Computers and Electronics in Agriculture 189 (2021) 106377

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

A dual-head attention model for time series data imputation[☆]

Yifan Zhang[a,b,*], Peter J. Thorburn[a]

[a] Agriculture & Food, CSIRO, Brisbane, QLD 4067, Australia
[b] Queensland Alliance for Agriculture and Food Innovation, The University of Queensland, Brisbane, QLD 4072, Australia

ARTICLE INFO

Keywords:
Data imputation
Deep learning
Sequence-to-sequence
Water quality

ABSTRACT

Digital agriculture increasingly relies on the availability and accuracy of measurement data collected from various sensors. Of this data, water quality attracts great attention due to its intended use for crop irrigation, livestock, and other farming activities. Accurate and reliable water quality measurements enable farmers to understand the landscape comprehensively, optimising resource utilisation and reducing the negative impacts of agriculture on the environment. In practice, missing and incomplete data can create biased estimations and reduce the efficiency of many of the valuable applications provided by digital agriculture. The purpose of this paper is to propose a dual-head sequence-to-sequence imputation model (Dual-SSIM) designed to impute missing time series data in sensor networks, therefore reducing the negative consequences of missing and incomplete data. Unlike standard sequence-to-sequence architecture, the Dual-SSIM model features two encoders with gated recurrent units (GRUs) which are used to process temporal information before and after the missing gap separately. Furthermore, the attention mechanism is applied to two encoder outputs concurrently, in order to allow the model to focus on the high relative inputs when estimating missing data. The performance efficacy of Dual-SSIM has been investigated through the monitoring of water quality, sourced from an Australian water quality information system. Experimental results of this investigation indicate that Dual-SSIM outperforms associated alternatives based on the mean absolute error (MAE), root mean square error (RMSE), and dynamic time warping (DTW) scores in imputing two different water quality variables. Therefore, it can be concluded that Dual-SSIM provides an effective and promising approach for water quality data imputation.

1. Introduction

High-frequency monitoring data is becoming increasingly essential for prediction and decision making in application areas such as environmental protection, industrial control, and agricultural management (Zhang et al., 2019a; Zhang et al., 2020). For example, the timely monitoring of water quality is of great practical significance to aquaculture in regards to high yield, health and safety (Huan et al., 2020). Additionally, real-time water quality information can facilitate the immediate evaluation of recent farming practices, which can assist farmers in understanding the impact of cropping on water quality (Vilas et al., 2020). While advanced sensor technologies are currently widespread used for high-frequency monitoring, the eventuation of missing data is inevitable due to sensor failure and poor network connections.

Missing time series data is problematic as many statistical analyses require complete data sets. A simple solution to this issue is to omit the missing data; however, this approach may result in biased or erroneous analysis results (Mohamed et al., 2007). Additionally, missing data in the time series increases the challenge of identifying temporal patterns, especially when consecutive data points are lost. As a consequence, effective methods for estimating missing values based on available data are required.

Whilst various methods have already been developed for missing data imputation, these methods either require the knowledge of domain experts or cannot make full use of the temporal patterns inherent in the variables being monitored. For example, Nelsen et al. (2018) proposed an empirical mode-spatial model for environmental time series data imputation. Although this method performs well in imputing temperature measurements, it also expects the data to possess the recurring cycle or oscillations in time. In most scenarios quasi-periodic processes may not be obvious for non-expert users. Phan et al. (2020) proposed a dynamic time warping based approach to fill in large gaps within time series data. When evaluating their method on water level data acquired from France, a window with size T needs to be determined for searching

[☆] This work was conducted within the CSIRO Digiscape Future Science Platform.

[*] Corresponding author at: Agriculture & Food, CSIRO, Brisbane, QLD 4067, Australia.

E-mail addresses: yifan.zhang@uq.edu.au (Y. Zhang), peter.thorburn@csiro.au (P.J. Thorburn).

https://doi.org/10.1016/j.compag.2021.106377
Received 6 January 2021; Received in revised form 2 July 2021; Accepted 4 August 2021

the similar sub sequence. This also requires users to clearly understand the trend, seasonal and cyclical changes about the target time series.

In addition, data-driven models have recently attracted increasing attention in imputing missing data as they do not assume stationary or linear data, and rely less on domain knowledge (Betrie et al., 2016; Rahman et al., 2015; Hamzah et al., 2020). Betrie et al. (2016) investigated the regression imputation model, expectation-maximization algorithm and the covariance matrix calculation method to impute missing water quality data. Rahman et al. (2015) extended the K-nearest neighbour (KNN) and Fourier transform methods in imputing the biomedical time series data. The proposed method was evaluated using real-world biological datasets and achieved high imputation accuracy for different ratios of missing data. Hamzah (Hamzah et al., 2020) reviewed several infilling techniques that are convenient to time series analyses in streamflow. In their study, spline and linear interpolation are applied to the streamflow data. These methods tread streamflow data as a sequence of points. Nevertheless, none of these methods can capture the temporal patterns in time series datasets. Many studies (Yang et al., 2020; Ma et al., 2020) demonstrated that there exist latent patterns and dependencies between collected data in each time step within a time series. For example, the temperature is usually higher in summer and lower in winter. The repetition of these patterns can help infill the missing values in temporal cycles (Suo et al., 2020). Taking into account the temporal relations can boost the imputation results significantly (Luo et al., 2018).

Neural networks with deep architectures provide a powerful way of extracting nonlinear temporal patterns hidden in time series sequences. Applying deep learning models when infilling missing data alongside vast amounts of collected sensor data has attracted substantial attention in recent years.

Various studies have exploited the power of customised recurrent neural networks (RNNs) in handling time series data with missing values (Cao et al., 2018; Yoon et al., 2018; Che et al., 2018; Habiba and Pearlmutter, 2020). Cao et al. (2018) proposed a bidirectional recurrent neural network model for time series data imputation. In their approach, the imputed values are estimated and updated during the training process of the model. Similarly, Yoon et al. (2018) developed a multi-directional recurrent neural network that interpolates within data streams and imputes across data streams. In this proposal, the model imputed missing data more accurately than 11 benchmarks. In the study led by Habiba and Pearlmutter (2020), they combined Neural ODEs (Ordinary Differential Equations) with the Gated Recurrent Unit to predict the missingness of the information in continuous time-series data. Experiments on the PhysioNet dataset demonstrate the effectiveness of this architecture. It should be noted that most existing methods require extra masking vectors (denoting which variables are missing at each time index and maintaining the time interval for each variable since its last observation) to locate missing data in the time series. Additionally, these approaches require an independent classification task in order to guide the data imputation process. When a suitable classification task cannot be found, pure imputation accuracy can be significantly degraded.

Sequence-to-sequence models are a general end-to-end approach for processing sequential data. These models encode the input sequence with a series of RNNs and generate a variable length output with another set of decoder RNNs, both of which interface via an attention mechanism (Gehring et al., 2017). This architecture has been shown to outperform traditional, single RNN based architecture in various application areas such as machine translation (Tiwari et al., 2020), speech recognition (Nguyen et al., 2020) and text summarization (Shi et al., 2021).

In our previous work (Zhang et al., 2019b), the sequence-to-sequence model was initially designed for the recovery of variable-length missing data sequences in wireless sensor networks. The achieved SSIM promised accuracy in imputing missing values in time series sequences, however its capability was limited by the standard sequence-to-sequence architecture with a single input. In this case, zero value vectors are still needed to separate the available information between missing gaps.

In this paper, we have proposed a dual-head sequence-to-sequence imputation model (Dual-SSIM) for imputing missing water quality sensor data. Substantially extending on our preliminary studie (Zhang et al., 2019b), the model proposed in this paper has improvements in namely three aspects: model architecture, attention mechanism and loss function.

1. We have designed a sequence-to-sequence model with two encoders to process temporal input information. Each encoder is based on the gated recurrent unit (GRU) and deals with data from one side of the missing gap. Compared to our previous work, no extra information has been determined to locate the missing gaps, which may be a heavy burden for most comparative imputation methods; and
2. Based on the new model architecture, we have developed a cross-head attention mechanism which is concurrently applied across the outputs of two encoders. When imputing the missing data, the cross-head attention focuses on high relative pieces of input information in order to yield accurate estimations for missing sensor data.
3. When imputing missing values over a specified period of time, we not only expect the estimated missing values to have a low average error, but also to have a high similarity compared to the actual time series trajectory. Therefore, instead of using Mean Squared Error (MSE) as in previous studies, we have introduced the distortion loss including shape and time (DILATE) loss function in our model as proposed by Guen (Vincent and Thome, 2019).
4. In the end, we have visualised the attention score as a methodology for model interpretation. Instead of applying the neural network model as a black-box tool, the visualisation provides a practical and intuitive explanation of the model's predictions.

The paper is organized as follows. Section 2 describes multi-step data imputation problems and the challenges. Then, the dual-head sequence-to-sequence imputation model is presented in Sections 3. Section 4 presents experimental results. Finally, Section 5 concludes the paper.

2. Multi-step data imputation problem

Handling a singular piece of missing data within a time series is straightforward. As evidenced in many studies (Du et al., 2017; Park and Kim, 2020), the linear/polynomial imputation can achieve promising results. Challenges emerge when the data is missing for extended periods of time (Zhang et al., 2019b). As the size of the missing gap increases, the performance of many imputation methods drops significantly (Moffat et al., 2007). Benefiting from advanced sensor technologies, a large amount of data is collected at a high velocity. This high frequency data collection has the potential to result in large gaps in the data if the network collapses. Therefore, we have focused on recovering consecutive missing data points within a time series.

The general problem of multi-step imputation is depicted in Fig. 1. In a multivariate time series, a consecutive number of data points are missing. An imputation model is expected to recover the missing data points with the help of all the available data on both sides of the gap.

To formulate the imputation problems, we define the multivariate time series $\mathbf{X}$ with the constant time interval as follows:

$$\mathrm{X} = \{x^1, \ldots x^n\}^\intercal = \{x_1, \ldots, x_T\} \in \mathbb{R}^{n\times T} \tag{1}$$

where $x^i = \{x^i_1, \ldots, x^i_T\}^\intercal \in \mathbb{R}^T$ is the ith time series and $x_t = \{x^1_t, \ldots, x^n_t\}^\intercal \in \mathbb{R}^n$ represents the vector of n time series at time step t.

Assuming k numbers of consecutive data points are missing within the time series X, the missing data M starts at time index $p+1$. It can be represented by

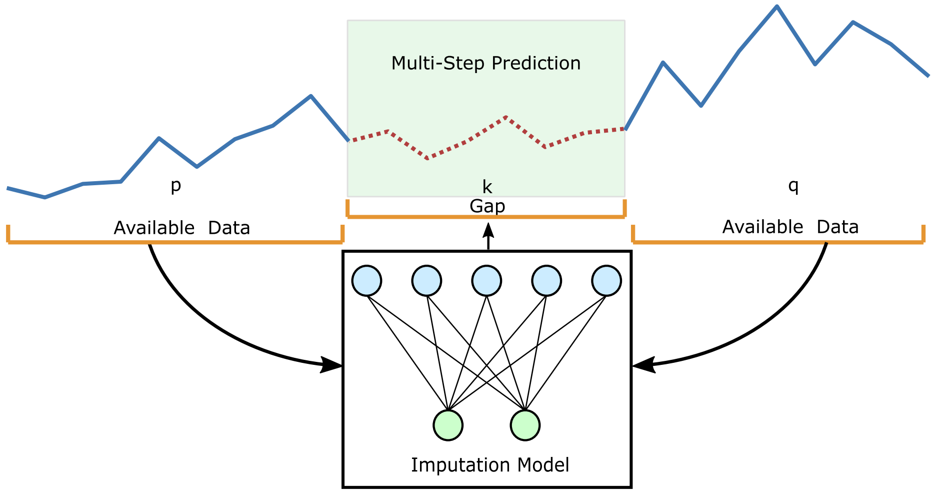


Fig. 1. Illustration of multi-step imputation problem. Dotted points represent the missing data in the time series. There are available data on both sides of the missing gap. p, k and q denote the number of data points for the left side of the gap, the missing data gap, and the right side of the gap, respectively.

$$\mathrm{M} = \left\{x^1_{p+1}, \ldots, x^1_{p+k}\right\} \in \mathbb{R}^{1\times k} \tag{2}$$

As illustrated in Fig. 1, data around the missing gap include valuable information to support predicting the missing data points. Let $\mathrm{L}_{\mathrm{available}}$ and $\mathrm{R}_{\mathrm{available}}$ represent the remaining data on the left and right side of the gap, they are described in the following equation:

$$\mathrm{L}_{\mathrm{available}} = \left\{x_1, \ldots, x_p\right\} \in \mathbb{R}^{n\times p}, \tag{3}$$

$$\mathrm{R}_{\mathrm{available}} = \left\{x_{p+k+1}, \ldots, x_{p+k+q}\right\} \in \mathbb{R}^{n\times q}, \tag{4}$$

where p and q indicate the size of the corresponding available time series. n represents the number of variables measured at each time step.

Hence, an imputation model is required to predict the missing values based on all the available data. We can formulate the prediction as

$$\widehat{\mathrm{M}} = \mathrm{Model}\left(\mathrm{L}_{\mathrm{available}} \cup \mathrm{R}_{\mathrm{available}}\right) \in \mathbb{R}^{1\times k} \tag{5}$$

For the above imputation problem, we plan to follow the supervised training paradigm to obtain the imputation model. As illustrated in Fig. 2 left part, a sliding window strategy is applied to prepare training data. For each training sample, a size k sequence is set as the target, and the model has two input sequences with size p and q. The performance of supervised learning highly relies on the quality of training datasets (Engelbrecht and Brits, 2002). Hence, it is necessary to prepare the complete and cleaned sensor data.

Benefit from the neural network architectures, when applying the imputation model to unseen data, the size of the model's inputs and outputs can differ from what is in the training process. In the right part of Fig. 2, we showed two different instances. In the top right example, there are enough available data between the gap, and the model accepts the size of input as it is fed during the training. In many scenarios like the right bottom example, the gap size and available data may vary. The imputation model can still handle these cases with the recurrent network design.

In the following section, a deep learning-based imputation model is proposed to predict a sequence of missing data $\widehat{\mathrm{M}}$ and the detailed model architecture is explained.

3. Proposed dual-head sequence-to-sequence imputation model

In this section, we propose a dual-head sequence-to-sequence model with the cross-head attention mechanism to impute missing water quality sensor data. Moreover, advanced training strategies are also proposed to help our model achieve the expected performance.

The Dual-SSIM extends the conventional sequence-to-sequence architecture as depicted in Fig. 3. The dual encoders process the input

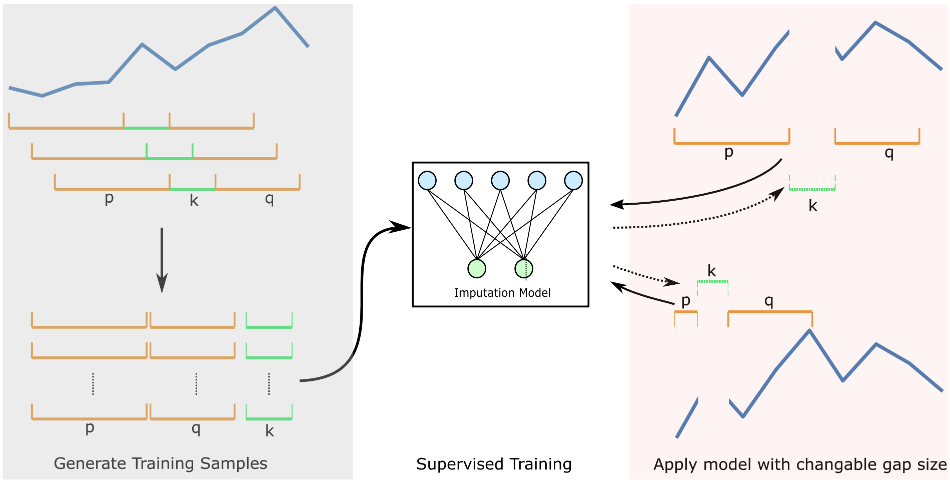

p
k
q
p
q
k
Generate Training Samples
Imputation Model
Supervised Training
p
q
k
k
p
q
Apply model with changable gap size

Fig. 2. Supervised Imputation Model with Changeable Gap Size. The left part explains how to generate imputation training samples. The right part presents two different scenarios in infilling missing data with different sizes.

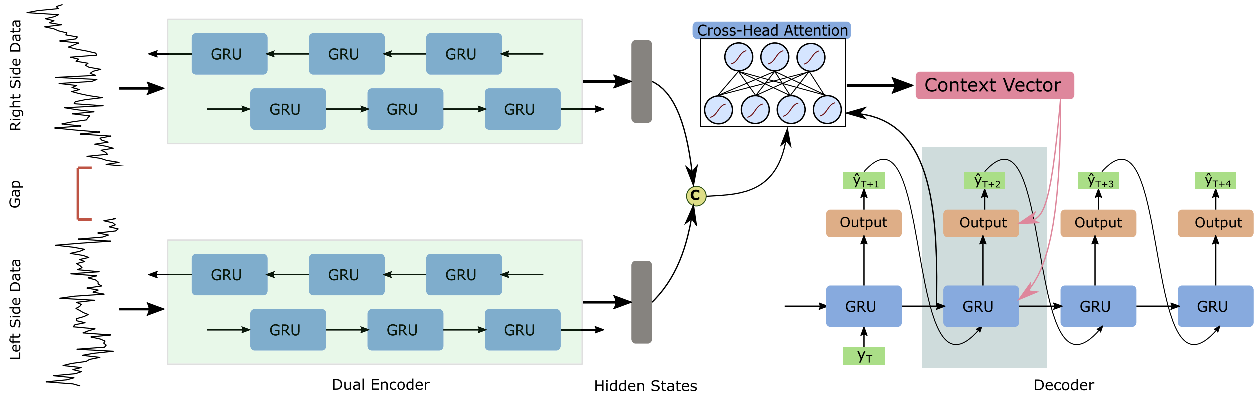


Fig. 3. Dual-SSIM architecture. The dual encoders are based on the bidirectional GRU. Each encoder is responsible for input data from one side of the gap. The hidden states of both encoders are concatenated and processed by the cross-head attention module. A Linear layer is stacked on top of the GRU decoder to generate numeric values. The grey box in the decoder highlights a predictive step when decoding the information passed from the encoder.

sequence around the missing gap, which can split the information before and after the missing gap naturally. In addition, the decoder with the cross-head attention mechanism can focus on the more relative pieces of the input information for different predictions.

3.1. Dual encoders with GRU

For the data imputation problem, each gap has available data around. Both the available information before and after the gap can contribute to the imputation task. Hence, we designed two encoders to process the input information from each side of the gap, separately. In this approach, the place of the gap within the time series input can be naturally positioned. No additional masking vectors or specific recurrent architectures are required to identify gaps.

In the proposed model, the Gated Recurrent Unit (GRU) is chosen to process time series inputs in the dual-head encoder. GRU is a specific type of recurrent neural network proposed by Cho et al. (2014). Compared with long short-term memory (LSTM) Schuster and Paliwal (1997), GRU with simplified structure can reduce the training parameters and speed up the convergence while ensuring the memory ability of the neurons, thereby improving the prediction accuracy. The internal information flow of the GRU unit can be expressed by the following formula:

$$\mathrm{h}_t = \left(1 - \mathrm{z}_t\right) \odot \mathrm{h}_{t-1} + \mathrm{z}_t \odot \widetilde{\mathrm{h}}_t, \tag{6}$$

$$\widetilde{\mathrm{h}}_t = \tanh\left(W_h \mathrm{x}_t + U_h\left(\mathrm{r}_t \odot \mathrm{h}_{t-1}\right) + \mathrm{b}_h\right), \tag{7}$$

$$\mathrm{z}_t = \sigma(W_z \mathrm{x}_t + U_z \mathrm{h}_{t-1} + \mathrm{b}_z), \tag{8}$$

$$\mathrm{r}_t = \sigma(W_r \mathrm{x}_t + U_r \mathrm{h}_{t-1} + \mathrm{b}_r), \tag{9}$$

where z_t and r_t are the update and reset gates of the GRU, respectively. tanh, σ and $\odot$ represent the tanh activation function, the sigmoid function and the element-wise multiplication.

Modelling the temporal information in both directions can significantly improve the performance of the recurrent network. Compared to the standard GRU, the bidirectional GRU combines the forward hidden layer and the backward hidden layer, which can access both the preceding and succeeding contexts (Liu and Guo, 2019). Hence, we choose to apply the bidirectional GRU unit in the dual-head encoder.

In the bidirectional GRU, the hidden states at time index i are concatenated as

$$\mathrm{h}_i = \left[\overrightarrow{\mathrm{h}_i}; \overleftarrow{\mathrm{h}_i}\right]. \tag{10}$$

Let $\mathrm{Encoder}_l$ and $\mathrm{Encoder}_r$ denote the GRU encoder for the input sequence on the left and right side of the missing gap, respectively. The output of each encoder is a sequence of hidden states $\mathrm{H} = \{\mathrm{h}_1, \mathrm{h}_2, \ldots, \mathrm{h}_n\}$, where n is the length of the input sequence.

3.2. Decoder with cross-head attention

Unlike the conventional sequence-to-sequence model with single encoder, the model proposed in this paper has twin encoders embedded. When making predictions for the future time index, the attention mechanism used in this model should be able to pick up the high relative information from the input sequences processed by both encoders.

Hence, we extend the global attention model proposed by (Luong et al., 2015) and make it support processing temporal representations learned from two different encoders (Fig. 4). According to the previous section, each encoder has a sequence of hidden states H generated. Therefore, in order to make use of all the temporal representation learned by both encoders, we concentrate these hidden states as the new input to the decoder as follow:

$$\mathrm{Decoder}_{\mathrm{input}} = \left[\mathrm{H}_l; \mathrm{H}_r\right], \tag{11}$$

where H_l and H_r represent the output from $\mathrm{Encoder}_l$ and $\mathrm{Encoder}_r$, respectively.

In the decoder, the predictions of the missing values are generated successively. At each time index t, the GRU unit is updated based on the previous states, prediction at time index $t-1$ and the attention vector c_t. A linear layer is stacked on top of the GRU layer to generate the numeric value. The above procedures are computed as follows:

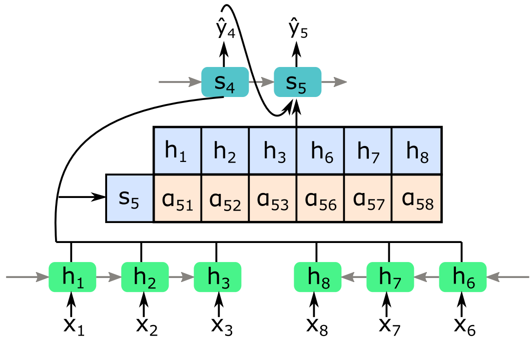


Fig. 4. Cross-head attention mechanism. The example shows how the attention module works when estimating the missing data at time index 4 and 5. The input data $\{x_1, x_2, x_3\}$ and $\{x_6, x_7, x_8\}$ are feed into two encoders, respectively. The hidden states learned by the dual encoders are joint to a single vector, and the corresponding attention scores are calculated based on the correlation between the hidden states $\{h_1, \ldots, h_8\}$ and the states of the decoder s_5.

$$y_t = \text{Linear}(W[s_t; c_t] + \text{b}), \tag{12}$$

$$\text{s}_t = \text{GRU}(y_{t-1}, \text{s}_{t-1}, \text{c}_t), \tag{13}$$

where s_t is the hidden state of the decoder at time index t, c_t is the attention context vector, and $[s_t; c_t]$ is a concatenation of the decoder hidden state and the context vector. The linear layers product the final prediction y_t.

In each decoding time index t, the attention context vector c_t can be described as a weighted sum of the hidden states passed by the dual-head encoder:

$$\text{c}_t = \sum_{i=1}^{n} \alpha_{ti} \text{h}_i, \text{h}_i \in \text{Decoder}_{\text{input}}. \tag{14}$$

The weight α_{ti} of each hidden states h_i is computed by

$$e_{ti} = a(\text{s}_{\text{t}-1}, \text{h}_\text{i}), \tag{15}$$

$$\alpha_{ti} = \text{softmax}(e_{ti}), \tag{16}$$

where e_{ti} represents the correlation between the hidden states around h_i and the output at time t. a is a neural network that can be jointly trained with the GRU decoder. A softmax activation function is applied to e_{ti} to ensure that the sum of all the attention weights is normalised to 1.

According to the cross-head attention mechanism described from (14)–(16), the decoder can reweight the input information based on the attention score as demonstrated in (12). Hence, the proposed model can figure out the most relevant information from the input sequences when predicting the value for missing data at different time index. Moreover, the attention mechanism provides an efficient way to interpret and visualize what information the model is looking at while generation predictions. These explanations are highly required when applying data-driven models in solving real-world problems. A thorough analysis of the performance of the cross-head attention mechanism will be conducted in Section 4.3.

3.3. *Enhanced training strategy*

The training stage can widely determine the success of a neural network application. A well designed neural network architecture may achieve poor performance because of improper training strategies (Tang et al., 2016). In this section, we introduce two strategies to optimize the training of Dual-SSIM architectures.

3.3.1. *Scheduled sampling*

As described in Section 2, sensor data usually have missing data points during a period of time. Therefore, the Dual-SSIM needs to generate predictions iteratively until the missing gap is filled. The predictive bias occurred at each time index during the imputation task could downgrade the predictive accuracy considerably.

Hence, we applied the similarly scheduled sampling strategy (Bengio et al., 2015) to improve the stability and accuracy of multi-step prediction for the Dual-SSIM.

When training the model to yield a prediction at time index t, we choose to use the true previous observation y_{t-1} with probability ε, or use the estimated $\widehat{y_{t-1}}$ coming from the model itself with probability $1 - \varepsilon$. During inference, the Dual-SSIM predicts values only depending on its own previously predicted values. This process is illustrated in Fig. 5.

By applying the scheduled sampling, the discrepancy between the training and inference can be mitigated. It leads to an imputation model that is more robust to correct its own mistakes at inference as it has learned to do so during training.

3.3.2. *Shape and temporal aware loss*

Loss functions are one of the most critical parts of training accurate machine learning models. The loss function can significantly affect the ability of the model to produce optimum results as one expects. (Liang et al., 2018). Mean Squared Error (MSE) and Mean Absolute Error (MAE) are applied by the vast majority of methods for regression tasks (Cuturi and Blondel, 2017). When imputing multiple missing values in a time series, we not only expect the estimated missing values have a low average error but also have a high similarity to the actual time series trajectory. Hence, we applied the distortion loss including shape and time (DILATE) proposed by Guen (Vincent and Thome, 2019) in our model.

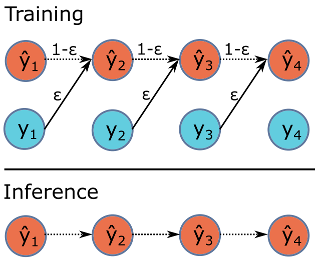


Fig. 5. Scheduled Sampling with Fixed Probability. In the training process, the model random decides to use the true previous observation or one estimated form the model itself based on the probability factor ε. On the contrary, the model only predicts the next step using its own predicted values in inference.

Let $\widehat{y}$ and $y \in \mathbb{R}^k$ denote the predicted and actual time series of length k. The DILATE loss $\mathfrak{L}(\widehat{y}, y)$ is designed to compare the prediction $\widehat{y}$ with the actual time series y as

$$\mathfrak{L}(\widehat{y}, y) = \alpha \mathfrak{L}_{\text{shape}}(\widehat{y}, y) + (1 - \alpha)\mathfrak{L}_{\text{temporal}}((\widehat{y}, y)) \tag{17}$$

where $\alpha \in [0, 1]$ is a hyperparameter used to balance two loss terms $\mathfrak{L}_{\text{shape}}$ and $\mathfrak{L}_{\text{temporal}}$.

Shape Term

The shape loss function $\mathfrak{L}_{\text{shape}}$ is based on the Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978), which can be formulated to the following optimization problem:

$$DTW\left(\widehat{y}, y\right) = \min_{A \in \mathscr{A}_{k,k}} \left\langle A, \Delta\left(\widehat{y}, y\right)\right\rangle \tag{18}$$

where the binary matrix $A \subset \{0, 1\}^{k \times k}$ is a warping path with $A_{i,j} = 1$ if $\widehat{y}_i$ is associated to y_j, and 0 otherwise. $\mathscr{A}_{k,k}$ is the set of all valid warping paths connecting the endpoints $(1, 1)$ to $(k,\ k)$. $\Delta(\widehat{y}, y) = [\delta(\widehat{y}_i, y_j)]_{i,j}$ represents the pairwise cost matrix, where δ is a given dissimilarity between $\widehat{y}_i$ and y_j, e.g. the euclidean distance.

The DTW loss in (18) focuses on the structural shape dissimilarity between the predicted $\widehat{y}$ and ground truth y. To make the DTW differentiable, the smooth min operator proposed in (Cuturi and Blondel, 2017) is applied to define the differentiable shape term $\mathfrak{L}_{\text{shape}}$:

$$\begin{aligned} \mathfrak{L}_{\text{shape}}(\widehat{y}, y) &= DTW_\gamma(\widehat{y}, y) \\ &= -\gamma log\left(\sum_{A \in \mathscr{A}_{k,k}} \exp\left(-\frac{\langle A, \Delta(\widehat{y}, y)\rangle}{\gamma}\right)\right) \end{aligned} \tag{19}$$

Temporal Term

The second term $\mathfrak{L}_{\text{temporal}}$ in (17) aims at penalizing temporal distortions between $\widehat{y}$ and y. Inspired from computing the Time Distortion Index (TDI) for temporal misalignment estimation (Frías-Paredes et al., 2017), the smoothed temporal loss is defined as:

$$\mathfrak{L}_{\text{temporal}}\left(\widehat{y}, y\right) = \frac{1}{Z} \sum_{A \in \mathscr{A}_{k,k}} \left\langle A, \Omega \right\rangle \exp^{-\frac{\langle A, \Delta(\widehat{y, y})\rangle}{\gamma}} \tag{20}$$

where Z is the partition function that $Z = \sum_{A \in \mathscr{A}_{k,k}} \exp^{-\frac{\langle A, \Delta(\widehat{y, y})\rangle}{\gamma}}$. Ω is a square matrix of size $k \times k$ penalizing each element $\hat{y}_i$ being associated to an y_j, for $i \neq j$. Here, Ω is chosen as a squared penalization that $\Omega\left(i, j\right) = \frac{1}{k^2}(i-j)^2$.

DILATE combines two terms for precise shape and temporal localization of time series with sudden changes. The imputation problem described in Section 2 requires to predict multiple values across a period of time. Hence, this loss function can be a proper choice for processing multi-step imputation tasks.

4. Evaluation

In this section, the predictive accuracy of the Dual-SSIM is evaluated by using the water quality sensor data collected by a water quality monitoring network in Australia.

4.1. Water quality monitoring network

The Great Barrier Reef Catchment loads monitoring program is a large-scale water quality monitoring program that helps track long-term trends in water quality entering the Great Barrier Reef lagoon from adjacent catchments along the east coast of Queensland (AU, 2018). The program monitors all intensive land use catchments. It includes 43 monitored sites across 20 key catchment areas for monitoring sediments and nutrients, and 20 sites for pesticides.

Each monitoring station has various water quality sensors deployed. For example, the acoustic Doppler current profiler (ADCP) is installed to measure the discharge and streamflow. All the monitoring data are collected automatically into a cloud-based data monitoring platform for further analysis. Data used in this study can be find from Kaggle water quality dataset (QLD, 2020).

4.1.1. Water quality data

Data were collected from an in-situ monitoring station in the Mulgrave-Russell catchment in the Great Barrier Reef, Australia (Fig. 6). Influenced by both the natural processes and anthropogenic interferences, the quality of river water is highly heterogeneous for different variables (Ishaq et al., 2012). For instance, conductivity from this station locates in a vast range, the minimum value is close to 0, and the maximum value is over 50000 ($\mu S/cm$). Similar situations can be found on other variables such as nitrate and turbidity. This is usually caused by heavy rainfall in a short period. Hence, data normalization is essential before applying any imputation algorithms. In this study, we rescaled all the data in range $[0, 1]$.

Fig. 7 illustrates two water quality variables monitored in this station. Observation showed that though water level rises and falls during the month, changes in water level apparent in a daily pattern. On the contrary, the temporal patterns of in-stream nitrate concentration cannot be identified in the daily or weekly scale (Neal et al., 2006). The different temporal variations bring great challenges in designing data-driven deep learning models.

Nitrate in creeks and rivers can harm aquatic and marine ecosystems, and reliable information on nitrate concentrations are needed to manage the problem (Vilas et al., 2020). An important parameter is the nitrate load, calculated from nitrate concentration and flow data, and missing data can hinder the calculation of loads. Hence, we choose to recover missing data for water level (an input to flow) and nitrate concentration.

4.1.2. Data preprocessing

All the water quality variables monitored from 2019 were used as inputs in this experiment (Table 1). Followed by the cross-validation strategy, we trained the model on three quarter' data and validated the model on the remaining quarter's data. All the values are normalized in the range of $[0, 1]$.

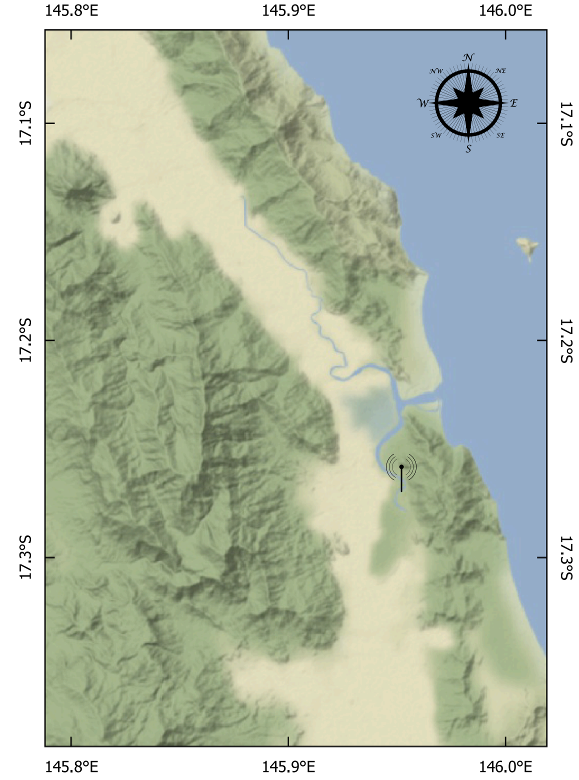


Fig. 6. Water quality monitoring station in the Mulgrave-Russell catchment, Australia. The icon represents the in-situ monitoring station located in the upstream of Russell River.

As mentioned in Fig. 2, we use the sliding window algorithm to generate all the training/test samples in the experiment. Each sample includes k missing data as target gap, p available data on the left size of the gap and q available data on the right size of the gap as we described in Eqs. (2) and (3). For example, for a time series with 23 time index $\{x_1, \ldots, x_{23}\}$, we choose $\{x_1, \ldots, x_{10}\}$ and $\{x_{14}, \ldots, x_{23}\}$ as two inputs, $\{x_{11}, \ldots, x_{13}\}$ as the target for supervised training.

Errors and anomalies are commonly found in real-time water quality monitoring. Many imputation methods are sensitive to outliers (Van Zoest et al., 2021). Therefore, it is critical to detect and remove outliers before applying imputation algorithms.

In this study, we applied three data filtering algorithms to remove the obvious outliers in the data steams (Vilas et al., 2020). A threshold filter removed the negative and extreme large values. A sensor reference filter fixed the sensor-related measurement errors. In addition, we also applied a changing rate filter to remove the measurements that have significant changes in a short period. The water quality expert configures all three data filtering algorithms to fit this specific dataset.

4.2. Parametrisation and benchmarks

We evaluate the performance of recovering missing data based upon the root mean square error (RMSE), mean absolute error (MAE) and dynamic time warping (DTW).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left|f_i - \hat{f}_i\right|\right)^2}, \tag{21}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - \hat{f}_i|, \tag{22}$$

$$DTW = \sqrt{\sum_{(i,j)\in P} \| f_i - \hat{f}_j \|^2}, \tag{23}$$

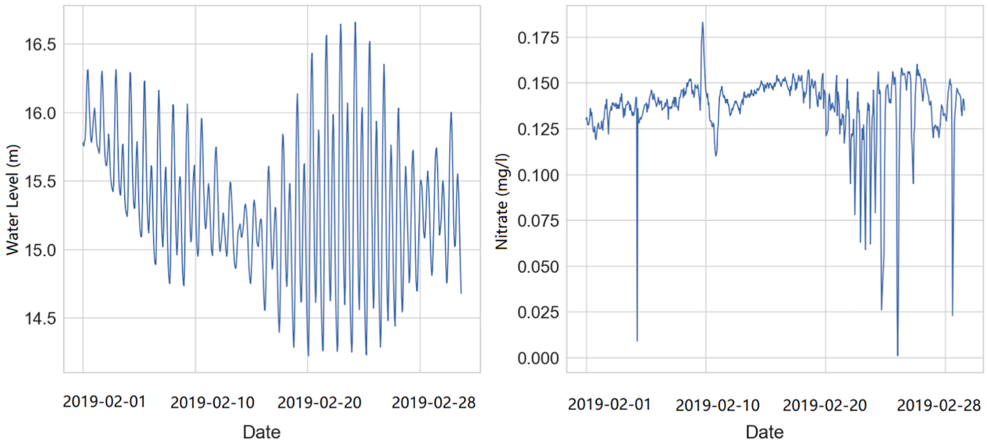


Fig. 7. The trend of two water quality variables during February, 2019. Water Level fluctuates in the daily pattern, while the changing pattern of nitrate concentration cannot be identified in the daily or weekly time scale.

Table 1
Hourly water quality data during 2019.

Parameters	Unit	Min	Max	Mean	SD
Water Temperature	°C	18.6	32.2	24.9	2.8
Water Level	m	14.0	17.0	15.1	0.5
Water discharge	m^3/s	−247.7	670.2	75.1	108.6
Conductivity	μS/cm	0.1	50825.8	3740.4	7607.9
Turbidity	NTU	0.5	124.3	5.8	5.6
Nitrate	mg/l	0.001	1.7	0.2	0.3

where P is the optimal alignment path between time series.

We also compared our model with the following five data imputation methods.

- EM. Expectation Maximization (EM) scheme in (Ghomrawi et al., 2011) is a probabilistic imputation method, which calculates maximum likelihood estimates from incomplete data set.
- KNN. K-nearest neighbour (KNN) imputation (Beretta and Santaniello, 2016) is designed to find k nearest neighbours to the observation with missing data and then impute them based on the non-missing values in the neighbours.
- SSIM. SSIM (Zhang et al., 2019b) is designed to impute time series sensor data with consecutive missing values. It is based on sequence-to-sequence architecture with global attention mechanisms.
- BRITS. BRITS (Cao et al., 2018) is a recurrent neural network based method for missing value imputation in time series data.
- M-RNN. M-RNN(Yoon et al., 2018) is a Multi-directional Recurrent Neural Network that interpolates within data streams and imputes across data streams. It provides a promising estimation of missing measurements by applying to five real-world medical datasets.

The optimized hyperparameters for the water quality sensor data are also shown in Table 2.

In this study, we applied a grid search over all hyperparameters for all neural network-based models. In detail, we tested the number of layers from 1 to 3 for both the encoder and decoder. In addition, the number of GRU units are tested in the range [25,50,75]. For imputation methods such as EM and KNN, we use the recommended parameter settings provided in the impyute package (Impyute, 2019).

Table 2
Hyperparameters of the Dual-SSIM.

Hyperparameters	Value
No. of Hidden Layers for Dual Encoder	1
No. of Hidden GRU Units per Layer	50
No. of Hidden Layers for Decoder	1
No. of Hidden GRU Units per Layer	50
Optimizer	AdamW
Batch Size	10

For SSIM, BRITS and M-RNN, we implemented the models on PyTorch platform (Paszke et al., 2017). In addition, limited by the BRITS and M-RNN's design, we only use water level or nitrate as the input for the corresponding imputation task. For KNN and EM, we used the implementations provided by impyute package (Impyute, 2019). We tested the proposed Dual-SSIM on the CSIRO Accelerator Cluster with Nvidia P100 GPU and 64 GB RAM.

4.3. Experimental results and discussion

In real-world scenarios, missing data happens randomly during the monitoring. The recurrent based encoder and decoder design in Section 3 promise the capability of our proposed model in generating imputation results with variable lengths. Beside this, the number of available data around the missing values can also be adjusted based on the user's configurations. Hence, the Dual-SSIM is designed to deal with the arbitrary size of data gaps in the time series.

In this experiment, to make a fair and consistent comparison between all the imputation methods, we choose to evaluate models on the fixed gap size with a constant number of data surrounded. All the algorithm evaluated in this study can be extended to support more flexible gap sizes.

Based on the analysis of the water quality monitoring data described in Table 1, over 90 % of the missing gaps have a size of less than 3. Hence, we choose to infill gaps with sizes 3 and 6, respectively. Furthermore, to cover helpful information near the gap, we use 10 data from the left side of the gap and 10 data from the gap's right side as all the model's input. The input size can also be changed based on the temporal patterns of the targeting variable.

With the above settings, the evaluated model can be suitable for recovering most of the gaps and also handle the larger gaps as well. Beside this, all the input data are preprocessed as described in Subsection 4.1.2.

Table 3 illustrates the normalized imputation performance for both imputing missing water level and nitrate data. As we can see, the proposed Dual-SSIM model achieved the best performance for RMSE, MAE and DTW scores in all the imputation tasks. For instance, when infilling the size 3 gap for water level, the Dual-SSIM obtained the scores of 0.015, 0.013 and 0.026 for RMSE, MAE and DTW, accordingly. Moreover, the variance of the performance is very low in these three test cases (± 0.001), which means the proposed Dual-SSIM is robust and ensures good generalization behaviour in practice (Markatou et al., 2005).

Table 3
Imputation Accuracy For Imputing Two Water Quality Variables With Different Gap Size.

Target	Gap	Metric	Dual-SSIM	SSIM	BRITS	M-RNN	EM	KNN
WaterLevel	3	RMSE	**0.015 (±0.001)**	0.045 (±0.02)	0.067 (±0.004)	0.164 (±0.006)	0.277 (±0.044)	0.312 (±0.125)
		MAE	**0.013 (±0.001)**	0.042 (±0.019)	0.052 (±0.004)	0.136 (±0.007)	0.216 (±0.04)	0.263 (±0.115)
		DTW	**0.026 (±0.001)**	0.073 (±0.032)	0.343 (±0.024)	0.290 (±0.027)	0.405 (±0.076)	0.47 (±0.201)
	6	RMSE	**0.026 (±0.006)**	0.057 (±0.01)	0.121 (±0.012)	0.221 (±0.002)	0.275 (±0.045)	0.44 (±0.004)
		MAE	**0.023 (±0.006)**	0.052 (±0.01)	0.092 (±0.01)	0.159 (±0.002)	0.216 (±0.04)	0.39 (±0.012)
		DTW	**0.058 (±0.013)**	0.114 (±0.018)	0.361 (±0.026)	0.367 (±0.012)	0.531 (±0.099)	1.01 (±0.013)
Nitrate	3	RMSE	**0.041 (±0.034)**	0.084 (±0.067)	0.107 (±0.098)	0.189 (±0.02)	0.183 (±0.085)	0.232 (±0.154)
		MAE	**0.037 (±0.03)**	0.079 (±0.063)	0.07 (±0.065)	0.142 (±0.004)	0.113 (±0.055)	0.15 (±0.106)
		DTW	**0.068 (±0.055)**	0.145 (±0.116)	0.229 (±0.209)	0.529 (±0.042)	0.216 (±0.104)	0.269 (±0.187)
	6	RMSE	**0.041 (±0.034)**	0.098 (±0.083)	0.117 (±0.107)	0.269 (±0.009)	0.194 (±0.089)	0.552 (±0.166)
		MAE	**0.037 (±0.03)**	0.089 (±0.076)	0.077 (±0.072)	0.233 (±0.044)	0.122 (±0.058)	0.483 (±0.227)
		DTW	**0.068 (±0.055)**	0.236 (±0.201)	0.262 (±0.24)	0.703 (±0.046)	0.316 (±0.148)	1.206 (±0.537)

Beside this, SSIM achieved the second best performance for most of the imputation tasks. For example, it outperformed BRITS, M-RNN, EM and KNN in recovering water level with both gap size 3 and 6 for all three performance criteria.

BRITS and M-RNN did not perform well in all four test cases, especially when the gap size is large. According to the BRITS and M-RNN's design, they used the multi-task learning approach to improve the imputation accuracy by applying the corresponding classification task. While in most scenarios, it is very hard to create a meaningful clarification task based on the collected time series data. For the pure imputation task, the accumulated predictive errors among consecutive outputs degrade these models' performance significantly.

Compared to machine learning-based models, EM and KNN performed poorly in recovering both water level and nitrate data. For example, EM had 0.275 RMSE scores in estimating water level data with gap size 6, and EM got 0.552 RMSE scores when processing nitrate data with gap size 6. The major drawback for these methods is that they ignored the temporal information, which could be very helpful in estimating the tendency of water quality.

Fig. 8 highlighted the performance improvement of the proposed Dual-SSIM as opposed to the other five imputation methods in four test cases. It is obvious that the Dual-SSIM performed nearly 20 times better than KNN in regarding recover the missing water level data with gap size 3 and 6 for all three criteria. Also, the DTW scores of Dual-SSIM are

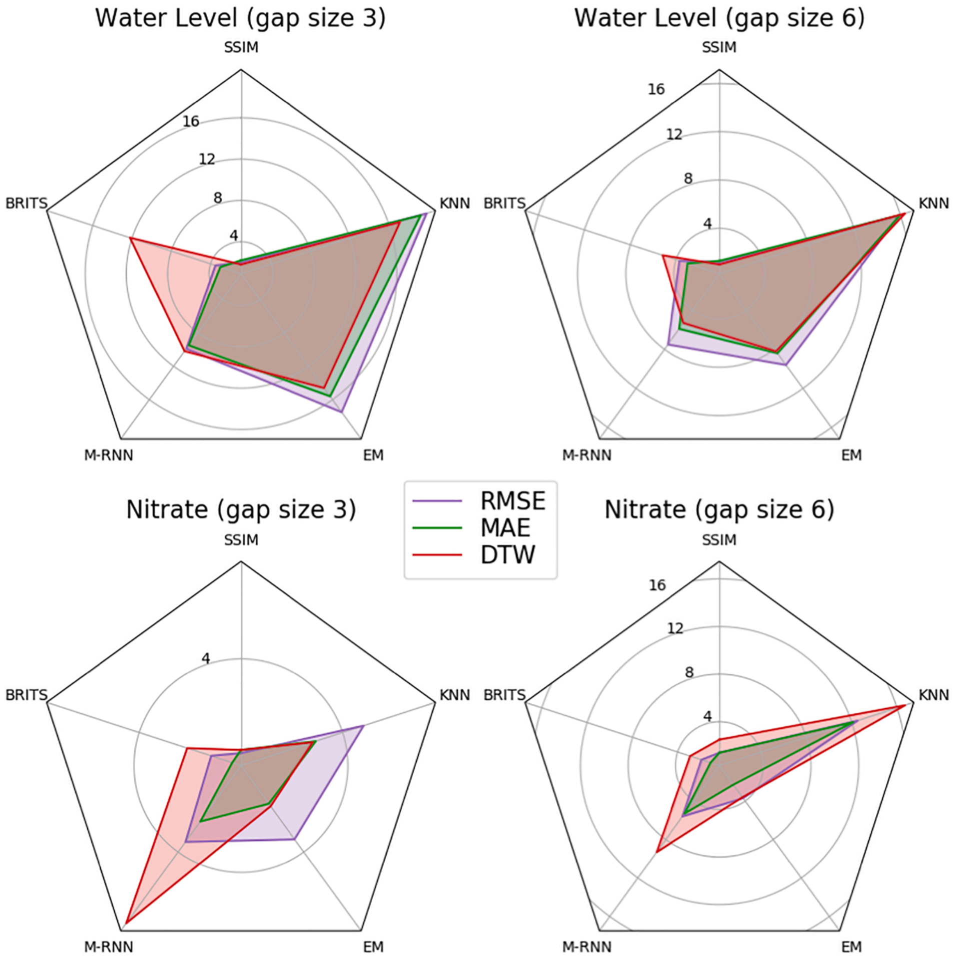


Fig. 8. The performance improvements of the proposed Dual-SSIM as opposed to the other five imputation methods in four test cases. Value at each spoke represents how much the Dual-SSIM can overperformance the corresponding benchmark method. (Unit 100%).

around 12 times better than that of BRITS in recovering water level with gap size 3, and 8 times better than that of M-RNN in recovering nitrate data with the same gap size.

We also plotted the detailed imputation results for all the listed models. Fig. 9a and 9b show how all the models performed when imputing 3 consecutive missing water level measurements.

In Fig. 9a, 3 consecutive measurements around the peak area were missed. Overall, Dual-SSIM, SSIM and BRITS can recover the missing data with the correct temporal pattern. While M-RNN, EM and KNN cannot capture the ascending and descending trends adequately. By feeding the available information before and after this gap (solid red line), the proposed Dual-SSIM recovered the missing data with the highest accuracy (dark blue line). This demonstrates the operational effectiveness of the proposed model architecture in Fig. 3.

In addition, we also picked up a descending period to test all the models (Fig. 9b). In this case, the water level measurements were decreasing, which is a more straightforward case compared to that in Fig. 9a. Dual-SSIM, SSIM, BRITS, KNN can offer the imputations with the downtrend. EM and M-RNN generated biased imputations.

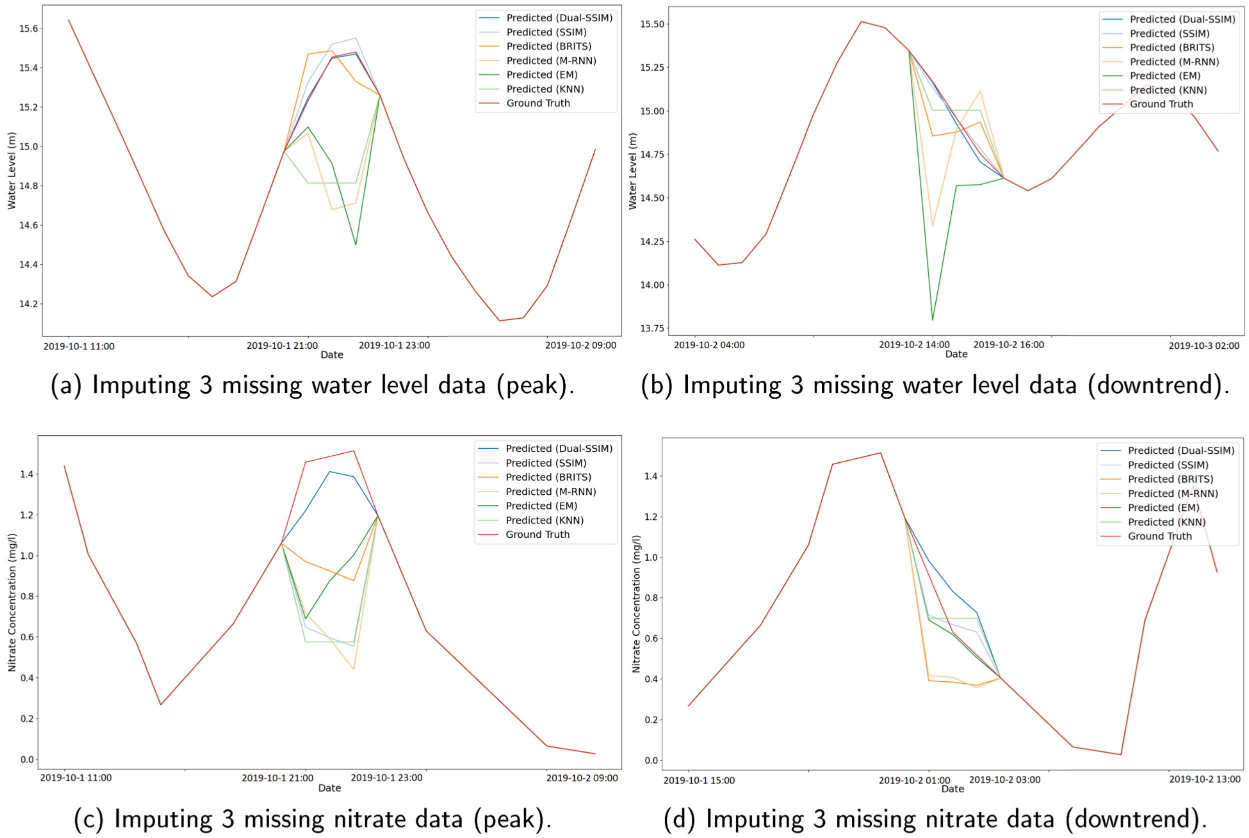
In Fig. 9c and d, we did the same test on the nitrate data. Considering nitrate measurements do not have a recognised daily or weekly pattern (Fig. 7), it can be more difficult to recover missing nitrate measurements. In Fig. 9c, only Dual-SSIM can offer proper imputations when missing measurements happened in the peak region. When the data has an apparent changing trend (Fig. 9d), most imputation models can generate promising results.

Fig. 10 exemplifies how the attention mechanism works when imputing missing nitrate measurements. The proposed Dual-SSIM model pays more attention to the inputs from the previous time than that from the future time. It is reasonable that the nitrate concentration variates among the time and the measures in the near past contribute considerably to the prediction of the missing value. Moreover, it is noticeable that the Dual-SSIM model also pays much attention to the first few inputs from the right side of the gap. This proves our design in Section 3 that feeding available data in the future time index of the gap can provide useful information in recovering the missing data. The available data in the future time index can guide the predictions of the Dual-SSIM and reduce the predictive bias accumulated through multiple time index. In addition, the attention scores vary between these two examples. This indicates that the model can focus on different parts of the inputs dynamically, which proves the cross-head attention mechanism contributes significantly to generate imputation value. Overall, the attention visualization gives us insight into how the inputs data from different time index are utilized by our model and proves the efficiency of the dual-head design.

5. Conclusion

Water quality measurements have been widely used to interpret current situations and trends in the water system, and support decision-makers in agricultural activities such as the use of irrigation, pesticides, and fertilisers. This paper proposes a dual-head sequence-to-sequence model (Dual-SSIM) for water quality sensor data imputation. In order to naturally support the time series data with missing gaps, two encoders with the gated recurrent unit have been designed to process the temporal information. In addition, an attention module has been designed to calculate the attention score by crossing the hidden states from two encoders.

We have evaluated all imputation models on a real-world dataset collected from a water quality monitoring system deployed in Australia. Experimental results demonstrated that Dual-SSIM outperforms other benchmarks such as EM, KNN, SSIM, BRITS and M-RNN. In imputing missing nitrate and water level values, Dual-SSIM achieved the best scores of RMSE, MAE and DTW in both imputation cases with gap sizes 3



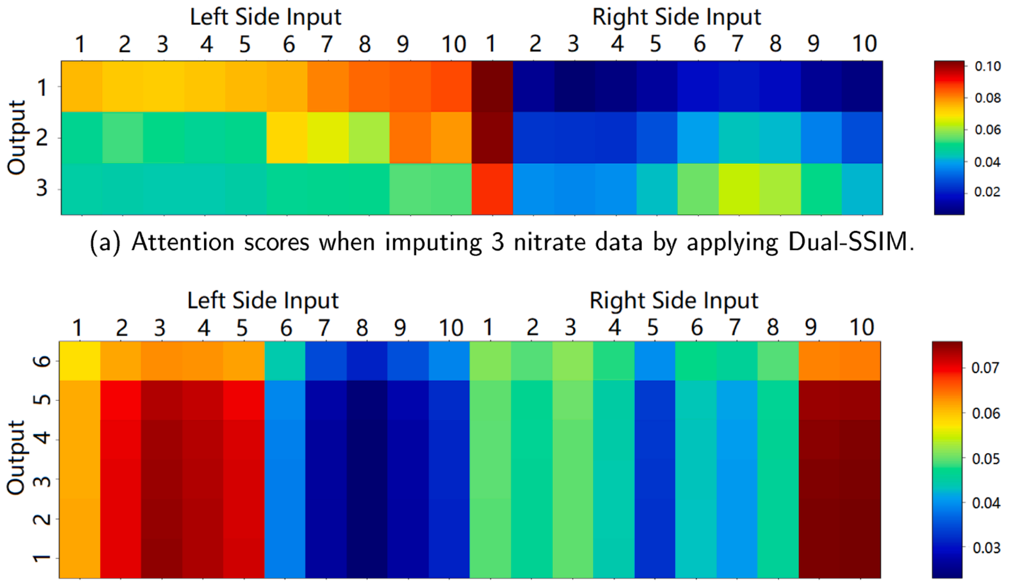
(a) Imputing 3 missing water level data (peak).

(b) Imputing 3 missing water level data (downtrend).

(c) Imputing 3 missing nitrate data (peak).

(d) Imputing 3 missing nitrate data (downtrend).

Fig. 9. Model performance in imputing 3 consecutive missing measurements for both water level and nitrate. The solid red line is the ground truth measurements. Other colours represent the imputation results generated by different models. 20 available data before and after the gap are used as the model's input.



(a) Attention scores when imputing 3 nitrate data by applying Dual-SSIM.

(b) Attention scores when imputing 6 nitrate data by applying Dual-SSIM.

Fig. 10. Examples of attention mechanism visualization. In the top and bottom examples, the Dual-SSIM has to impute 3 and 6 nitrate data points, respectively. There are 10 left side inputs and 10 right side inputs for the model. The colour in each block denotes the calculated attention weight for input at each time index. The warmer the colour is, the higher the attention score is.

and 6. When checking the estimated predictions, the outputs of Dual-SSIM matched the temporal changing patterns in both peak and non-peak periods precisely. Consequently, this model could be successfully used to impute missing time-series measurements, thereby helping in water quality monitoring and management.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank the Great Barrier Reef Catchment Loads Monitoring Program for providing valuable real-time water quality monitoring data sets. This research work was supported by the Kaggle Open Data Research Grant.

References

AU. Reef 2050 water quality improvement plan. https://www.reefplan.qld.gov.au/ (accessed: 2018-07-20).
Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N., 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 1171–1179.
Beretta, L., Santaniello, A., 2016. Nearest neighbor imputation algorithms: a critical evaluation. BMC Med. Informat. Decision Making 16, 74.
Betrie, G.D., Sadiq, R., Tesfamariam, S., Morin, K.A., 2016. On the issue of incomplete and missing water-quality data in mine site databases: Comparing three imputation methods. Mine Water Environ. 35, 3–9.
Cao, W., Wang, D., Li, J., Zhou, H., Li, L., Li, Y., 2018. Brits: bidirectional recurrent imputation for time series. In: Advances in Neural Information Processing Systems, pp. 6775–6785.
Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y., 2018. Recurrent neural networks for multivariate time series with missing values. Sci. Rep. 8, 6085.
Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
Cuturi, M., Blondel, M., 2017. Soft-dtw: a differentiable loss function for time-series. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR.org, pp. 894–903.
Du, H., Xiao, Y., Duan, L., Gao, S., 2017. An algorithm for vessel's missing trajectory restoration based on polynomial interpolation. In: 2017 4th International Conference on Transportation Information and Safety (ICTIS). IEEE, pp. 825–830.
Engelbrecht, A.P., Brits, R., 2002. Supervised training using an unsupervised approach to active learning. Neural Process. Lett. 15, 247–260.
Frías-Paredes, L., Mallor, F., Gastón-Romeo, M., León, T., 2017. Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors. Energy Convers. Manage. 142, 533–546.
Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N., 2017. Convolutional sequence to sequence learning,. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, pp. 1243.–1252.
Ghomrawi, H.M., Mandl, L.A., Rutledge, J., Alexiades, M.M., Mazumdar, M., 2011. Is there a role for expectation maximization imputation in addressing missing data in research using womac questionnaire? comparison to the standard mean approach and a tutorial. BMC Musculoskeletal Disorders 12, 109.
Habiba, M., Pearlmutter, B.A., 2020. Neural odes for informative missingess in multivariate time series. In: 2020 31st Irish Signals and Systems Conference (ISSC). IEEE, pp. 1–6.
Hamzah, F.B., MohdHamzah, F., Razali, S.F.M., Jaafar, O., AbdulJamil, N., 2020. Imputation methods for recovering streamflow observation: A methodological review. Cogent Environ. Sci. 6, 1745133.
Huan, J., Li, H., Wu, F., Cao, W., 2020. Design of water quality monitoring system for aquaculture ponds based on nb-iot. Aquacult. Eng. 90, 102088.
Impyute. https://impyute.readthedocs.io/ (accessed: 2019-11-20).
Ishaq, S.E., Agada, P.O., Rufus, S., 2012. Spatial and temporal variation in water quality of river Benue, Nigeria. J. Environ. Prot.
Liang, S., Sun, R., Li, Y., Srikant, R., 2018. Understanding the loss surface of neural networks for binary classification. arXiv preprint arXiv:1803.00909.
Liu, G., Guo, J., 2019. Bidirectional lstm with attention mechanism and convolutional layer for text classification. Neurocomputing 337, 325–338.
Luo, Y., Cai, X., Zhang, Y., Xu, J., Yuan, X., 2018. Multivariate time series imputation with generative adversarial networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 1603–1614.
Luong, T., Pham, H., Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. In: Proc. 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp. 1412–1421.
Ma, Q., Li, S., Cottrell, G., 2020. Adversarial joint-learning recurrent neural network for incomplete time series classification. IEEE Trans. Pattern Anal. Mach. Intell.
Markatou, M., Tian, H., Biswas, S., Hripcsak, G., 2005. Analysis of variance of cross-validation estimators of the generalization error. J. Mach. Learn. Res. 6, 1127–1168.
Moffat, A.M., Papale, D., Reichstein, M., Hollinger, D.Y., Richardson, A.D., Barr, A.G., Beckstein, C., Braswell, B.H., Churkina, G., Desai, A.R., et al., 2007. Comprehensive comparison of gap-filling techniques for eddy covariance net carbon fluxes. Agric. For. Meteorol. 147, 209–232.
Mohamed, A.K., Nelwamondo, F.V., Marwala, T., 2007. Estimating missing data using neural network techniques, principal component analysis and genetic algorithms. In: Proceedings of the Eighteenth Annual Symposium of the Pattern Recognition Association of South Africa.
Neal, C., Jarvie, H.P., Neal, M., Hill, L., Wickham, H., 2006. Nitrate concentrations in river waters of the upper Thames and its tributaries. Sci. Total Environ. 365, 15–32.
Nelsen, B., Williams, D., Williams, G., Berrett, C., 2018. An empirical mode-spatial model for environmental data imputation. Hydrology 5, 63.
Nguyen, T.S., Stueker, S., Niehues, J., Waibel, A., 2020. Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation. In: ICASSP

2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 7689–7693.

Park, J., Kim, S., 2020. Improved interpolation and anomaly detection for personal pm2. 5 measurement. Appl. Sci. 10, 543.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch.

Phan, T.T.H., Caillault, Émilie Poisson, Lefebvre, A., Bigand, A., 2020. Dynamic time warping-based imputation for univariate time series data. Pattern Recogn. Lett. 139, 139–147.

QLD. Great barrier reef real time water quality data. https://www.kaggle.com/ivivan/real-time-water-quality-data (accessed: 2020-07-20).

Rahman, S.A., Huang, Y., Claassen, J., Heintzman, N., Kleinberg, S., 2015. Combining fourier and lagged k-nearest neighbor imputation for biomedical time series data. J. Biomed. Informat. 58, 198–207.

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. 26, 43–49.

Schuster, M., Paliwal, K., 1997. Bidirectional recurrent neural networks. Trans. Sig. Proc. 45, 2673–2681.

Shi, T., Keneshloo, Y., Ramakrishnan, N., Reddy, C.K., 2021. Neural abstractive text summarization with sequence-to-sequence models. ACM Trans. Data Sci. 2, 1–37.

Suo, Q., Zhong, W., Xun, G., Sun, J., Chen, C., Zhang, A., 2020. Glima: Global and local time series imputation with multi-directional attention learning. In: 2020 IEEE International Conference on Big Data (Big Data). IEEE, pp. 798–807.

Tang, Z., Wang, D., Zhang, Z., 2016. Recurrent neural network training with dark knowledge transfer. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 5900–5904.

Tiwari, G., Sharma, A., Sahotra, A., Kapoor, R., 2020. English-hindi neural machine translation-lstm seq2seq and convs2s. In: 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE, pp. 871–875.

Van Zoest, V., Liu, X., Ngai, E., 2021. Data quality evaluation, outlier detection and missing data imputation methods for iot in smart cities. In: Machine Intelligence and Data Analytics for Sustainable Future Smart Cities. Springer, pp. 1–18.

Vilas, M.P., Thorburn, P.J., Fielke, S., Webster, T., Mooij, M., Biggs, J.S., Zhang, Y.F., Adham, A., Davis, A., Dungan, B., et al., 2020. 1622wq: A web-based application to increase farmer awareness of the impact of agriculture on water quality. Environ. Model. Softw. 132, 104816.

Vincent, L., Thome, N., 2019. Shape and time distortion loss for training deep time series forecasting models. In: Advances in Neural Information Processing Systems, pp. 4191–4203.

Yang, S., Dong, M., Wang, Y., Xu, C., 2020. Adversarial recurrent time series imputation. IEEE Trans. Neural Netw. Learn. Syst.

Yoon, J., Zame, W.R., van der Schaar, M., 2018. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. IEEE Trans. Biomed. Eng.

Zhang, Y., Fitch, P., Thorburn, P.J., 2020. Predicting the trend of dissolved oxygen based on the kpca-rnn model. Water 12, 585.

Zhang, Y., Fitch, P., Vilas, M.P., Thorburn, P.J., 2019a. Applying multi-layer artificial neural network and mutual information to the prediction of trends in dissolved oxygen. Front. Environ. Sci. 7, 46.

Zhang, Y., Thorburn, P., Xiang, W., Fitch, P., 2019b. SSIM -a deep learning approach for recovering missing time series sensor data. IEEE Internet Things J. 6, 6618–6628.