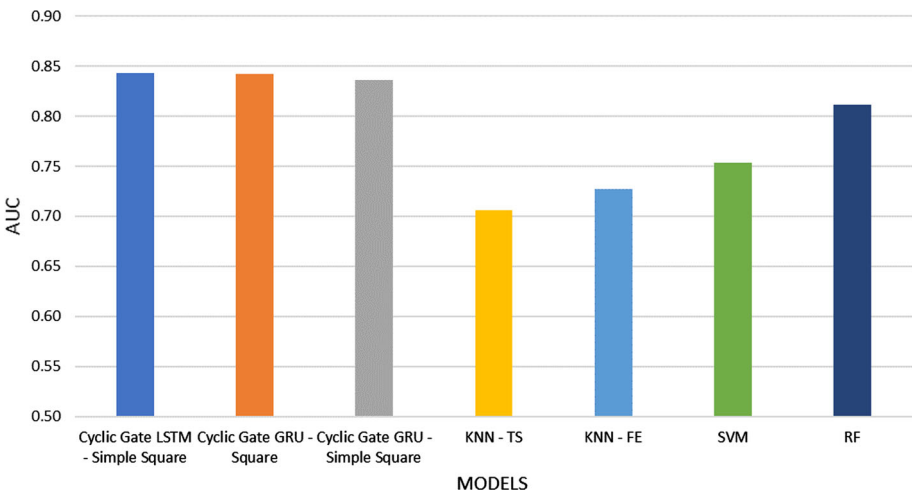


Fig. 3 Comparing average AUC results across all the datasets using NN and non-NN models

The preceding experimental results used the MCAR missing value generation mechanism with "Last Observation Carried Forward" (LOCF) replacement of Nan observations. In the next set of experiments, we validated the performance of the models using MAR and MNAR data generation mechanisms and included zero value imputation. The multivariate datasets with two or more features, Internal Bleeding Dataset and EOG Dataset, were used for these experiments. The results are shown in Table 7a–f.

The results in Table 7 present the top three performing Cyclic Gate models as well as the six comparative models for each dataset. The ability for Cyclic Gate models to exceed the performance of the comparative LSTM and GRU base models and their variants is maintained with experiments involving zero value imputation and MAR missing value generation, as shown in Tables 7a–d. However, for the MNAR missing values, the Cyclic Gate models provide comparable results but do not provide the best performance in both datasets, as shown in Tables 7e and f. The GRU with missing indicators provided higher AUC values than the Cyclic Gate models for MNAR missing values. The results show that MNAR data is particularly difficult to model and indicates the requirement for learning a joint distribution for both the data and the missingness mechanisms [54], which shall be further discussed in Sect. 5.

Figure 4 provides a visualisation of the resulting confusion matrix for the Internal Bleeding dataset with MAR missing value generation, comparing the Cyclic GRU—Fourier Triangle

Table 7 (a) Internal Bleeding Dataset, MCAR, 0 Value Imputation, (b) EOG Dataset, MCAR, 0 Value Imputation (c) Internal Bleeding Dataset, MAR, LOCF (d) EOG Dataset, MAR, LOCF (e) Internal Bleeding Dataset, MNAR, LOCF (f) EOG Dataset, MNAR, LOCF

Model	Precision	Recall	F1_Score	AUC
(a)				
Cyclic GRU- SS	0.1795	0.17	0.16	0.7878
Cyclic LSTM – SS	0.8725	0.9896	0.8176	0.7346
Cyclic GRU- FT	0.9989	0.7212	0.756	0.70543
LSTM	0.31283	0.4687	0.3231	0.5886
GRU	0.104	0.0944	0.0882	0.648
LSTM Missing Ind	0.0497	0.0739	0.0559	0.6369
GRU Missing Ind	0.0859	0.0677	0.0658	0.6425
T-LSTM	0.0769	0.0577	0.0583	0.5196
GRU-D	0.0964	0.0833	0.0829	0.6948
(c)				
Cyclic GRU - FT	0.494	0.4948	0.4546	0.9575
Cyclic GRU - FS	0.3709	0.4405	0.3819	0.9309
Cyclic GRU - SS	0.2954	0.4063	0.3096	0.929
LSTM	0.0782	0.0938	0.0678	0.7625
GRU	0.201	0.2635	0.184	0.8362
LSTM Missing Ind	0.0868	0.119	0.0898	0.7513
GRU Missing Ind	0.2755	0.4	0.2577	0.8792
T-LSTM	0.1455	0.1276	0.121	0.7661
GRU-D	0.1171	0.1224	0.1135	0.7685
(e)				
Phased GRU	0.3006	0.375	0.2825	0.9215
Cyclic GRU - FS	0.2438	0.3141	0.2266	0.9173
Cyclic GRU - SS	0.3014	0.3261	0.2667	0.8935
LSTM	0.1878	0.25	0.2022	0.8075
GRU	0.3035	0.4286	0.3233	0.9195
LSTM Missing Ind	0.1347	0.1786	0.1324	0.7902
GRU Missing Ind	0.4818	0.5284	0.4526	0.9584
T-LSTM	0.149	0.1615	0.1481	0.7926
GRU-D	0.1856	0.17	0.1538	0.8119

Model	Precision	Recall	F1_Score	AUC
(b)				
Cyclic GRU - SS	0.6178	0.605	0.5997	0.9181
Cyclic GRU - FT	0.5133	0.4834	0.4894	0.8589
Cyclic LSTM - SS	0.4071	0.4061	0.3893	0.8582
LSTM	0.2382	0.2293	0.23	0.7057
GRU	0.4132	0.4033	0.403	0.8097
LSTM Missing Ind	0.2258	0.2293	0.2247	0.7089
GRU Missing Ind	0.3882	0.3785	0.3784	0.786
T-LSTM	0.2579	0.2459	0.2488	0.7075
GRU-D	0.4953	0.4862	0.4796	0.8888
(d)				
Phased GRU	0.6036	0.5856	0.5907	0.8964
Cyclic LSTM - FT	0.5082	0.5028	0.4964	0.8759
Cyclic GRU - FS	0.5212	0.4807	0.4839	0.8687
LSTM	0.4946	0.4641	0.4646	0.854
GRU	0.401	0.3702	0.3755	0.8239
LSTM Missing Ind	0.395	0.384	0.3789	0.8161
GRU Missing Ind	0.4167	0.3923	0.395	0.8152
T-LSTM	0.3607	0.3508	0.3509	0.8191
GRU-D	0.4837	0.489	0.4835	0.8788
(f)				
Cyclic LSTM - FS	0.2864	0.3011	0.2891	0.7678
Cyclic LSTM - FT	0.3626	0.3591	0.3583	0.7647
Phased LSTM	0.3434	0.3481	0.3355	0.7613
LSTM	0.2642	0.2707	0.2562	0.7486
GRU	0.3055	0.2956	0.2922	0.752
LSTM Missing Ind	0.349	0.2928	0.2952	0.7906
GRU Missing Ind	0.3579	0.3508	0.3396	0.7896
T-LSTM	0.2673	0.2652	0.2595	0.7542
GRU-D	0.5117	0.489	0.4804	0.8662

SS = Simple Square, FS = Fourier Square, FT = Fourier Triangle

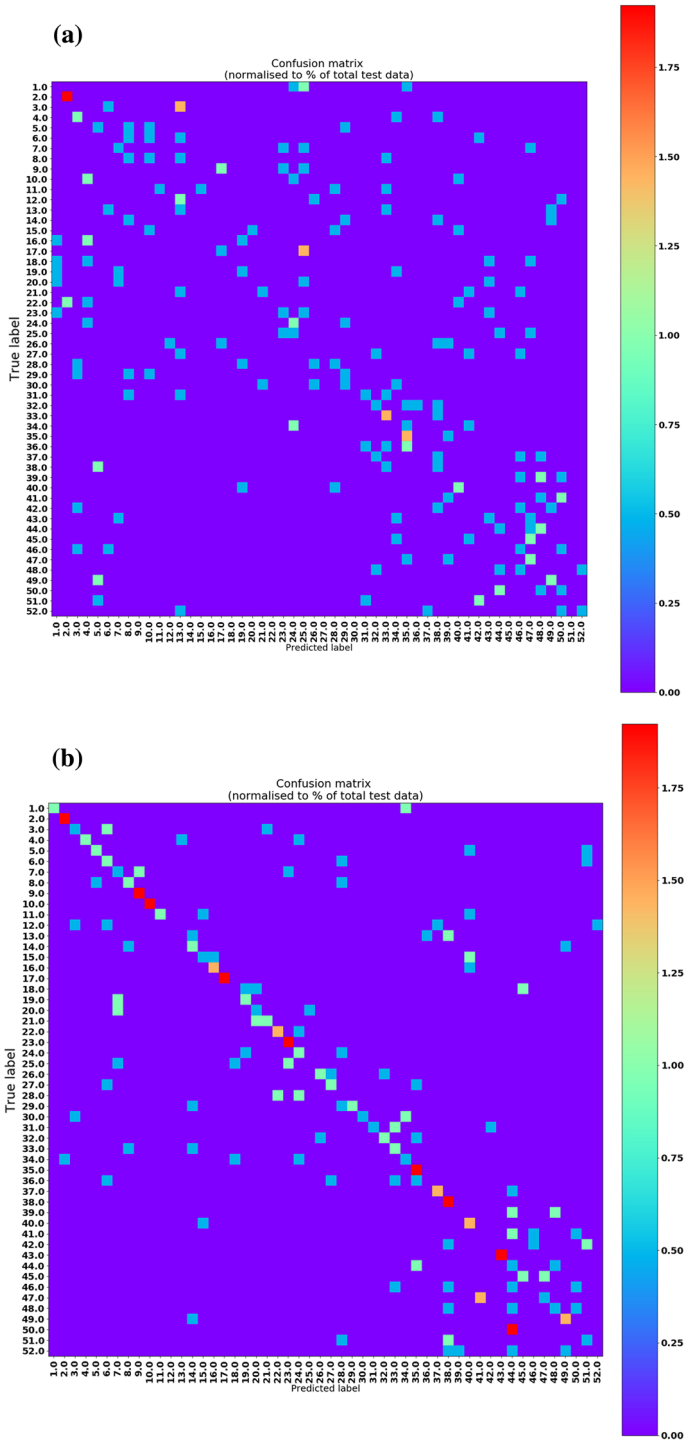
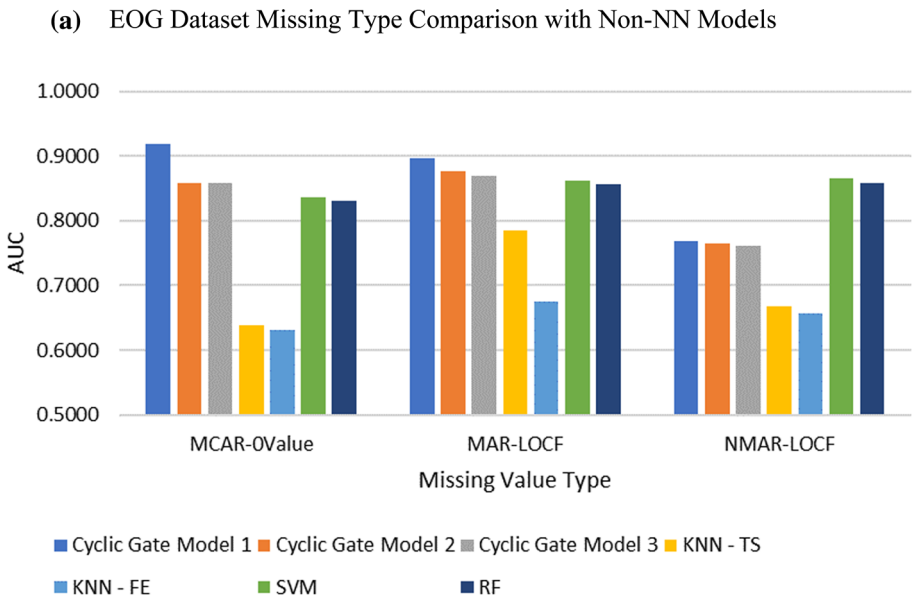


Fig. 4 **a** Confusion Matrix for LSTM model of Internal Bleeding dataset with MAR missing values (**b**) Confusion Matrix for Cyclic GRU—Fourier Triangle waveform model of Internal Bleeding dataset with MAR missing values



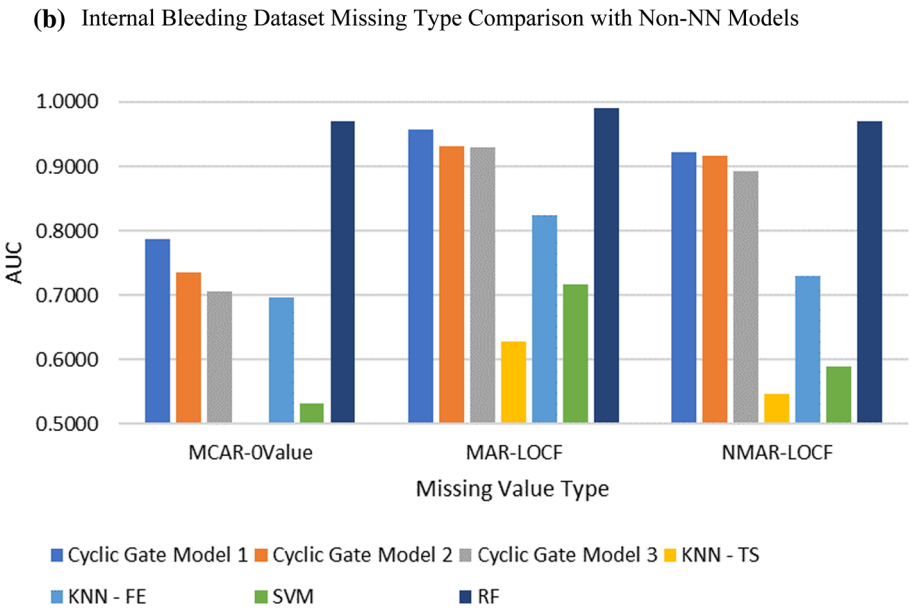


Fig. 5 **a** EOG Dataset Missing Type Comparison with Non-NN Models. **b** Internal Bleeding Dataset Missing Type Comparison with Non-NN Models

waveform to the LSTM model. Comparison of the matrices illustrates an example of the improved classification performance of a Cyclic Gate model over the baseline LSTM.

As a final comparison, we ran the EOG and Internal Bleeding datasets on the four non-NN models KNN-TS, KNN-FE, SVM and RF, using the same missing value generation mechanisms presented in Table 7. Figure 5a and b provide a graphical representation of the AUC values' results for each missing value type and algorithm. The results from the three leading Cyclic Gate RNN models in each sub-table of Table 7 are labelled as Models 1 to 3 and used for comparison against the non-NN models.

The comparative results for the EOG dataset show that the Cyclic Gate RNN models exceed the non-NN models for each missing value category, with the exception of the MNAR category, in which the SVM and RF algorithms perform better than the Cyclic Gate RNN models. For the Internal Bleeding dataset, the Cyclic Gate RNN models exceed the non-NN models, excluding the RF algorithm, which performed best in every category for this dataset. As identified in the earlier set of experiments, the classification of the Internal Bleeding dataset is handled exceptionally well by the Random Forest model due to its feature importance identification. Overall, the results shown in Fig. 5 are consistent with earlier results which demonstrated the superior performance of the Cyclic Gate RNN models on MCAR and MAR data, while their performance was lower on MNAR data.

5 Discussion

In our experimental datasets, there are a variety of periodicity levels, ranging from the EOG dataset which has very low periodicity, the ECG dataset which has a single period per sample, the Internal Bleeding dataset which has between 2 and 13 periods per sample and the Earthquakes dataset which contains up to 100 cycles per sample. The Cyclic Gate recurrent models are best suited to datasets with a high level of periodicity within a sample or where there are segments of data within samples that are significant to the classification objective and recur at consistent regions within the sample (i.e., with similar timestep values). In such cases, the Cyclic Gate models can identify data patterns or important data segments which minimise the impact of missing values. Where pattern occurrences are more random or important segments of data occur at irregular regions, the effectiveness of the cyclic gates will be reduced. The results show that the performance improvements of the Cyclic Gate models from the baselines GRU and LSTM cells and the more recent variants T-LSMT and GRU-D are more pronounced in the datasets with strong periodic behaviour. In datasets that have very low periodic behaviour within a sequence but have consistent behaviour across samples, the cyclic gates can still show an improvement over baseline and recent recurrent cell models when dealing with missing values in time series data.

The three general waveforms used by the Cyclic Gate models were the square wave, triangle wave and sawtooth wave. These waveforms had simple equation implementations as well as Fourier Series implementations. The difference between square and triangle waveforms in terms of gate activation is that the triangle wave gate has stages of opening and closing, which result in very fine control of the fully open period. In contrast, the square wave gate does not have opening stages and has a broader fully open period. This would indicate that where a data sequence was more dependent on very fine segments of important data and a portion of the immediately surrounding data, then the triangle waveform may be beneficial. The square wave would be advantageous if the data sequence was more dependent on a broader segment of important data. However, the ability to focus on specific sizes of data segments

is also addressed by the "Ratio On" and Period parameters. Therefore, it is unlikely that this difference in waveforms would result in a significant difference in performance on our selected datasets, which is reflected by our experimental results. The square wave performs marginally better than the triangle wave on all datasets except the ECG dataset, where the triangle wave performs slightly better. The sawtooth waveform does not perform as well as the square and triangle waveforms.

A general comparison of results was provided by experiments to show the relative performance of the Cyclic Gate RNNs to traditional machine learning algorithms. The results indicated that the AUC values of the proposed models showed better consistency across the datasets and a margin of improvement in performance. However, the Random Forest algorithm did perform exceptionally well on the Internal Bleeding (IB) dataset and yielded the best performance values. It was identified that the IB dataset contains a small set of extracted statistical features with a very strong correlation with the classification target. Given that the implemented RF model relies on feature extraction prior to classification and RF models have the ability to select these superior features better than alternate models [55], this is the likely reason the RF model performed well on this dataset. This specific result indicates that the RF algorithm may be more effective on certain datasets with extracted features that have strong correlations with classification outcomes. In contrast, the Cyclic Gate models will tend to be more effective on datasets with complex temporal patterns where extracted features have subtle correlations with classification targets or where missing values significantly degrade these correlations.

Experiments included evaluating the models against different missing value data generation mechanisms, MCAR, MAR and MNAR. With MNAR data, the missingness is related to unobserved variables and therefore, the distribution of the missing data cannot be ignored, unlike MCAR and MAR data. To properly handle MNAR data, there is a requirement for models to account for the inference based on observed data as well as the missing data in order to reduce estimate bias [56]. Therefore, modelling MNAR is generally more difficult than modelling MCAR and MAR data [57]. Although the Cyclic Gate models provide comparable results to alternate recurrent cell models and show some improvement over standard LSTM and GRU models, they do not present a clear advantage over all the recurrent cell variants. This outcome was also evident in experimental results for the KNN, SVM and RF models, where the Cyclic Gate models did not provide a clear advantage for MNAR data. It is expected that the Cyclic gate models, similar to the standard LSTM and GRU, require additional model architecture to account for the MNAR missing value mechanism. Therefore, the Cyclic Gate models by themselves will have limited advantages in this case. This is supported by the superior performance of recurrent models with missing indicators on MNAR data, which provide a degree of missing value modelling.

For all the experiments conducted, each neuron's phase shift, S, was uniformly chosen from the interval [0, T], where T was the period length. The parameters T and S were learnt during training, while the Ratio On parameter R was manually set during training. R was set at 0.5 for datasets with short period lengths (e.g., period < 20 timesteps) and at 0.1 otherwise. The model allows the R parameter to be learnt if required; however, for our experimentation results, we identified that allowing all of the waveform parameters to be learnt did not produce the best results. Therefore, R was set at fixed values to reduce the number of learnt parameters that defined the waveform.

It was identified that setting the initial period parameter with an appropriate value was an important step in producing an optimal outcome from the model. A heuristic approach was used to set the period parameter's starting value. The oscillation period was drawn uniformly from an exponential distribution $T \sim \text{Exp}(U(A, B))$, where A and B were initially selected

based on the dominant frequency identified from the discrete Fourier transform of the data sequence. For cases resulting in very large period lengths relative to the sequence length, there was too much loss of data from a limited gate opening time within the period length, and therefore the initial period length was reduced to less than 50 timesteps. Our experimentation investigated increasing the learning rate so that the initial period setting was not so significant; however, the resulting performance metrics were not optimal. We believe there is future work in this area associated with customised learning rates for the waveform parameters, which may reduce the dependency of the initial period value prior to learning.

For the Fourier series equations, we used 50 harmonics to define the series in each experiment. Initial experimentation also trialled 100 harmonics; however, any resulting increase in classification performance was not considered worth the additional processing time associated with the additional harmonics. Further increase in the number of harmonics above 100 was found to be impractical in terms of the extensive processing time required to perform classifications on the datasets. Similar to the number of harmonics used in the waveform equations, the hidden size of the recurrent cells as well as the associated gate size for the additional cyclic gates significantly impacted on the processing time of the models.

6 Conclusion

This paper proposes a modified gated recurrent cell with cyclic waveforms with learnt parameters to enhance the existing LSTM and GRU cells to better handle missing values. The proposed cell structure can learn an optimal waveform equation for rhythmic activation and inactivation of the cell state and hidden state gates, enabling timing discrimination to minimise the adverse effects of missing data. The model can potentially identify periodic segments of missing data which degrade a model's forecasting outcome or otherwise reduce the effect of randomly missing data within a sequence by effectively dropping out some of these missing values.

Experimental results on a series of sequence classification tasks with a high rate of missing data demonstrate that the proposed method can achieve superior performance metrics over baseline gated recurrent models and recent state of the art recurrent cell variants. The proposed models also show a general performance improvement and greater consistency across datasets over several traditional non-neural network models. The Cyclic Gate models provided the best performance on MCAR and MAR missing value generation mechanisms, as well as comparable results for MNAR data.

Our overall research findings indicate that Cyclic Gate models learn to select informative parts of the input and discard uninformative parts in a more finely tuned manner than current recurrent neural network architectures. Their performance in handling missing data is optimal when informative information is periodic instead of where informative observations are arbitrarily scattered across the time sequence. Our model adaptations are focused at the cell level and their resulting flexibility provides for integration into sophisticated high-level network structures and data augmentation with missing indicators to allow for further improvements in performance.

Although the periodic activation of our model results in a shorter gradient backpropagation path, it is acknowledged that the greater computation complexity associated with a Fourier Series generated waveform more than offsets any speed and efficiency benefits of shorter backpropagation. Future work will involve reducing this computation complexity with experimentation on the ideal number of harmonics in the Fourier Series and the size of

the wave equation learnt variables. Simple waveforms provide one form of implementation with lower computational complexity and impressive results, with the Cyclic Gate LSTM and GRU with a simple square wave being two of the leading models across the tested datasets.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Neil D, Pfeiffer M, Liu S-C (2016) Phased LSTM: accelerating recurrent network training for long or event-based sequences. In: Neural Inf Process Syst, pp. 3889–3897. http://papers.nips.cc/paper/by-source-2016-1928
2. Kwak SK, Kim JH (2017) Statistical data preparation: management of missing values and outliers. Korean J Anesthesiol 70:407–411. https://doi.org/10.4097/kjae.2017.70.4.407
3. Cao W, Wang D, Li J, Zhou H, Li L, Li Y (2018) BRITS: bidirectional recurrent imputation for time series. In: NIPS'18 Proc 32nd Int Conf Neural Inf Process Syst. pp. 6776–6786. http://papers.nips.cc/paper/by-source-2018-3408
4. Zhang Y, Thorburn P, Xiang W, Fitch P (2019) SSIM -a deep learning approach for recovering missing time series sensor data. IEEE Internet Things J 6:6618–6628. https://doi.org/10.1109/JIOT.2019.2909038
5. Dabrowski J, Rahman A (2019) Sequence-to-sequence imputation of missing sensor data. Australas Conf Artif Intell. https://doi.org/10.1007/978-3-030-35288-2_22
6. Luo Y, Cai X, Zhang Y, Xu J, Xiaojie Y (2018) Multivariate time series imputation with generative adversarial networks. In: Adv Neural Inf Process Syst 31 (NIPS 2018), Curran Associates, Inc. pp. 1596–1607. http://papers.nips.cc/paper/7432-multivariate-time-series-imputation-with-generative-adversarial-networks.pdf
7. Pham T, Tran T, Phung D, Venkatesh S (2016) DeepCare: a deep dynamic memory model for predictive medicine. In: PAKDD 2016 Proceedings, Part II, 20th Pacific-Asia Conf Adv Knowl Discov Data Min, Springer International Publishing, Cham, 2016: pp. 30–41. https://doi.org/10.1007/978-3-319-31750-2_3
8. Baytas IM, Xiao C, Zhang X, Wang F, Jain AK, Zhou J (2017) Patient subtyping via time-aware LSTM networks. In: Proc. 23rd ACM SIGKDD Int Conf Knowl Discov Data Min, ACM, New York, NY, USA, 2017: pp. 65–74. https://doi.org/10.1145/3097983.3097997
9. Little R, Rubin D (2014) Statistical analysis with missing data, 2nd edn. Wiley, Hoboken
10. Weerakody PB, Wong KW, Wang G, Ela W (2021) A review of irregular time series data handling with gated recurrent neural networks. Neurocomputing 441:161–178. https://doi.org/10.1016/j.neucom.2021.02.046
11. Che Z, Purushotham S, Cho K, Sontag D, Liu Y (2016) Recurrent neural networks for multivariate time series with missing values. Sci Rep 8:6085. https://doi.org/10.1038/s41598-018-24271-9
12. Lipton Z, Kale D, Wetzel R (2016) Modeling missing data in clinical time series with RNN. In: Proc 1st Mach Learn Healthc Conf. pp. 6776–6786. http://proceedings.mlr.press/v56/Lipton16.html
13. Lai G, Chang W-C, Yang Y, H. Liu H (2018) Modeling long- and short-term temporal patterns with deep neural networks. In: 41st Int ACM SIGIR Conf Res Dev Inf Retr. 2018: pp. 95–104. https://doi.org/10.1145/3209978.3210006
14. Choi E, Bahadori T, Sun J (2016) Doctor AI: predicting clinical events via recurrent neural networks. In: Proc 1st Mach Learn Healthc Conf 56:301–318. http://proceedings.mlr.press/v56/Choi16.html
15. Aydilek IB, Arslan A (2012) A novel hybrid approach to estimating missing values in databases using K-nearest neighbors and neural networks. Int J Innov Comput Inf Control 8:4705–4717

16. Tan Q, Ye M, Yang B, Liu S, Ma AJ, Yip TC-F, Wong GL-H, Yuen P (2020) DATA-GRU: dual-attention time-aware gated recurrent unit for irregular multivariate time series. Proc AAAI Conf Artif Intell 34:930–937. https://doi.org/10.1609/aaai.v34i01.5440
17. Li Q, Xu Y (2019) VS-GRU: a variable sensitive gated recurrent neural network for multivariate time series with massive missing values. Appl Sci 9:3041. https://doi.org/10.3390/app9153041
18. Andiojaya A, Demirhan H (2019) A bagging algorithm for the imputation of missing values in time series. Expert Syst Appl 129:10–26. https://doi.org/10.1016/J.ESWA.2019.03.044
19. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proc. 27th Int Conf Neural Inf Process Syst. 2:3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
20. Oh J, Chockalingam V, Singh S, Lee H (2016) Control of memory, active perception, and action in Minecraft. In: Proc. 33rd Int Conf Int Conf Mach Learn - Vol. 48, JMLR.org, 2016: pp. 2790–2799
21. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput. https://doi.org/10.1162/neco.1997.9.8.1735
22. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc 2014 Conf Empir Methods Nat Lang Process (2014) 1724–1734. https://doi.org/10.3115/v1/D14-1179
23. Chung J, Gülçehre Ç, Cho K, Bengio Y, Empirical evaluation of gated recurrent neural networks on sequence modeling, NIPS 2014 Work Deep Learn (2014) 9. http://arxiv.org/abs/1412.3555.
24. Malhotra P, Vig L, Shroff G, Agarwal P (2015) Long short term memory networks for anomaly detection in time series. In: Proceedings Eur Symp Artif Neural Networks, Comput. Intell. Mach. Learn. pp. 89–94
25. Hsu D (2017) Time series forecasting based on augmented long short-term memory, CoRR. http://arxiv.org/abs/1707.00666
26. Malhotra P, Vishnu T, Vig L, Agarwal P, Shroff G (2017) TimeNet: pre-trained deep recurrent neural network for time series classification. In: ESANN 2017 Eur Symp Artif Neural Networks, Comput. Intell. Mach. Learn. http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2017-100.pdf
27. Qin Y, Song D, Cheng H, Cheng W, Jiang G, Cottrell GW (2017) A dual-stage attention-based recurrent neural network for time series prediction. In: Proc 26th Int Jt Conf Artif Intell, AAAI Press, 2017: pp. 2627–2633. https://www.ijcai.org/Proceedings/2017/0366.pdf
28. Shukla SN, Marlin BM (2019) Interpolation-prediction networks for irregularly sampled time series. In: Int Conf Learn Represent. https://openreview.net/forum?id=r1efr3C9Ym
29. Wang X, Cai Z, Luo Y, Wen Z, Ying S (2022) Long time series deep forecasting with multiscale feature extraction and Seq2seq attention mechanism. Neural Process Lett. https://doi.org/10.1007/s11063-022-10774-0
30. Zhou G, Wu J, Zhang C, Zhou Z-H (2016) Minimal gated unit for recurrent neural networks. Int J Autom Comput 13:226–234. https://doi.org/10.1007/s11633-016-1006-2
31. Nina O, Rodriguez A (2015) Simplified LSTM unit and search space probability exploration for image description. In: 2015 10th Int Conf Information, Commun Signal Process. pp. 1–5. https://doi.org/10.1109/ICICS.2015.7459976
32. Hu J, Wang X, Zhang Y, Zhang D, Zhang M, Xue J (2020) Time series prediction method based on variant LSTM recurrent neural network. Neural Process Lett 52:1485–1500. https://doi.org/10.1007/s11063-020-10319-3
33. Jozefowicz R, Zaremba W, Sutskever I (2015) An empirical exploration of recurrent network architectures. In: 32nd Int Conf Mach Learn. https://doi.org/10.1109/CVPR.2015.7298761
34. Rahman L, Mohammed N, al Azad AK (2016) A new LSTM model by introducing biological cell state. In: 2016 3rd Int Conf Electr Eng Inf Commun Technol. pp 1–6
35. Pulver A, Lyu S (2017) LSTM with working memory. In: 2017 Int Jt Conf Neural Networks. pp. 845–851. https://doi.org/10.1109/IJCNN.2017.7965940
36. Mirza A (2018) Online additive updates with FFT-IFFT operator on the GRU neural networks. In: 2018 26th Signal Process Commun Appl Conf. pp. 1–4. https://doi.org/10.1109/SIU.2018.8404456
37. Zhou J, Huang Z (2018) Recover missing sensor data with iterative imputing network. In: Work 32 AAAI Conf Artif Intell. https://aaai.org/ocs/index.php/WS/AAAIW18/paper/download/17154/15570
38. Campos V, Jou B, Giró-i-Nieto X, Torres J, Chang S-F (2017) Skip RNN: learning to skip state updates in recurrent neural networks. In: Int Conf Learn Represent abs/1708.0. http://arxiv.org/abs/1708.06834
39. Saab S, Fu Y, Ray A, Hauser M (2021) A dynamically stabilized recurrent neural network. Neural Process Lett. https://doi.org/10.1007/s11063-021-10676-7
40. Carta A, Sperduti A, Bacciu D (2021) Incremental training of a recurrent neural network exploiting a multi-scale dynamic memory BT - machine learning and knowledge discovery in databases. In: Hutter F, Kersting K, Lijffijt J, Valera I (Eds) Springer International Publishing, Cham, 2021: pp. 677–693

41. Hayou S, Doucet A, Rousseau J (2019) On the impact of the activation function on deep neural networks training. In: Int Conf Mach Learn. J Mach Learn Res. https://arxiv.org/pdf/1902.06853.pdf
42. Shrestha A, Fang H, Wu Q, Qiu Q (2019) Approximating back-propagation for a biologically plausible local learning rule in spiking neural networks. In: Proc Int Conf Neuromorphic Syst Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/3354265.3354275
43. Sak H, Senior A, Beaufays F (2014) Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, ArXiv Prepr. arXiv:1402. https://arxiv.org/abs/1402.1128
44. Pascal Bugnion AK, Nicolas PR (2017) Scala: applied machine learning, 1st edn. Packt Publishing, Birmingham
45. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Bengio Y, LeCun Y (Eds) 3rd Int Conf Learn Represent {ICLR} 2015, San Diego, CA, USA, May 7–9, 2015, Conf. Track Proc., 2015. http://arxiv.org/abs/1412.6980
46. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE (2000) PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation 101:e215–e220
47. Bagnall E, Lines A, Vickers J, Keogh W (2018) The UEA & UCR time series classification repository. http://www.timeseriesclassification.com
48. Guillame-Bert M, Dubrawski A (2017) Classification of time sequences using graphs of temporal constraints. J Mach Learn Res 18:1–34
49. Fang F, Shinozaki T (2018) Electrooculography-based continuous eye-writing recognition system for efficient assistive communication systems. PLoS ONE 13:e0192684. https://doi.org/10.1371/journal.pone.0192684
50. Wang X, Zhang J, Xun L, Wang J, Wu Z, Henchiri M, Zhang S, Zhang S, Bai Y, Yang S, Li S, Yu X (2022) Evaluating the effectiveness of machine learning and deep learning models combined time-series satellite data for multiple crop types classification over a large-scale region. Remote Sens. https://doi.org/10.3390/rs14102341
51. Xi Y, Ren C, Tian Q, Ren Y, Dong X, Zhang Z (2021) Exploitation of time series sentinel-2 data and different machine learning algorithms for detailed tree species classification. IEEE J Sel Top Appl Earth Obs Remote Sens 14:7589–7603
52. Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, Darrell T (2017) Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans Pattern Anal Mach Intell 39:677–691. https://doi.org/10.1109/TPAMI.2016.2599174
53. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W, Woo W (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: NIPS'15 Proc 28th Int Conf Neural Inf Process Syst. pp. 802–810. https://doi.org/10.1093/toxsci/kfr046
54. Little RJA (1995) Modeling the drop-out mechanism in repeated-measures studies. J Am Stat Assoc 90:1112–1121. https://doi.org/10.2307/2291350
55. Menze BH, Kelm BM, Masuch R, Himmelreich U, Bachert P, Petrich W, Hamprecht FA (2009) A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinf 10:213. https://doi.org/10.1186/1471-2105-10-213
56. Rubin DB (1976) Inference and missing data. Biometrika 63:581–592. https://doi.org/10.2307/2335739
57. Johansson ÅM, Karlsson MO (2013) Comparison of methods for handling missing covariate data. AAPS J 15:1232–1241. https://doi.org/10.1208/s12248-013-9526-y