Water Resources Management
https://doi.org/10.1007/s11269-022-03408-6

Novel Methods for Imputing Missing Values in Water Level Monitoring Data

Thakolpat Khampuengson[1,2] · Wenjia Wang[1]

Received: 3 July 2022 / Accepted: 13 December 2022

Abstract
Hydrological data are collected automatically from remote water level monitoring stations and then transmitted to the national water management centre via telemetry system. However, the data received at the centre can be incomplete or anomalous due to some issues with the instruments such as power and sensor failures. Usually, the detected anomalies or missing data are just simply eliminated from the data, which could lead to inaccurate analysis or even false alarms. Therefore, it is very helpful to identify missing values and correct them as accurate as possible. In this paper, we introduced a new approach - Full Subsequence Matching (FSM), for imputing missing values in telemetry water level data. The FSM firstly identifies a sequence of missing values and replaces them with some constant values to create a dummy complete sequence. Then, searching for the most similar subsequence from the historical data. Finally, the identified subsequence will be adapted to fit the missing part based on their similarity. The imputation accuracy of the FSM was evaluated with telemetry water level data and compared to some well-established methods - Interpolation, k-NN, MissForest, and also a leading deep learning method - the Long Short-Term Memory (LSTM) technique. Experimental results show that the FSM technique can produce more precise imputations, particularly for those with strong periodic patterns.

Keywords Time series · Incomplete subsequence · Water level telemetry monitoring · Missing data imputation

1 Introduction

Using telemetry stations is a cost-effective approach to automatically collect the hydrological data for monitoring water levels in a country in real time. However, there are several factors that might disrupt the operation of stations such as environmental, technological issues and human activities, and consequently result in anomalous or missing data in the

Thakolpat Khampuengson
T.Khampuengson@uea.ac.uk

Wenjia Wang
Wenjia.Wang@uea.ac.uk

[1] School of Computing Sciences, University of East Anglia, Norwich, United Kingdom

[2] Hydro-Informatics Institute of Ministry of Higher Education, Bangkok, Thailand

Published online: 05 January 2023

collected water level data. Although, there are many methods, as discussed by Blázquez-García et al. (2020) and Yang et al. (2017), for discovering anomalies and missing data, they often remove anomalies from a series of data, or replace them with some constants, which are problematic because the missing data or inaccurate data can lead to erroneous analysis results. Thus, effective approaches for predicting missing values from accessible data are needed.

Replacing the missing gap by using the values from their most similar subsequence is the extensively utilized in many domains. Dynamic Time Warping (DTW) is an excellent technique of this kind and applied in a variety of problems. For example, the work by Tormene et al. (2009) uses DTW to discover the most similar incomplete time series from the stored reference of an arm movement sensor. Another example is the research by Caillault et al. (2020) which applied the derivative dynamic time warping (DDTW) developed by Keogh and Pazzani (2001) to search the subsequences before a missing gap, and then to repair the gap by using with the next most similar subsequence. The disadvantage of using dynamic time warping is time-consuming, which is addressed by extracting sequence features in sliding windows using a shape-feature extraction algorithm (Caillault et al. 2016), then calculating DDTW only if the correlation between the shape-features of this window and the subsequences before the missing gap is very high. The results demonstrate that their method produces superior outcomes when dealing with time series with a high correlation and strong seasonality.

Although DTW can find the most similar patterns that have similar dynamics but it may warp the shape by expanding or compressing, so the position of missing gaps may not be at the same position of the original pattern, as illustrated in Fig. 1.

The another way that uses for searching the most similar subsequence is to find two subsequences that have the lowest Euclidean distance as an indication of similarity. Since we need to calculate the distance of every pairwise in time series, the time needed to search for matches in a large time series dataset can be long and hence is considered as a disadvantage of this method. To address this issue Yeh et al. (2016) developed the techniques called Matrix Profile (MP), to speed up the process. An MP gives the distances between all subsequences and their nearest neighbours and thus can be used to efficiently extract some patterns characterised by a time series, such as motifs and discords. Very similar subsequences in a time series are called motifs, whereas very differing subsequences are called discords.

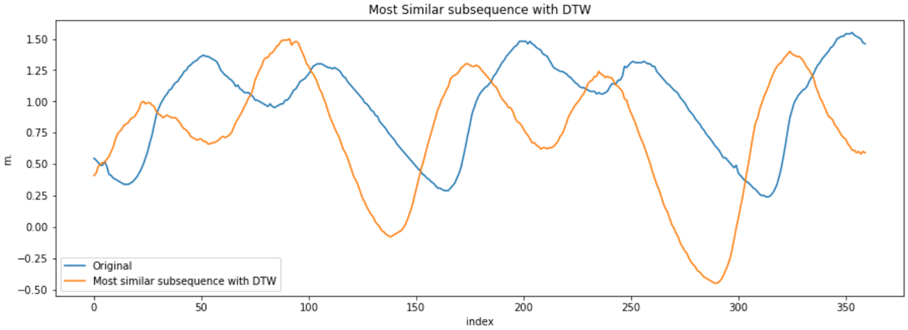


Fig. 1 An example of the most similar subsequence that has the same dynamic but has a different pattern to the original time series

This research aims to develop new techniques for imputing missing values or anomalies more accurately and efficiently. Specifically, our research focuses on imputing the missing values in time series of telemetry surface water level data. Our basic idea is inspired by the facts that water levels usually vary with some similar patterns over a year, so, we can utilize this phenomenon by reproducing the pattern from the most similar subsequence in the historical data. As a result, we built an effective approach for imputing missing data based on some simple operations for pattern searching and matching.

The remainder of the paper is organised as follows: Section 2 describes the associate works. The data characteristic in Section 3. Section 4 shows the detail of proposed method. Section 5 shows the details datasets, comparative methods, experiment setting and evaluation. Results and discussion are presented in Section 6. Finally, the paper is concluded in Section 7.

2 Related Work

Time series imputation methods can be classified into two categories in term of variables used: univariate and multivariate. The first approach uses a single variable to impute missing values. The second estimates missing data by examining the relationship between several variables.

Several fundamental methods in filling missing data in univariate including as mean, median, last observation carried forward (LOCF), and interpolation techniques, are frequently utilised (Peugh and Enders 2004; Pratama et al. 2016; Osman et al. 2018). When only one or a few consecutive missing data points are present, those methods provide acceptable results. However, when the missing gaps are large, the results are bad.

Some studies have been conducted in the recent decade with the goal of imputing missing values in meteorological and hydrological data. For example, the study by Yang et al. (2017) used the mean of nearby points imputation methods for imputing the missing data of the integrated data set from the water level and atmospheric data. Lai and Kuok (2019) suggested a method known as Bayesian Principal Component Analysis (BPCA) to impute missing values in rainfall data; the findings showed that BPCA outperformed KNN when dealing with large continuous missing gaps. While the research by Gao et al. (2018) and Pratama et al. (2016) provided a review of applied statistical and machine learning methodologies for imputed anomalies and missing data.

With the significant contribution of machine learning (ML) in many domains, it has also been applied to imputation tasks. Kim et al. (2015) compared ML, artificial neural network, and physical-based model for recovering streamflow data. The result revealed that ML were generally better than other at capturing high flows. Dwivedi (2022) used random forest to impute the continuous gaps and extreme of sub-hourly ground water. Li et al. (2022) improve the availability of IoT multivariate data and ability of anomaly detection by applied the imputation techniques to impute the detected anomalies data. Others research that using a ML model to estimate the missing values are averaging the prediction from two directions (forward and backward) as the final imputed values (Akouemo and Povinelli 2014; Bokde et al. 2018; Phan 2020). Two forecasting models for estimating the missing value based on the forecaster and the backcaster of time series were proposed by Moahmed et al. (2014).

In the last two decades, deep learning techniques have been widely applied to various problems in time series analysis. Various studies have exploited deep learning to impute the missing data. Zhang and Thorburn (2021) proposed a dual-head sequence-to-sequence imputation model (Dual-SSIM) for water quality data imputation. By averaging the prediction from gated recurrent units (GRU) with the information before and after the missing gaps. In this proposal, the model imputes missing data more accurately than 5 benchmarks. Kulanuwat et al. (2021) reported work that used three approaches for imputing missing data, including linear interpolation, spline interpolation, and bidirectional LSTM for data imputation on telemetry water level data. Spline interpolation performed better on non-cyclical data, while bidirectional long short-term memory (BiLSTM) beat other interpolation approaches on a particular tidal data pattern. But, one common disadvantage that all deep learning neural networks have is very time consuming, which makes them less practical in real time applications, such as water level analysis and flood forecasting.

In this paper, we propose a novel approach for computing the missing values in incomplete subsequences, called Full Subsequence Matching (FSM). Instead of splitting it into two subsequences, we replace missing data with some temporary constant values to produce a dummy complete subsequence. We then search for the most similar subsequence from the simulated complete subsequence. The missing data is then recreated by imitating the pattern of subsequences that are the most similar to each other.

3 Water Level Data

Telemetry stations have been installed in various locations to monitor the changes of river water levels in Thailand. In general, there are three key causes that can affect the changes of water level in a river with different behaviours.

- *Tidal*: The stations installed near the mouth of a river connecting to the sea will have a strong periodic pattern due to the effect from tidal.
- *Irrigation*: The water level from the station that was installed in the canal will be affected by the irrigation operation. Because this canal was constructed to convey water from the main canal, which may be controlled by floodgates, in order to conduct irrigation in the distant areas or to prevent flooding. As a result, the water level will vary depending on the event and usually has low fluctuation, none periodic, and few change of water level. However, when the gate is in operation, such as closing or opening, the water level data from the nearby station changes rapidly. The measured water level in the canal away from the floodgate, on the other hand, has a few changes.
- *Rain*: The station that has been installed in the natural river that far from the sea. When there is rainfall in the catchment area of a river, the water level can be affected in a variety of ways - in both upward and downward patterns.

In summary, these factors: raining, irrigating and tidal, have their patterns over the seasons and days. So the water levels that reflect their patterns can be explored and utilized to impute missing values.

4 Proposed Methods

To achieve our objective of building an effective and efficient framework for imputing missing values in water level data, we proposed a novel imputation approach, called Full Subsequence Matching (FSM), through finding the most similar subsequence. We compared our methods to the traditional idea of searching for the most similar subsequences of those patterns after splitting the subsequences, which is known as Partial Subsequence Matching (PSM). Each method is described in detail below:

4.1 Full Subsequence Matching (FSM)

In a time series, the data surrounding a missing gap can contain valuable information related to the gap and hence should be used for imputation of the missing points. A key question is how these pieces of useful information can be extracted and utilised in an efficient and effective manner. In this research, rather than separating the subsequence into two parts, before and after the missing gap, we replaced the missing gap with some temporary constant values to construct a dummy full sequence. Then, for parity in searching with historical data, we set the data in each sliding window at the same position as the same replaced constant values in the dummy full sequence. So, we can search for similar subsequences with the subsequences before and after a missing gap at the same time.

Our proposed FSM method consists of four main steps, which are explained as follows:

For a given time series $X = \{x_1, ..., x_N\}$, where N is the length, i.e. the number of data point in a time series.

Step one - Identifying a missing gap: Firstly, we identify the first missing point x_t and the last point x_T of a missing gap with T number of consecutive points, $[x_t, ..., x_{t+T}]$.

Step two - Extracting an extended subsequence: We then extract a subsequence I that contains the identified missing gap sandwiched with two subsequences of m and n consecutive data points at the left side and the right side of the gap respectively. This extended subsequence can be represent as:

$$I = \{x_{t-m}, ...x_{t-1}, [x_t, ..., x_{t+T}], x_{t+T+1}, ..., x_{t+T+n}\} \tag{1}$$

We then assign constant values c for every value of missing values in I as follows:

$$I = \{x_{t-m}, ..., x_{t-1}, [c, ..., c], x_{t+T+1}, ..., x_{t+T+n}\}. \tag{2}$$

Step three - Matching: This step searches and matches I with other subsequences in X. It is done by a sliding window technique. We set $W = \{w_1, w_2, ..., w_i\}$ to denote the subsequence in a sliding window where w_i is the set of consecutive values of X at position i with length z. We then compute the Euclidean distance of I with each subsequence in W. However, because the missing values in I have been replaced with constants, before computing the distance, we must replace all values in each subsequence of the sliding windows at the same position as the missing values in I with the same constant *c*. The most similar subsequence, denoted by *S*, is the one with the shortest distance, as shown in equation 3.

$$S = min\{d(I, W)\} \tag{3}$$

Step four - Imputation : We developed two different techniques to impute missing values: difference imputation and scaling imputation, as shown in Algorithm 1. They are explained further below.

1. *Difference Imputation* (FSM_D): If we know the difference between every two consecutive values in the any sequence, we can recreate the original series even if some values are missing. We calculate the difference between each pair of consecutive values in *S*, starting with the first pair of values at the same position of missing data in *I*. Then addition those value with the first values before the missing gap in *I* to calculate the first missing values. The difference between the following pairwise values in *S* is computed and added to the latest imputed values in *I*. We do so until all missing values have been imputed.
2. *Scaling Imputation* (FSM_S): The scale of the query subsequence and the scale of the most similar subsequence should be the same or almost the same. Hence, we can adjust the scale of the most similar subsequence to the scale of query subsequence to regenerate the values in missing gaps.

Algorithm 1 FSM Imputation

```
Input: S - the most similar subsequence.
       I - query subsequence.
       m - the number of points of the right subsequence.
       T - the length of missing gap.
procedure FSMD($S, I, m, T$)
    $D = \{\}$
    for $i$=0 to $T - 1$ do
        $D \leftarrow S[m+1+i]-S[m+i]$        difference between two consecutive values
    end for
    $imp = \{I[m]\}$
    for index $i$ in $D$ do
        $val = imp[-1] + D[i]$
        $imp \leftarrow val$
    end for
    return $imp$
end procedure
procedure FSMS($S, I, m, T$)
    $diff = S[m] - I[m]$        difference between two subsequences
    $max = Max(S[m : m + T + 1]) - diff$
    $min = Min(S[m : m + T + 1]) - diff$
    $val = I[m : m + T + 1]$
    $imp = MinMaxScaler(val, feature\_range = (min, max))$
    return $imp$
end procedure
```

4.2 Partial Subsequence Matching (PSM)

The basic idea behind the Partial Subsequence Matching method is that instead of using full subsequences for search and matching, only partial subsequences are used, which could speed up the process. The PSM is explained in detail as follows:

Step one - Identifying a missing gap: This step is the same as that of the FSM, i.e. finding the start and end indices of the missing gap in *X*.

Step two - Dividing: We then extract subsequence with *m* points from left (*L*) and *n* points from right (*R*) side of the missing gap in *X*. That is , we have that

$$L = \{x_{t-m}, ..., x_{t-1}\} \tag{4}$$

and

$$R = \{x_{t+T+1}, ..., x_{t+T+n}\} \tag{5}$$

Step three - Matching: We then search the most similar subsequences to *L* and *R*, denoted by S_L and S_R, respectively. It is done by computing the Euclidean distance of *L* and *R* with each subsequence in sliding windows *W*. The most similar subsequences can be represent by

$$S_L = min\{d(L, W)\} \tag{6}$$

and

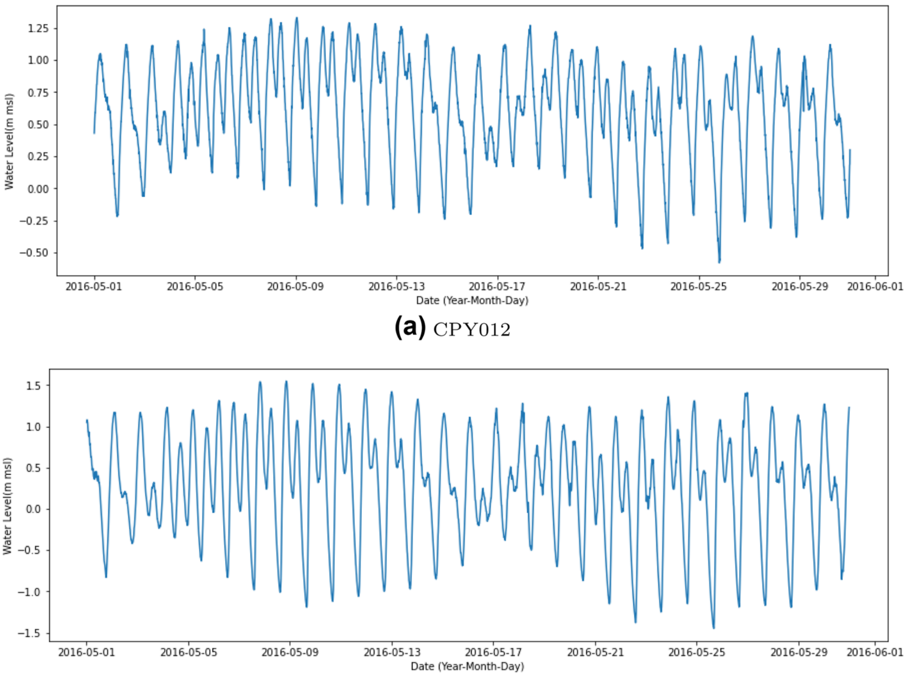
$$S_R = min\{d(R, W)\} \tag{7}$$

Step four - imputation: Four different techniques have been developed to impute the missing values as represented in Algorithm 2. We use S_L to generate the forward subsequence, and S_R to generate the backward subsequence, then combine those generated subsequences to impute the missing values. The missing values have been imputed by 4 different methods, as follows:

1. *Average Imputation* (PSM_A): We extracted the consecutive subsequence on the right side of S_L, and on the left side of S_R that was the same length as the missing gap. Then, to calculate the difference between each pair of consecutive values, we combined them with the average method before using them to impute missing values.
2. *Forward Imputation* (PSM_F): Instead of using an average difference from both side of the most similar subsequence, we then use only the calculated difference from subsequence on the right side of S_L to impute the missing values.
3. *Backward Imputation* (PSM_B): We used only the calculated difference from the subsequence on the left side S_R to impute the missing values.
4. *Weighted Imputation* (PSM_W): The basic idea is that the values closest to the missing gap have more effect than the values farthest away. We will assign higher weights to closer points and decrease the weight as the time interval grows. The missing values are then imputed by multiplying the difference between each pair of consecutive subsequences by their weighted score.

Algorithm 2 PSM Imputation

```
        p - last index of $S_L$
        k - first index of $S_R$
procedure PSMA($S_L, S_R, m, T, X$)
    $D_L = \{\}$
    $D_R = \{\}$
    for $i$=0 to $T - 1$ do
        $D_L \leftarrow X[p + i] - X[p - 1 + i]$
        $D_R \leftarrow X[k - i] - X[k - 1 - i]$
    end for
    $D_{avg} = avg(D_L, D_R)$
    $imp = \{X[p - 1]\}$
    for index $i$ in $D_{avg}$ do
        $val = imp[-1] + D_{avg}[i]$
        $imp \leftarrow val$
    end for
    return $imp$
end procedure
procedure PSMF($X, p, T$)
    $D_L = \{\}$
    for $i$=0 to $T - 1$ do
        $D_L \leftarrow X[p + i] - X[p - 1 + i]$        difference between two neighbours
    end for
    $imp = \{X[p - 1]\}$
    for index $i$ in $D_L$ do
        $val = imp[-1] + D_L[i]$
        $imp \leftarrow val$
    end for
    return $imp$
end procedure
procedure PSMB($X, k, T$)
    $D_R = \{\}$
    for $i$=1 to $T$ do
        $D_R \leftarrow X[k - 1] - X[k - 1 - i]$        difference between two neighbours
    end for
    $imp = \{X[k]\}$
    for index $i$ in $D_R$ do
        $val = imp[-1] + D_R[i]$
        $imp \leftarrow val$
    end for
    return $inverse(imp)$
end procedure
procedure PSMW($X, p, k, T$)
    $weightNumber = \{1, 2, 3, ..., T\}$
    $Weight_{fwd} = MinMaxScaler(weightNumber, feature_range = (0, 1))$
    $Weight_{bwd} = Inverse(Weight_{fwd})$
    $imp = \{L[m]\}$
    for $i$=1 to $T$ do
        $val = \frac{D_L[i]*Weight_{fwd}[i]+D_R[T-i]*Weight_{bwd}[i]}{Weight_{fwd}[i]+Weight_{bwd}[i]}$
        $imp \leftarrow val$
    end for
    return $imp$
end procedure
```



(a) CPY012

(b) CPY015

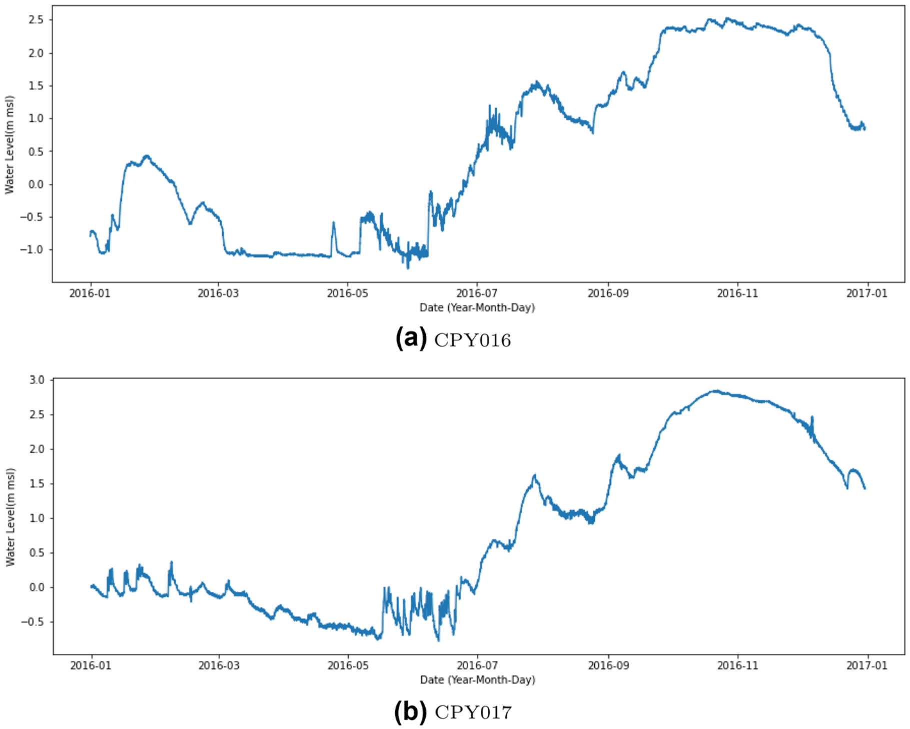
Fig. 2 Water level data with tidal influences

5 Experimental Set-up

5.1 Dataset

HII's telemetry sites in Thailand provided us with 10-minute time series water level data between 2012 and 2018. For the whole year, there are 52,560 data points. Missing values and abnormalities in raw data are commonly noticed as a result of faulty sensors or unexpected events. As a result, prior to any further analysis, a preliminary data pre-processing step is unavoidable.

We chose two years of data (2015 and 2016) from six representative stations (CPY012, CPY015, CPY016, CPY017, CHM003, and CHR004) to test the accuracy and generalisation of our proposed methods when dealing with different data behaviours. CPY012 and CPY015 stations represent the data with tidal effects that have strong periodic patterns, as depicted in Fig. 2. While the data from CPY016 and CPY017 that have fluctuation characteristics with few upward and downward as a result of irrigation operating in the canal that has telemetry stations installed, as shown in Fig. 3. Additionally, the data from CHM003 and CHR004 show fluctuations and many upward and downward patterns as a result of the rain effect, as shown in Fig. 4.



(a) CPY016

(b) CPY017

Fig. 3 Water level data with irrigation influences

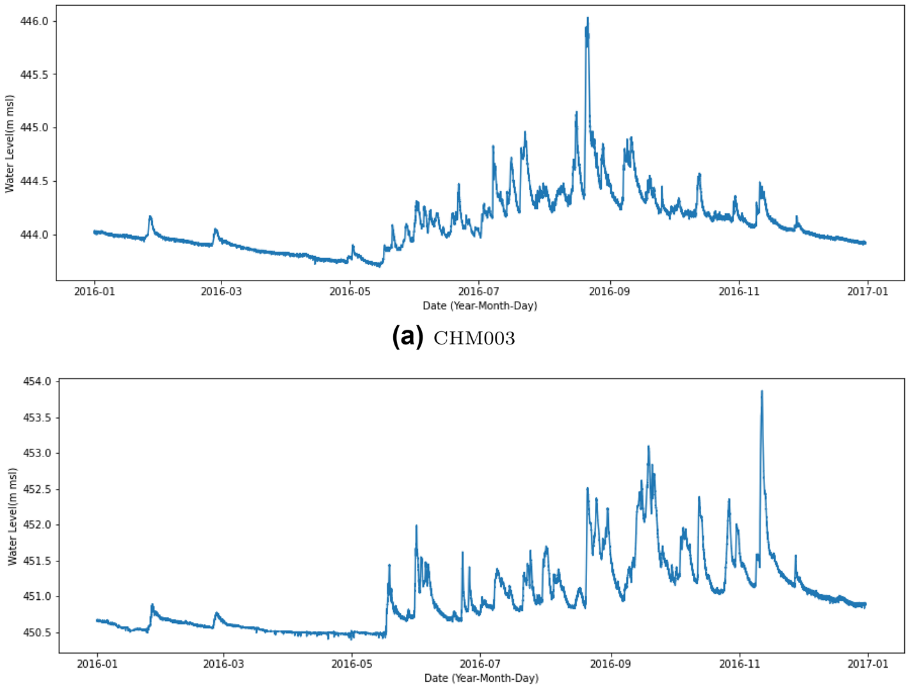
5.2 Missing Data Generation

We were unable to examine the accuracy of imputation algorithms on genuine missing data because the true values were not available. But we can simulate some missing data with some methods on complete data in order to evaluate the performance of imputation approaches. To produce datasets with missing data, we delete consecutive values from the dataset under the assumption that it happens at random.

To simulate various missing data situations, we generated missing gaps of sizes 6, 12, 18, 36, 72, and 144 (1 hour, 2 hours, 3 hours, 6 hours, 12 hours, and 1 day), and the length of a consecutive subsequence before and after the missing gap equal to the size of the missing gap. For instance, if the missing gap is six in length, the lengths of the subsequences before and after the missing gap are also six.

5.3 Comparative Imputation Methods

We chose some well-known representative imputation methods for comparing our methods. These are interpolation of linear and polynomial, k-Nearest Neighbours (k-NN), MissForest (MF), and a deep learning method - Long Short Term Memory (LSTM).



(a) CHM003

(b) CHR004

Fig. 4 Water level data with rain influences

- *Interpolation:* When a time series data has a gap, the available data on each side of the gap or at a few particular locations within the gap, can be used with an interpolation method to estimate values in the gap. Two methods have been chosen to impute the gap in water level data:

  (a) *Linear Interpolation (Li-inter):* It just fits a straight line between two adjacent points of a missing gap for estimating the missing values. This method is simple and fast and hence is often used as a baseline in imputation.

  (b) *Polynomial Interpolation (Poly-Deg2):* It is an enhanced interpolation approach that attempts to find the optimal polynomial function to match the data. It can be used to estimate data in the form of a curve, and thus should be suitable for water level data.

- *k-Nearest Neighbour (k-NN):* It works on the assumption that neighbouring data points belong to the same class. In other words, a new data point is more likely to have the same class label as its k-nearest neighbours than distant data points (Peterson 2009). k-NN identifies the neighboring points through a measure of distance and the missing values can be estimated using completed values of neighboring observations.
- *MissForest (MF):* It is another machine learning-based data imputation technique that based on the Random Forest (RF) algorithm which has been created by Stekhoven and

Bühlmann (2012). It can be divided into 3 main steps. Firstly, replace the missing values with the mean (for continuous variables) or the most frequent class (for categorical variables). Secondly, the observed observations are served as the training set and the missing observations are served as the prediction set. The training sets and the prediction sets are fed into a RF model. Then, the RF model's predictions are put in place of the prediction set, creating a transformed dataset. Finally, one imputation loop is complete when all missing variables are imputed. Imputations are repeated.

- *Long-Short Term Memory (LSTM):* LSTM is one of the architectures of artificial recurrent neural network (RNN) that has been utilised for a number of purposes, including, petroleum industry (Sagheer and Kotb 2019), handwriting recognition (Nogra et al. 2019), anomaly detection (Maleki et al. 2021), and data imputation (Yuan et al. 2018). A typical LSTM is made up of four units. (1) The cell that remembers values across arbitrary time intervals, (2) Input gate, (3) Output gate, and (4) a forget gate. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. There may be lags of unknown duration between important events in a time series. As a result, LSTM is well-suited to categorising, analysing, and forecasting time series data, and is considered as the state of the art method. We utilised two LSTM models ($LSTM_F$ and $LSTM_B$) with the same architecture for predicting and backcasting the missing gaps. In the event of backcasting, we invert the position of the subsequence after the missing data and feed it into the $\mathrm{LSTM_B}$ model. Furthermore, we created two new imputations based on the results of both $\mathrm{LSTM_F}$ and $\mathrm{LSTM_B}$. The first is $LSTM_A$, which takes the average of the outputs of both models and uses it as the final output. The second is $LSTM_W$, which we weight the values of output from $\mathrm{LSTM_F}$ and $\mathrm{LSTM_B}$ using the same notion of weighting imputation as in FSM_W by assigning the greatest weighting score to the data that is closest to the current values.

5.4 Experimental Setting

The datasets are divided depending on each method, and we run each method 500 times and average the results. The dataset is divided into training/searching and testing/removing. The training data is used to fit neural network model and search the most similar subsequence, while the testing data is used to generate the missing subsequences and assess model performance. For training purposes, we used data from 2015, whereas for testing purposes, we used data from 2016.

For interpolation technique we used interpolate class in panda.DataFrame[1] python library, which is a method for filling missing value using an interpolation method. We specified a linear approach for linear interpolation and a polynomial method with an order of 2 for polynomial interpolation.

We used the grid search technique to find the best *k* number of nearest neighbours, ranging from 2 to the number of members in the query subsequence for k-NN models. Since MF models require multivariate data, we chose the simplest way to convert our data from univariate to multivariate by dividing the query sequence into three subsequences to use as input for MF models.

For partial subsequence matching, we used the matrix profile python library called STUMPY (Law 2019) which is a powerful and scalable library. In order to properly train

[1] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.interpolate.html

LSTM models, we used one LSTM layer and one hidden layer that uses only dense layer, 30 training epochs, prevent time-consuming and over-fitting with early stopping with the patience values of 5 and mini-batch size of 128. In practical, we tried with different number of neurons per layer (64, 128, and 256) and found that 128 neuron per layer give the best result. The input of LSTM is the subsequence of before and after missing values for prediction the missing values.

All the experiments were coded with Python Programming Language (V3.6) and TensorFlow 2.8, and run on a personal computer with an Intel Core i5-7500 CPU @ 3.4 GHz, 32 GB RAM, 64-Bit Operating System.

5.5 Evaluation Metrics

The error or accuracy of an imputation method is measured with three metrics: root mean square error (RMSE), mean absolute error (MAE), and similarity (Sim) are defined as follows:

- *RMSE:* The average squared difference between the imputed value $\hat{y}$ and the respective genuine value y is referred to as the Root Mean Square Error (RMSE). This metric is very useful for determining overall correctness. The technique with the lowest RMSE would be the most accurate.

$$RMSE(\hat{y}, y) = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(\hat{y}_i - y_i)^2} \tag{8}$$

where T is the number of missing values.

- *MAE:* The Mean Absolute Error is compute as average of the absolute difference between imputed values $\hat{y}$ and actual values y, which calculated by:

$$MAE(\hat{y}, y) = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{9}$$

The method that is more effective will have a lower MAE.

- *Similarity:* $\mathrm{Sim}(\hat{y}, y)$ defines the similar percentage between the imputed value ($\hat{y}$) and the actual data (y). It is calculated by:

$$Sim(\hat{y}, y) = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{1 + \frac{|\hat{y}_i - y_i|}{max(y) - min(y)}} \tag{10}$$

A higher similarity indicates a more accurate imputation of missing values.

6 Results and Discussion

6.1 Results

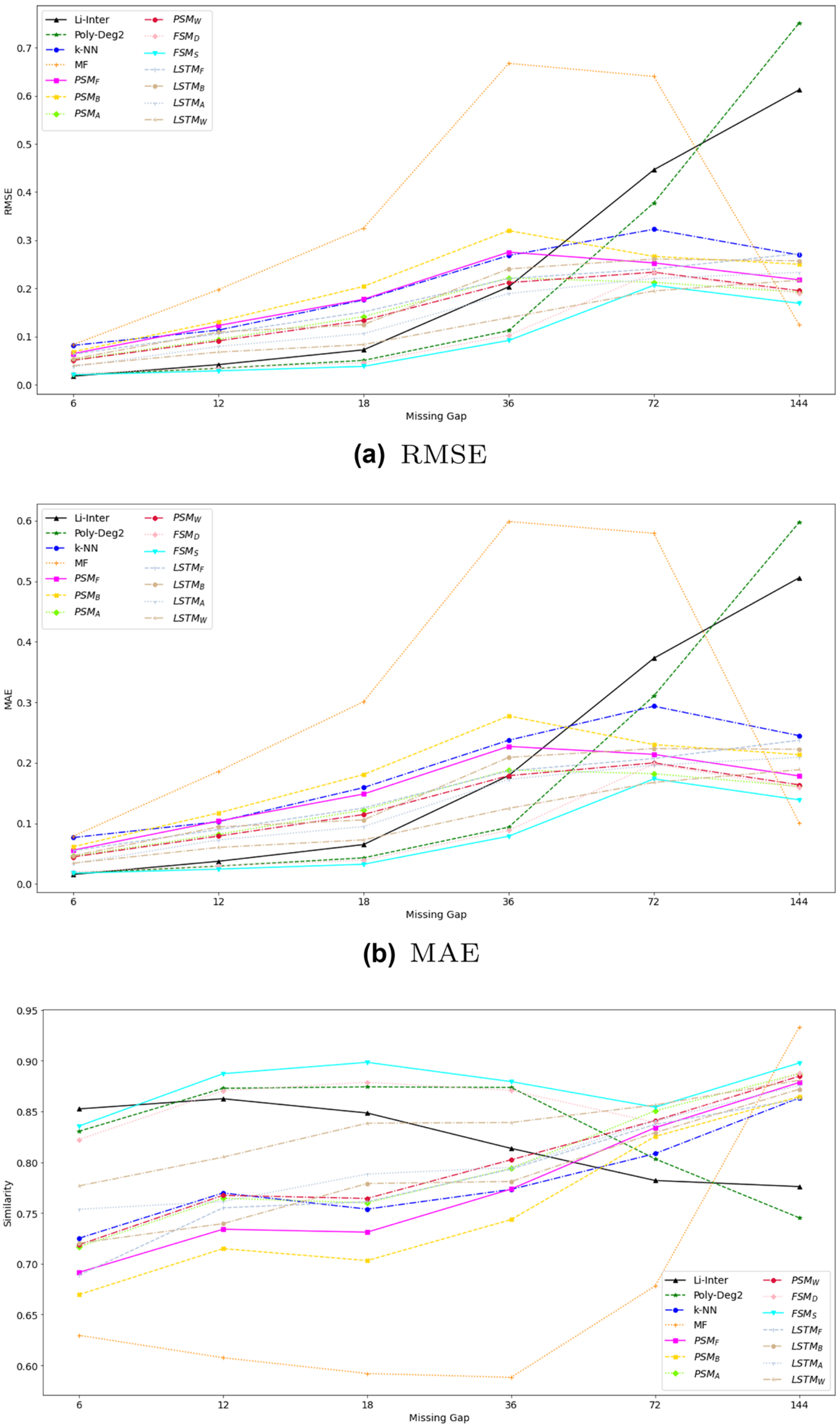
Incomplete subsequence matching methods as described in Section 4 guarantee that our suggested model will be capable of producing imputation results with varying lengths.

Table 1 The average imputation performance indexes of 14 methods on telemetry water level data with tidal influence (the best score for each row in each gap size is shown in bold)

Method	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim
Li-Inter	6	**0.0180**	**0.0158**	**0.8525**	12	0.0419	0.0376	0.8625	18	0.0726	0.0652	0.8485
Poly-Deg2		0.0202	0.0177	0.8305		0.0343	0.0297	0.8728		0.0507	0.0433	0.8742
k-NN		0.0818	0.0768	0.7251		0.1136	0.1030	0.7697		0.1763	0.1593	0.7538
MF		0.0835	0.0790	0.6295		0.1976	0.1862	0.6074		0.3253	0.3011	0.5919
PSM_F		0.0647	0.0560	0.6915		0.1231	0.1040	0.7340		0.1776	0.1487	0.7311
PSM_B		0.0688	0.0618	0.6696		0.1315	0.1174	0.7149		0.2044	0.1809	0.7031
PSM_A		0.0522	0.0460	0.7167		0.0944	0.0822	0.7645		0.1417	0.1223	0.7600
PSM_W		0.0509	0.0448	0.7187		0.0912	0.0792	0.7674		0.1339	0.1149	0.7643
FSM_D		0.0231	0.0198	0.8223		0.0349	0.0300	0.8704		0.0460	0.0395	0.8788
FSM_S		0.0208	0.0178	0.8356		**0.0290**	**0.0247**	**0.8872**		**0.0383**	**0.0327**	**0.8984**
$LSTM_F$		0.0634	0.0551	0.6888		0.1062	0.0901	0.7551		0.1513	0.1257	0.7610
$LSTM_B$		0.0540	0.0470	0.7203		0.1096	0.0944	0.7395		0.1251	0.1060	0.7790
$LSTM_A$		0.0376	0.0340	0.7537		0.0797	0.0729	0.7613		0.1064	0.0950	0.7883
$LSTM_W$		0.0398	0.0345	0.7767		0.0679	0.0605	0.8050		0.0836	0.0730	0.8384
Li-Inter	36	0.2034	0.1792	0.8136	72	0.4464	0.3729	0.7820	144	0.6121	0.5056	0.7760
Poly-Deg2		0.1125	0.0939	0.8736		0.3777	0.3107	0.8033		0.7508	0.5970	0.7453
k-NN		0.2685	0.2375	0.7732		0.3229	0.2937	0.8085		0.2695	0.2451	0.8633
MF		0.6674	0.5987	0.5881		0.6402	0.5792	0.6781		**0.1253**	**0.1014**	**0.9331**
PSM_F		0.2754	0.2273	0.7736		0.2531	0.2142	0.8340		0.2180	0.1786	0.8785
PSM_B		0.3202	0.2776	0.7435		0.2666	0.2303	0.8254		0.2503	0.2137	0.8646
PSM_A		0.2220	0.1885	0.7940		0.2128	0.1822	0.8506		0.1923	0.1616	0.8873
PSM_W		0.2121	0.1790	0.8024		0.2338	0.2001	0.8410		0.1954	0.1639	0.8847
FSM_D		0.1029	0.0887	0.8707		0.2329	0.1982	0.8392		0.1904	0.1593	0.8877
FSM_S		**0.0921**	**0.0790**	**0.8794**		0.2066	0.1737	0.8542		0.1692	0.1391	0.8977
$LSTM_F$		0.2205	0.1869	0.7933		0.2405	0.2073	0.8372		0.2724	0.2375	0.8622

Table 1 (continued)

Method	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim
$LSTM_B$		0.2406	0.2092	0.7810		0.2612	0.2237	0.8291		0.2574	0.2227	0.8719
$LSTM_A$		0.1895	0.1729	0.7948		0.2208	0.1962	0.8408		0.2334	0.2094	0.8758
$LSTM_W$		0.1394	0.1250	0.8391		**0.1949**	**0.1679**	**0.8564**		0.2172	0.1889	0.8807



(a) RMSE

(b) MAE

(c) Similarity

◀ **Fig. 5** The performance for imputing telemetry water level data with tidal influence for various missing gap size

Aside from that, the number of available data points around the missing values can also be adjusted. As a result, the *FMS* is built to deal with random size of data gaps in time series. We random remove the subsequence 500 times from telemetry water level data and the results described as follow.

We first consider the tidal-influenced data whose recurrent upward and downward trends are noticeably and frequent changing with a similar magnitude. The average imputation performance of each methods are depicted in Table 1. As expected, when the size of the missing gaps is small, e.g., 6, linear interpolation techniques (Li-Inter) achieve the best performance for RMSE, MAE, and Sim with 0.0180, 0.0158, and 0.8525, respectively. However, their performance degrades steadily when dealing with gaps bigger than 6. Similar to polynomial interpolation (Poly-Deg2), which performed well when imputed missing data with a small gap but poorly when the gap increased. The MissForest (MF) technique performed the poorest results with gap size lower than 144, particularly on gap size 36, with 0.6674 (RMSE), 0.5987 (MAE), and 0.5991 (Sim), but it performed best when it filled in the missing gaps with a size of 144, with 0.1253 (RMSE), 0.1014 (MAE), and 0.9331 (Sim). Our proposed solution, FSM_S, outperforms all others when imputed missing gaps of sizes 12, 18, and 36. LSTM_W performed best on gap size 72, with RMSE, MAE, and Sim of 0.1949, 0.1678, and 0.8564, respectively. When the missing gaps were imputed at size 144, MF beat other models with the lowest RMSE of 0.0383, the lowest MAE of 0.0322, and the highest Sim of 0.9331.

We also plotted the average imputation performance for 14 methods using telemetry water level data with tidal features, as illustrated in Fig. 5. As we can see, the interpolation method, Li-Inter and Poly-Deg2, appeared to decrease in performance as the number of missing gaps rose. When filling in data gaps with sizes of 12, 18, and 36, our suggested method, FSM_S, is clearly better than all others. After increasing the gap size to 144, performance of k-NN often improves. While performance of MF was the poorest while trying to impute missing data with a gap size of 72 or less, it improved to the best when the gap size was increased to 144. Although the set of PSM techniques performs poorly when the missing gap size is small, their performance improves when the missing gap size is equal to or higher than 72. It is interesting to note that as the amount of input data goes up, the performance of LSTM approaches gets better as the number of missing gaps goes up.

Figure 6 shows the comparison of the critical difference between the different imputation models. The number associated with each algorithm is the average rank of the

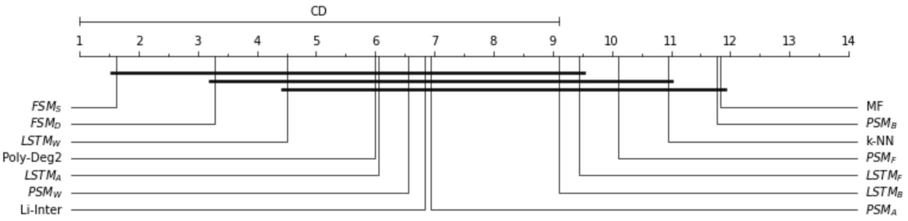


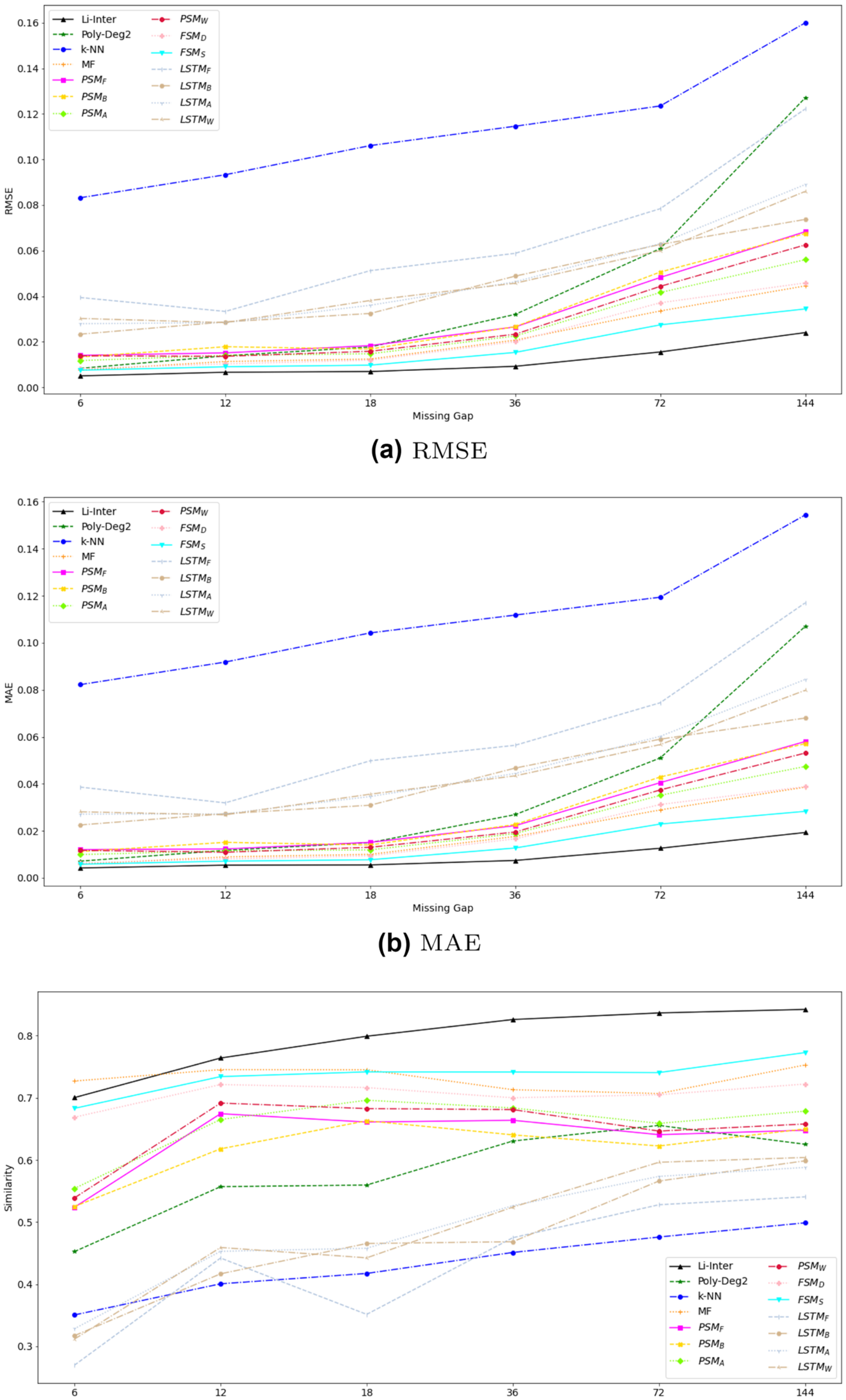
Fig. 6 A critical difference diagram for 14 different imputation techniques on tidal influence datasets of telemetry water level data

Table 2 The average imputation performance indexes of 14 methods on telemetry water level data with irrigation influence (the best score for each row in each gap size is shown in bold)

Method	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim
Li-Inter	6	**0.0050**	**0.0042**	**0.7004**	12	**0.0066**	**0.0054**	**0.7641**	18	**0.0069**	**0.0055**	**0.7992**
Poly-Deg2		0.0083	0.0071	0.4525		0.0138	0.0117	0.5570		0.0177	0.0150	0.5596
k-NN		0.0832	0.0822	0.3505		0.0933	0.0918	0.4006		0.1061	0.1042	0.4173
MF		0.0075	0.0058	0.7271		0.0113	0.0089	0.7454		0.0124	0.0100	0.7452
$\mathrm{PSM_F}$		0.0140	0.0120	0.5239		0.0151	0.0123	0.6745		0.0183	0.0151	0.6610
$\mathrm{PSM_B}$		0.0133	0.0112	0.5251		0.0178	0.0151	0.6179		0.0168	0.0140	0.6631
$\mathrm{PSM_A}$		0.0117	0.0099	0.5542		0.0140	0.0115	0.6651		0.0147	0.0119	0.6962
$\mathrm{PSM_W}$		0.0138	0.0117	0.5389		0.0136	0.0109	0.6916		0.0158	0.0130	0.6827
$\mathrm{FSM_D}$		0.0082	0.0063	0.6689		0.0103	0.0081	0.7214		0.0119	0.0094	0.7165
$\mathrm{FSM_S}$		0.0075	0.0058	0.6831		0.0090	0.0071	0.7342		0.0097	0.0077	0.7418
$\mathrm{LSTM_F}$		0.0394	0.0386	0.2696		0.0333	0.0319	0.4421		0.0512	0.0498	0.3514
$\mathrm{LSTM_B}$		0.0233	0.0225	0.3171		0.0287	0.0273	0.4167		0.0324	0.0309	0.4655
$\mathrm{LSTM_A}$		0.0279	0.0271	0.3283		0.0284	0.0272	0.4529		0.0360	0.0346	0.4579
$\mathrm{LSTM_W}$		0.0302	0.0281	0.3116		0.0284	0.0268	0.4591		0.0381	0.0356	0.4424
Li-Inter	36	**0.0092**	**0.0074**	**0.8261**	72	**0.0155**	**0.0126**	**0.8367**	144	**0.0240**	**0.0193**	**0.8424**
Poly-Deg2		0.0320	0.0269	0.6305		0.0609	0.0510	0.6556		0.1271	0.1070	0.6252
k-NN		0.1146	0.1118	0.4511		0.1235	0.1194	0.4759		0.1600	0.1544	0.4988
MF		0.0207	0.0176	0.7131		0.0335	0.0289	0.7069		0.0445	0.0386	0.7529
$\mathrm{PSM_F}$		0.0265	0.0222	0.6639		0.0482	0.0405	0.6407		0.0683	0.0580	0.6479
$\mathrm{PSM_B}$		0.0266	0.0227	0.6404		0.0506	0.0429	0.6225		0.0675	0.0571	0.6500
$\mathrm{PSM_A}$		0.0225	0.0188	0.6837		0.0416	0.0351	0.6589		0.0560	0.0474	0.6786
$\mathrm{PSM_W}$		0.0233	0.0195	0.6811		0.0443	0.0374	0.6465		0.0625	0.0531	0.6579
$\mathrm{FSM_D}$		0.0200	0.0166	0.7001		0.0371	0.0312	0.7050		0.0458	0.0388	0.7222
$\mathrm{FSM_S}$		0.0153	0.0127	0.7416		0.0274	0.0229	0.7407		0.0344	0.0283	0.7729
$\mathrm{LSTM_F}$		0.0588	0.0564	0.4748		0.0784	0.0745	0.5279		0.1222	0.1170	0.5406

Table 2 (continued)

Method	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim
LSTM_B		0.0488	0.0467	0.4684		0.0627	0.0590	0.5661		0.0737	0.0680	0.5986
LSTM_A		0.0463	0.0444	0.5264		0.0631	0.0602	0.5735		0.0890	0.0844	0.5880
LSTM_W		0.0457	0.0434	0.5241		0.0600	0.0567	0.5963		0.0860	0.0798	0.6040



(a) RMSE

(b) MAE

(c) Similarity

◀ **Fig. 7** The performance for imputing telemetry water level data with irrigation influence for various missing gap size

imputation models on each type of datasets and solid bar group classifiers with no significant difference. For the data type with tidal effect, FSM$_S$ achieved the top rank follow with FSM$_D$, LSTM$_W$, and Poly-Deg2, respectively. MF not only provided the lowest ranking but also significant difference from FSM-based technique.

Table 2 shows the imputing findings for the irrigation-affected data. Li-Inter is not only the best imputation model for missing gaps of size 6, but also performs well for larger missing gaps. k-NN, on the other hand, performed the poorest with every missing gap size and has a score difference from Li-Inter of roughly 0.08 in every performance metric.

As seen in Fig. 7, the performance of all approaches fell progressively as the size of the missing gap rose, with the exception of the similarity score of LSTM models, which tended to improve performance as the gap size increases.

The CD diagram in Fig. 8 revealed that Li-Inter took first place, followed by our suggested technique (FSM$_S$), and MF, respectively, while the group of LSTM models performed the worst. A collection of PSM and FSM models works well and offers a considerable improvement over LSTM-based imputation approaches.

Regarding the impacts of rain, Li-Inter outperformed across all gap sizes and evaluation metrics, with the exception of gap 6, where performance was slightly lower than MF, around 0.0259, for Sim score, as illustrated in Table 3. Moreover, the line charts in Figure 9 present the performance for imputing telemetry water level data with rain influence for various missing gap sizes. As we can see, the set of LSTM models scores the lowest on all evaluation metrics. Li-Inter and Poly-Deg2 have a tendency to perform worse as the number of missing values grows, while the set of PSM and FSM strategies maintain a consistent level of performance (Fig. 10).

When used to impute the missing data on the water level with rain-effected, Li-Inter, FSM$_S$, and MF maintained their top rankings. However, k-NN dropped to the bottom of the list. LSTM-based techniques still provided the low ranking and significant difference from Li-Inter and FSM$_S$.

6.2 Discussion

Li-inter and Poly-Deg2 produced the best performance on data with non-cyclical and periodic patterns, like data with rain and irrigation effects. Moreover, with the small missing gap size Li-inter outperformed the others with all data behaviours, which is expected.

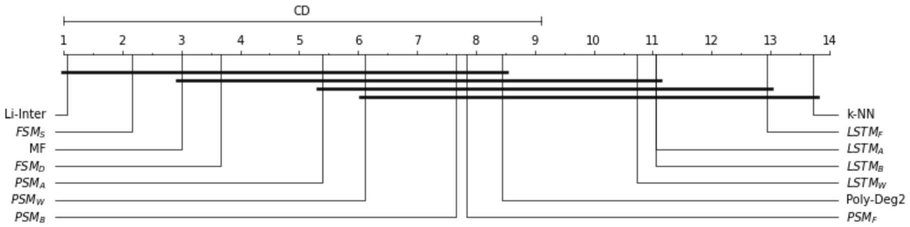


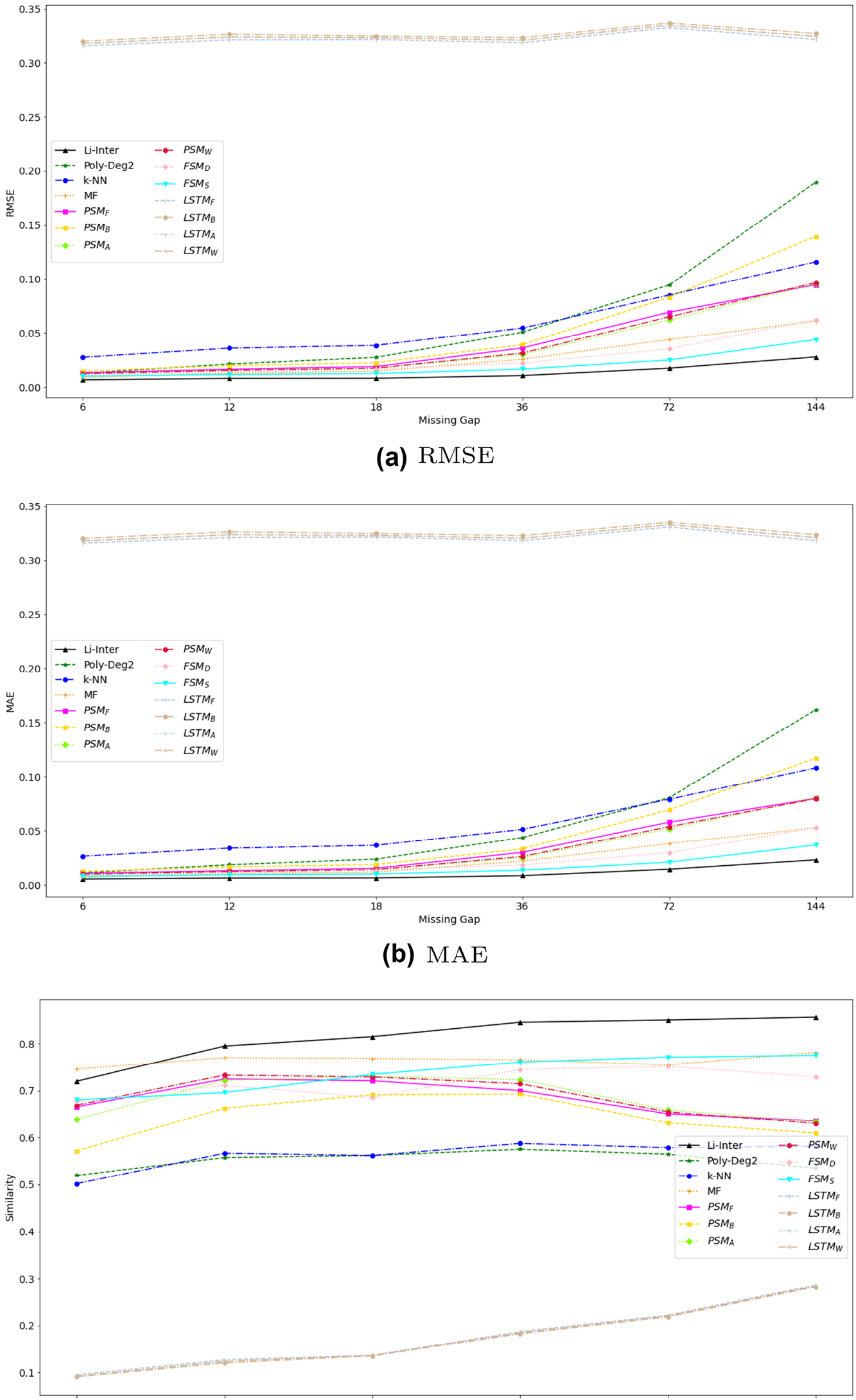
Fig. 8 A critical difference diagram for 14 different imputation techniques on irrigation influence datasets of telemetry water level data

Table 3 The average imputation performance indexes of 14 methods on telemetry water level data with rain influence (the best score for each row in each gap size is shown in bold)

Method	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim
Li-Inter	6	**0.0066**	**0.0053**	0.7200	12	**0.0079**	**0.0062**	**0.7949**	18	**0.0080**	**0.0063**	**0.8148**
Poly-Deg2		0.0126	0.0110	0.5195		0.0211	0.0184	0.5577		0.0273	0.0236	0.5621
k-NN		0.0274	0.0263	0.5021		0.0358	0.0338	0.5668		0.0384	0.0364	0.5620
MF		0.0089	0.0069	**0.7459**		0.0128	0.0102	0.7701		0.0145	0.0117	0.7685
PSM_F		0.0131	0.0106	0.6658		0.0163	0.0129	0.7248		0.0188	0.0151	0.7210
PSM_B		0.0147	0.0124	0.5719		0.0196	0.0164	0.6628		0.0223	0.0186	0.6912
PSM_A		0.0120	0.0097	0.6395		0.0151	0.0122	0.7218		0.0173	0.0140	0.7305
PSM_W		0.0122	0.0098	0.6694		0.0153	0.0121	0.7330		0.0173	0.0139	0.7291
FSM_D		0.0110	0.0086	0.6773		0.0137	0.0109	0.7114		0.0165	0.0137	0.6853
FSM_S		0.0100	0.0078	0.6804		0.0115	0.0092	0.6959		0.0124	0.0099	0.7346
$LSTM_F$		0.3161	0.3159	0.0952		0.3216	0.3211	0.1272		0.3219	0.3215	0.1364
$LSTM_B$		0.3204	0.3202	0.0912		0.3268	0.3263	0.1204		0.3252	0.3248	0.1354
$LSTM_A$		0.3182	0.3180	0.0912		0.3241	0.3236	0.1239		0.3235	0.3230	0.1357
$LSTM_W$		0.3184	0.3180	0.0922		0.3243	0.3236	0.1238		0.3237	0.3231	0.1361
Li-Inter	36	**0.0105**	**0.0084**	**0.8453**	72	**0.0173**	**0.0143**	**0.8500**	144	**0.0277**	**0.0229**	**0.8561**
Poly-Deg2		0.0506	0.0436	0.5753		0.0944	0.0804	0.5647		0.1895	0.1618	0.5348
k-NN		0.0545	0.0512	0.5877		0.0850	0.0790	0.5786		0.1158	0.1080	0.5818
MF		0.0256	0.0218	0.7656		0.0438	0.0380	0.7548		0.0609	0.0527	0.7813
PSM_F		0.0360	0.0299	0.7001		0.0691	0.0578	0.6509		0.0945	0.0798	0.6358
PSM_B		0.0391	0.0331	0.6928		0.0830	0.0693	0.6315		0.1394	0.1169	0.6097
PSM_A		0.0300	0.0249	0.7234		0.0620	0.0517	0.6591		0.0955	0.0803	0.6342
PSM_W		0.0313	0.0261	0.7148		0.0649	0.0537	0.6549		0.0964	0.0795	0.6304
FSM_D		0.0221	0.0182	0.7452		0.0352	0.0295	0.7518		0.0625	0.0528	0.7296
FSM_S		0.0164	0.0134	0.7607		0.0249	0.0208	0.7711		0.0437	0.0367	0.7750
$LSTM_F$		0.3190	0.3181	0.1874		0.3328	0.3307	0.2219		0.3219	0.3182	0.2862

Table 3 (continued)

Method	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim	Gap	RMSE	MAE	Sim
LSTM_B		0.3237	0.3228	0.1827		0.3371	0.3351	0.2185		0.3275	0.3237	0.2824
LSTM_A		0.3212	0.3203	0.1844		0.3348	0.3328	0.2204		0.3246	0.3208	0.2842
LSTM_W		0.3214	0.3203	0.1849		0.3351	0.3329	0.2202		0.3249	0.3210	0.2840



(a) RMSE

(b) MAE

(c) Similarity

◀ **Fig. 9** The performance for imputing telemetry water level data with rain influence for various missing gap size

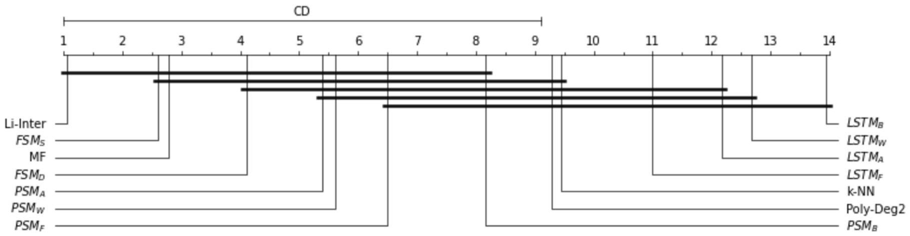


Fig. 10 A critical difference diagram for 14 different imputation techniques on rain influence datasets of telemetry water level data

When imputing the missing values on water level data with tidal influence, our approaches $\mathrm{FSM_S}$ outperformed the others. It is mostly due to the fact that some cyclical patterns of tidal influence are repeated in water level data over time, and our approaches are capable of finding and matching the most similar pattern in the past to impute the missing data more accurately. However, when we took a close look into the very short missing gaps, which still seem more like linear, then it is not surprising to see that Li-inter performed better.

The utilized LSTM model was trained on the input subsequence and its reversed copy that preserved both past and future information of a specific time frame. With this advantage, LSTM models are able to understand the context better and thus in principle they should be suitable for imputing the missing data in telemetry water level. However, they did not produce the best performance. According to our experiments on the data with different behaviours, LSTM models incorrectly estimated strongly fluctuated data, for example the data with some raining effects. On the other hand, they are capable of imputing the missing values on the data with tidal and irrigation effects with periodic and without frequent trends.

Since MissForest(MF) does not use data for training, it performed poorly when dealing with short subsequences but produced excellent outcomes for missing data with large gap sizes. In other words, MF performed better when appropriate data is available. However, since MF models need multivariate data, which is not always the case in this application and transformation from univariate data to multivariate data can introduce noise or misrepresentations. Splitting the sequence into many subsequences is the simplest technique to generate multivariate data. This raises the challenge of determining the optimal number of subsequence splits.

7 Conclusion

This paper introduced two sequence matching methods: Full and Partial methods, for searching the most similar subsequence and filling the missing values in telemetry water level data. They were tested with real-world water level data collected from 6 water level monitoring stations and their results were compared with a range of other existing methods including some commonly used methods and the latest so called state of the art deep learning methods - Long Short-Term Memory (LSTM). The results showed that our new methods, particularly the Full Sequence Matching with scaling imputation technique (FSM_S), are better then all of them.

The FSM approach uses the Euclidean distance technique to search for the most similar subsequence of the query subsequences, then calculate missing data values based on the pattern of those subsequence. However, rather than dividing it into two subsequences we replaced missing data with constant values and searched as a single subsequence. The proposed methods were evaluated using missing data simulated on six time series water level data with three distinct data behaviours. The results indicate that FSM with scaling imputation, FSM_S, outperforms other imputation methods when dealing with large missing gap sizes.

The FSM_S performs well on data that has strong periodic and cyclical pattern such as data of water level with tidal effects. While the linear interpolation approach works well with data that fluctuates and has a number of up and down trends such as water level data with rain and irrigation effects. LSTM and MF models show increased performance when the missing gap size is increased. However, for large datasets, LSTM is computationally costly and time-consuming. Although we may train the model using long periods of historical data with high performance computing (HPC) to shorten training time, we have no way of knowing when we need to retrain the model, which is a significant downside of this approach. While MF needs to transform univariate data to multivariate data which difficult to find the appropriate techniques of transformation.

For further work, it should be useful to explore reconstructing the missing data from a different station for more robust data imputation. This is due to the fact that some installed stations that are close to each other and in the same canal/river are likely to have a similar pattern. Another work can be to extend the approaches for dealing with multidimensional information.

Acknowledgements The authors would like to thank the Hydro-Informatics Institute of Ministry of Higher Education, Science, Research and Innovation, Thailand, for providing the scholarship and the data for Thakolpat Khampuengson to do his PhD at the University of East Anglia.

Declarations

Ethics Approval and Consent to Participate This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for Publication All authors have consented to publish this manuscript.

Conflict of Interest No potential conflict of interest was reported by the authors.

References

Akouemo HN, Povinelli RJ (2014) Time series outlier detection and imputation. In: 2014 IEEE PES General Meeting Conference & Exposition, IEEE, pp 1–5

Blázquez-García A, Conde A, Mori U, Lozano JA (2020) A review on outlier/anomaly detection in time series data. Preprint at http://arxiv.org/abs/2002.04236

Bokde N, Beck MW, Álvarez FM, Kulat K (2018) A novel imputation methodology for time series based on pattern sequence forecasting. Pattern Recogn Lett 116:88–96

Caillault EP, Bigand A et al (2016) Comparative study on supervised learning methods for identifying phytoplankton species. In: 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), IEEE, pp 283–288

Caillault ÉP, Lefebvre A, Bigand A et al (2020) Dynamic time warping-based imputation for univariate time series data. Pattern Recogn Lett 139:139–147

Dwivedi D, Mital U, Faybishenko B, Dafflon B, Varadharajan C, Agarwal D, Williams KH, Steefel CI, Hubbard SS (2022) Imputation of contiguous gaps and extremes of subhourly groundwater time series using random forests. J Mach Learn Model Comput 3(2)

Gao Y, Merz C, Lischeid G, Schneider M (2018) A review on missing hydrological data processing. Environ Earth Sci 77(2):1–12

Keogh EJ, Pazzani MJ (2001) Derivative dynamic time warping. In: Proceedings of the 2001 SIAM International Conference on Data Mining, SIAM, pp 1–11

Kim M, Baek S, Ligaray M, Pyo J, Park M, Cho KH (2015) Comparative studies of different imputation methods for recovering streamflow observation. Water 7(12):6847–6860

Kulanuwat L, Chantrapornchai C, Maleewong M, Wongchaisuwat P, Wimala S, Sarinnapakorn K, Boonya-Aroonnet S (2021) Anomaly detection using a sliding window technique and data imputation with machine learning for hydrological time series. Water 13(13):1862

Lai WY, Kuok K (2019) A study on bayesian principal component analysis for addressing missing rainfall data. Water Resour Manage 33(8):2615–2628

Law SM (2019) STUMPY: a powerful and scalable python library for time series data mining. J Open Source Softw 4(39):1504

Li L, Wang H, Wang Y, Chen M, Wei T (2022) Improving iot data availability via feedback-and voting-based anomaly imputation. Futur Gener Comput Syst 135:194–204

Maleki S, Maleki S, Jennings NR (2021) Unsupervised anomaly detection with lstm autoencoders using statistical data-filtering. Appl Soft Comput 108

Moahmed TA, ElGayar N, Atiya AF (2014) Forward and backward forecasting ensembles for the estimation of time series missing data. In: IAPR Workshop on Artificial Neural Networks in Pattern Recognition, Springer, pp 93–104

Nogra JA, Romana CLS, Maravillas E (2019) LSTM neural networks for Baybáyin handwriting recognition. In: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), IEEE, pp 62–66

Osman MS, Abu-Mahfouz AM, Page PR (2018) A survey on data imputation techniques: Water distribution system as a use case. IEEE Access 6:63279–63291

Peterson LE (2009) K-nearest neighbor. Scholarpedia 4(2):1883

Peugh JL, Enders CK (2004) Missing data in educational research: A review of reporting practices and suggestions for improvement. Rev Educ Res 74(4):525–556

Phan TTH (2020) Machine learning for univariate time series imputation. In: 2020 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), pp 1–6, 10.1109/MAPR49794.2020.9237768

Pratama I, Permanasari AE, Ardiyanto I, Indrayani R (2016) A review of missing values handling methods on time-series data. In: 2016 International Conference on Information Technology Systems and Innovation (ICITSI), IEEE, pp 1–6

Sagheer A, Kotb M (2019) Time series forecasting of petroleum production using deep lstm recurrent networks. Neurocomputing 323:203–213

Stekhoven DJ, Bühlmann P (2012) Missforest-non-parametric missing value imputation for mixed-type data. Bioinformatics 28(1):112–118

Tormene P, Giorgino T, Quaglini S, Stefanelli M (2009) Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artif Intell Med 45(1):11–34

Yang JH, Cheng CH, Chan CP (2017) A time-series water level forecasting model based on imputation and variable selection method. Comput Intell Neurosci 2017

Yeh CCM, Zhu Y, Ulanova L, Begum N, Ding Y, Dau HA, Silva DF, Mueen A, Keogh E (2016) Matrix profile I: All pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, pp 1317–1322

Yuan H, Xu G, Yao Z, Jia J, Zhang Y (2018) Imputation of missing data in time series for air pollutants using long short-term memory recurrent neural networks. In: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, pp 1293–1300

Zhang Y, Thorburn PJ (2021) A dual-head attention model for time series data imputation. Comput Electron Agric 189