Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **8**, Number **3**, August **2023**

Combining Convolution and Transformer for Missing Time Series Data Imputation

Yi-Fan Wang

School of Information Network Security
People's Public Security University of China
Beijing, 100038, China
2021211477@stu.ppsuc.edu.cn

Shuai-Yu Bu

Institute of Electrical Engineering
Chinese Academy of Sciences
Beijing, 100190, China
bushuaiyu@mail.iee.ac.cn

Jing-Hua Yan*

School of Information Network Security
People's Public Security University of China
Beijing, 100038, China
ppsuc477@sina.com

Zhi-Wen Hou

School of Information Network Security
People's Public Security University of China
Beijing, 100038, China
houzhiwen@stu.ppsuc.edu.cn

Ling-Bin Bu

School of Information Network Security
People's Public Security University of China
Beijing, 100038, China
2018211241@stu.ppsuc.edu.cn

Fan-Xu Meng

School of Information Network Security
People's Public Security University of China
Beijing, 100038, China
1015215859@qq.com

*Corresponding author: Jing-Hua Yan

Received December 9, 2022, revised March 9, 2023, accepted May 19, 2023.

Abstract. *When dealing with time series data in real life, it is not possible to avoid generating missing data due to network signal interruptions, environmental conditions and sensor failures. Although many methods for imputing data have been proposed, there are still limitations. To begin with, most imputation models do not consider the data's local information, such as its trend. Second, there will always be gaps in time series data at random points. However, most previous research used particular imputation based on known missing positions, and no simulation trials for the case where all positions are missing are done. This is not applicable for future imputation tasks. To this end, this paper proposes the time series combining convolutional and transformer (TimeCT) model, which fuses the transformer's global information modeling capability with convolutional neural networks to extract local features and predict missing location data in both directions for both historical and future data. Meanwhile, we develop a deficient dataset constructor algorithm to simulate the location of missing values in a real time series data set (K-fold missing construction). Finally, we conduct numerous experiments to confirm the model's effectiveness. Our model generates better results when compared to the most sophisticated comparison methods. This method addresses the issue of filling in numerous missing cases to improve the accuracy of subsequent tasks like prediction and classification*

Keywords: time series data, missing data, imputation, Convolution, Transformer, missing construction algorithm.

1. **Introduction.** In the fields of finance, meteorology, hydrology, signal processing, and engineering technology, time series data is frequently used. In time series analysis, imputation of missing values has always been a major concern [1]. Real-world time series are frequently collected for various reasons and always contain missing entries, resulting in incomplete data, which affects data quality and makes downstream time series classification or regression problems difficult to solve. Furthermore, local information on the context of missing data is rarely used, and missing data might occur at any time, making the location of missing data distribution generally uncertain. Then the information utilized to generate the missing at different locations is different. It makes it difficult to convert simple static imputation to dynamic imputation, which makes the imputation work harder when the missing data position is unknown.

A neural network model would be a good choice to solve these issues. The neural network will analyze the data and look for correlations between the data for imputation. As we will discuss in the next section, there is a large literature devoted to solving missing and irregular patterns in time series datasets. By reading some related literature [2], we consider the benefits of convolutional neural networks. We propose TimeCT, a combining convolutional and transformer depth imputation approach, to address these concerns. Through feature fusion [3], the model combines the local characteristics collected by CNN with the global features extracted by transformer [4]. As a result, we turn the missing value imputation problem into a missing position prediction problem using the missing position context data. Meanwhile, we simulate a missing situation and propose a K-fold missing construction algorithm to generate missing in all positions in a dataset. Training the model on this dataset improves the model's applicability, while increasing the amount of data makes model training easier. The experimental findings demonstrate that the TimeCT model performs better when filling in the missing data. The following are this paper's main contributions:

• 1) We design a combinatorial convolution and transformer in-depth imputation model called TimeCT. The model uses both CNN feature extraction and transformer's content attention and location self-attention mechanisms [5] to capture global information from temporal data, allowing the model to consider both local and global information when

predicting missing values and thus improving overall model performance. It has outperformed CNN and transformer in terms of performance.

• 2) We propose a construction algorithm for missing data (K-fold missing construction) to simulate the situation where missing data may occur at any position, increasing the data sample and extending the applicability of the model.

• 3) We validated the TimeCT model on the Australian water quality time series dataset and the Beijing air pollution time series dataset to show its validity.

The remainder of the paper is structured as follows: We briefly touch on related work in Section 2. In Section 3, we formally introduce the setup and problem description of the time series imputation task. We present the overall architecture and each module of the TimeCT model, as well as our time series imputation method. Then we describe the K-fold missing construction algorithm in detail. To assess the model's performance, we carry out extensive experiments in Section 4. Finally, we conclude the work in Section 5.

2. **Related Works.** Missing data strongly affects the performance of downstream tasks in time series datasets, and time series data imputation challenges [6] are particularly widespread in real life. And most of the datasets for any downstream tasks are subjected to data imputation preprocessing operations. Although there has been a lot of research into dealing with missing values, such as simple deletion methods, they may overlook critical information. There are also statistical methods such as eigenvalue padding, mean, median, and common values of statistical data for imputation. The multiple interpolation method [7], using multiple imputation by chained equations (MICE), in various iterations to estimate the missing values.

In addition to statistical methods, many researchers used machine learning [8] models for data imputation, such as the nearest node algorithm [9], multilayer perceptron [10], and random forest [11]. These are commonly used in data imputation algorithms. These methods, on the other hand, ignore the time dependence of time series data.

In recent years, many researchers have used deep learning methods to process time series data [12], and recurrent neural networks (RNN) have shown good results in modeling time series data. For example, Jiang et al. [13] used RNN for short-term urban traffic prediction. Meanwhile, various RNN-based data imputation and prediction approaches have been presented [14, 15], such as long-short time memory networks (LSTM) [16]. Li et al. [17] used long-short time memory with support vector machines (SVM-LSTM) to fill in missing values in real-time traffic monitoring data. Generative adversarial networks (GAN) are also often used for data imputation. Yoon et al. [18] utilized a generative adversarial network to fill time series data, while Luo et al. [19] used an end-to-end generative adversarial network to fill the missing data. Furthermore, convolutional neural networks perform well in the prediction of time series. Despite the fact that convolutional neural networks play a critical role in solving computer vision problems, and time series prediction is a very different challenge, we may still use CNN's benefits to help with time series prediction tasks [20] or other time series tasks [21, 22]. For example, Guo et al. [23] proposed using one-dimensional convolution with GRU and other algorithms for short-term water consumption. There are also other traditional time series forecasting models that use causal convolution internally, such as the temporal convolutional network [24] (TCN) and the WaveNet [25] model. It can be seen that convolution operations have good results in time series prediction tasks.

In the prediction [26] of time series data, the recent popular transformer model has shown improved results [27]. For example, Wu et al. [28] used the transformer model to accurately predict influenza prevalence using a time-series dataset of influenza epidemic

cases. Zerveas et al. [29] proposed a new framework for learning multivariate time series representations based on the transformer encoder architecture and demonstrated the effectiveness of this architecture by evaluating it on a publicly available dataset. In the field of data imputation, Yu et al. [30] proposed a geometry-sensitive point cloud imputation transformer for missing point cloud generation by modeling the point cloud as a set of disordered point agents and applying the transformer's encoder-decoder structure. Wan et al. [31] took advantage of transformer's ability to obtain global structural information to generate structural information for the image's missing component. These results demonstrate that transformer has a unique advantage in solving time series problems. This approach represents each input sequence element simultaneously by taking into account the context (future-past) of each input sequence element and learns complex patterns from time series data using a self-attentive mechanism [32]. While taking into account many correlations between various representation subspaces and input elements, multiple attention headers are especially well suited for time series data.

Based on the transformer model's efficiency in the task of time series data imputation and prediction, we propose a combining convolutional and transformer model. This model not only uses transformer's self-attention mechanism to learn the complex structure of time series data, but also uses convolution to extract the data's local information. The multi-headed attention mechanism now focuses on the shifting trend of time series data rather than just the information of a single data point. This is the first time we apply the transformer model to the imputation of time series data. At the same time, we construct the missing dataset using the algorithm we designed to simulate the possible cases of missing and verify the validity of this model.

3. **Methods.**

3.1. **Problem Description.**

Due to objective factors such as equipment malfunction, lack of constant power supply, data transmission issues, and so on, there is always a completely random absence of certain data in most sensors. It is possible for data to be absent at any point in time in time series data. This can lead to different prior knowledge being utilized at different locations of the absence. As a result, determining the location of the missing values is critical.

To formulate this type of problem, we assume that the time series data produces completely random missing at each moment. As a technique to verify the model's and algorithm's stability, we use a sliding window with a window of 1 to create a missing value for 24 hours of time series data. To test the model's scalability, the missing construction method with a sliding window of 4 is utilized to simulate the data and generate continuous missing [33]. The 24-hour time series is denoted as:

$$T = t_1, t_2, \ldots, t_i, \ldots, t_{24} \tag{1}$$

where denotes the data at the moment.

After constructing the positions of the missing values, the time series can be expressed as:

$$T^{'} = t_1, t_2, \ldots, t_{i-1}, 0, t_{i+1}, \ldots, t_{24} \tag{2}$$

or

$$T^{'} = t_1, t_2, \ldots, t_{i-1}, 0, 0, 0, 0, t_{i+4}, \ldots, t_{24} \tag{3}$$

where (2) indicates a simple random position missing in the time series data and (3) denotes four consecutive missing data points. In the next chapter, we will describe the specific construction methods in detail.

To solve the problem, we design a deep learning model that learns the relationship between sensor monitoring parameters in the past and future to fill in missing values based on missing positions in time series data. Furthermore, we offer a way for producing a large number of training samples for both training and validation. Finally, we use the test set to train and assess the proposed model.

3.2. **Models & Algorithms.** The structure of the data imputation method proposed in this paper is shown in Figure 1. For long time series data, processing with models such as RNN and LSTM does not solve the problem of gradient disappearance and gradient explosion, resulting in LSTM not easily obtaining global information. However Transformer solves this problem and is able to capture the global information between data more accurately. In addition, the combination of CNN's local feature extraction capability to find local correlations between data allows for more accurate complementation. Among them, the TimeCT model effectively reduces local redundancy by improving the transformer model and adding a convolutional layer in Self-Attention while capturing global dependencies using the transformer. This not only improves the ability to model contextual information in time series, but it also allows the transformer to capture global information, which helps predict missing location data.

We also propose a K-fold missing construction algorithm by simulating missing data on a real-time series dataset to give a data basis for data imputation, due to the randomness and continuity of missing data.

Data imputation should be able to see both past and future information to fill in the missing location data, unlike the conventional approach of using only past data to predict future data. So we get inspiration from bi-directional complementary time series data in Bi-Directional GAN [34] and Multi-Directional RNN [35]. We split the missing data set into past and future data using the missing location as a barrier. Using past and future data, the model will predict missing values in both directions.

In Figure 1, we construct the complete data as missing data by the proposed missing construction algorithm and divide the remaining complete data into historical data and future data. We then fed the two segments of data into the model separately for training. The input time series is convolved, embedded, and fused to obtain the extracted feature vectors. The features are sent into the transformer's encoder layer, which outputs a forward embedding vector or a backward embedding vector. The encoder layer is made up of two layers, each of which has two sub-layers: the first is a multi-headed attention mechanism that computes the input self-attention, and the second is a simple fully connected layer. Residual networks and normalized (ADD & Norm) are employed in each sub-layer. Then the vector passes through the decoder, which is made up of a linear layer, to produce the prediction results. The bidirectional loss function MSELoss is also designed for training, and the output of the result with the minimum loss is obtained for final filling.

3.3. **TimeCT model.** TimeCT employs the transformer's encoder's multi-header attention method, such as the encoder shown in Figure 1. The encoder takes the missing values from the input time series and maps them to the hidden layer. To forecast the value of the missing position, the deep neural network (DNN) functions as a model decoder.

The model presented in this paper extends the classic transformer model's encoder with a convolutional layer. We perform convolution operations to compute the query and key of the multi-headed attention mechanism using a convolution kernel of size greater than

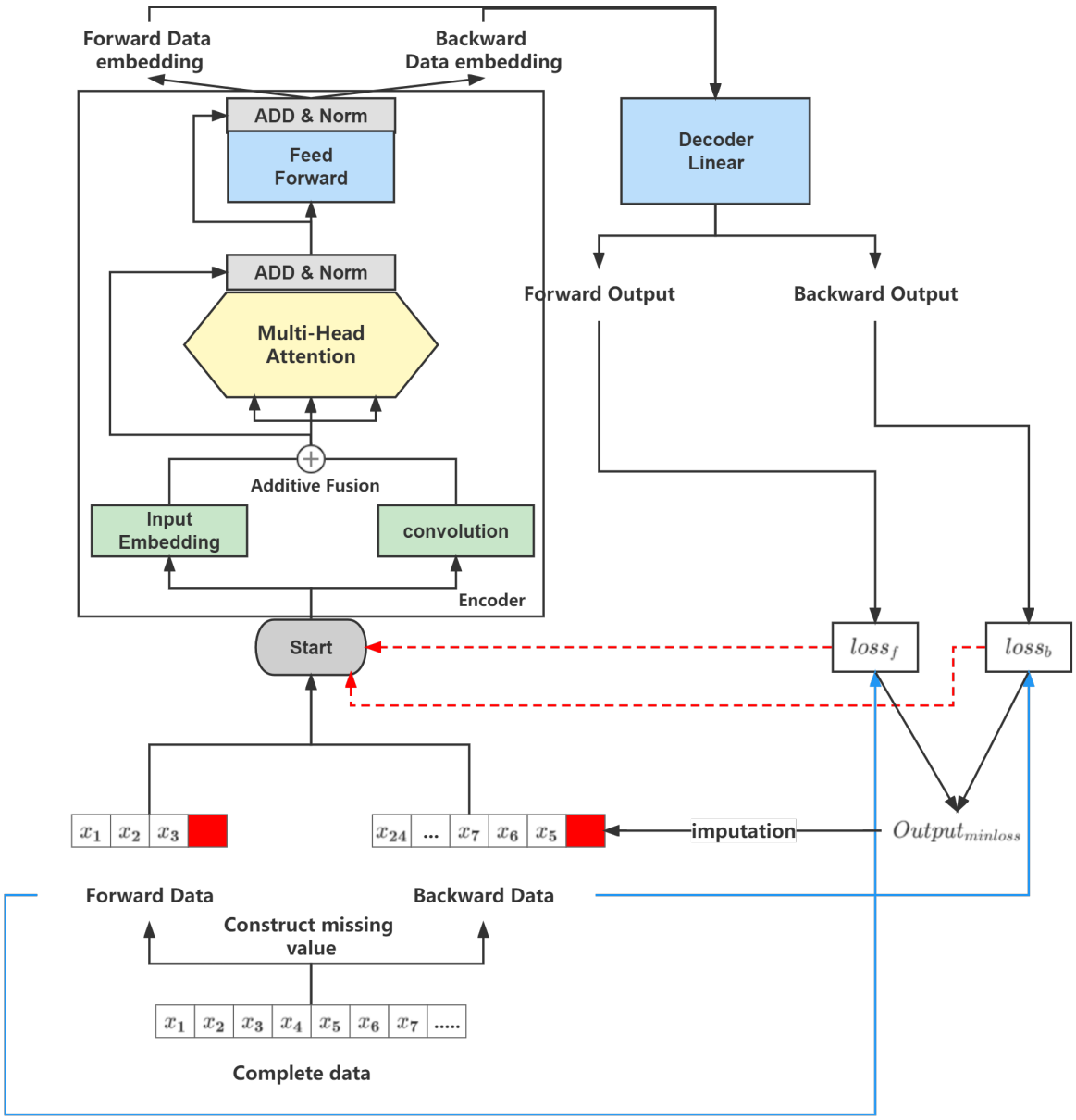

FIGURE 1. Data Imputation Method Structure

1 [36, 37], so that the attention can pay more attention to the local context of the data itself and extract the relevant features for matching.

For the time series with missing values $x = x_1, x_2, ..., x_n$ where $x_i \in R^i$, each of x_i is a multi-source time series with d-dimensional features. For the input temporal data, in order for the network to identify its sequential relationships, the position information of each temporal data must be provided to the transformer. So, we add position encoding, which internally uses a linear transformation of the sum function to give the model the sequence information of the temporal data, mapping the position information to a point in the position space, thus giving relative position information to the input vector. Equation (4) is the embedding operation of the input sequenceX, the output embedding vector $X_{embedding}$, Equation (5) is the location encoding operation of the embedding vector, the

output vector $X_{embedding}$:

$$X_{embedding} = Embedding(X) \tag{4}$$

$$X_{embedding} = pos\_encoder(X_{embedding}) \tag{5}$$

where Embedding is the vector encoding function and *pos_encoder* is the position encoding function.

To extract the local features from the input data, we perform a one-dimensional convolution operation (matrix multiplication of the input) with a layer of convolution kernel size of 2×2 and a step size of 1 for the input data. Convolution improves the model's ability to simulate local contextual information in time series, lowering the impact of outliers on prediction outcomes and increasing prediction accuracy. So, we use the convolution operation on the input X to extract its local features and output the feature vector X_conv after the activation function:

$$X\_conv = Sigmoid(Convolution(X)) \tag{6}$$

where Sigmoid is the activation function, and Convolution is a one-dimensional convolution function with a convolution kernel size of 2×2 and a step size of 1.

Subsequently, in order that the long-range dependency information obtained by the transformer module is not missing, the convolved temporal data is fused with the encoded data before calculating the query, key, and value:

$$Input = X\_embedding + X\_conv \tag{7}$$

$$Q = Linear_q(Input) = Input * W_Q \tag{8}$$

$$K = Linear_k(Input) = Input * W_K \tag{9}$$

$$V = Linear_v(Input) = Input * W_V \tag{10}$$

$$X_{att} = Self\_Attention(Q, K, V) \tag{11}$$

where Linear is the linear layer, W_Q, W_K, and W_V are the linear layer learning parameters, and Self_Attention is the self-attention mechanism function.

Multiple groups of self-attention are composed of multi-head attention. The attention mechanism computes $(W_0^Q, W_0^K, W_0^V), (W_1^Q, W_1^K, W_1^V)$ focus on different contexts, respectively. Since the attention score in the whole time series prediction only evaluates the association between single time points, such as in (A) in Figure 2, which only considers values with similar time points without taking into account its own local data information, we add a convolutional layer to enhance the modeling ability of the local context, as in (B) in Figure 2. After using convolution, the model focuses on its own information, such as the change trend of the data.

3.4. **K-fold missing construction algorithm.** We offer a method for processing the dataset that simulates the condition where time series data may be missing at any point. To construct missing data, different values might be set according to the sliding window K. Here is an example of K = 1 and K = 4. The method of construction using a sliding window of 1 emphasizes the randomness of the missing position more. The time series data may be missing at any time, as seen in Figure 3, where red indicates the missing data. To construct a missing value, the sliding window is set from 24 hours to 0 hours with a step size of 1, and K = 1, referred to as the 1-fold missing construction method. As illustrated in Figure 4, the construction method with a sliding window of 4 simulates the condition of continuous missing time series data.

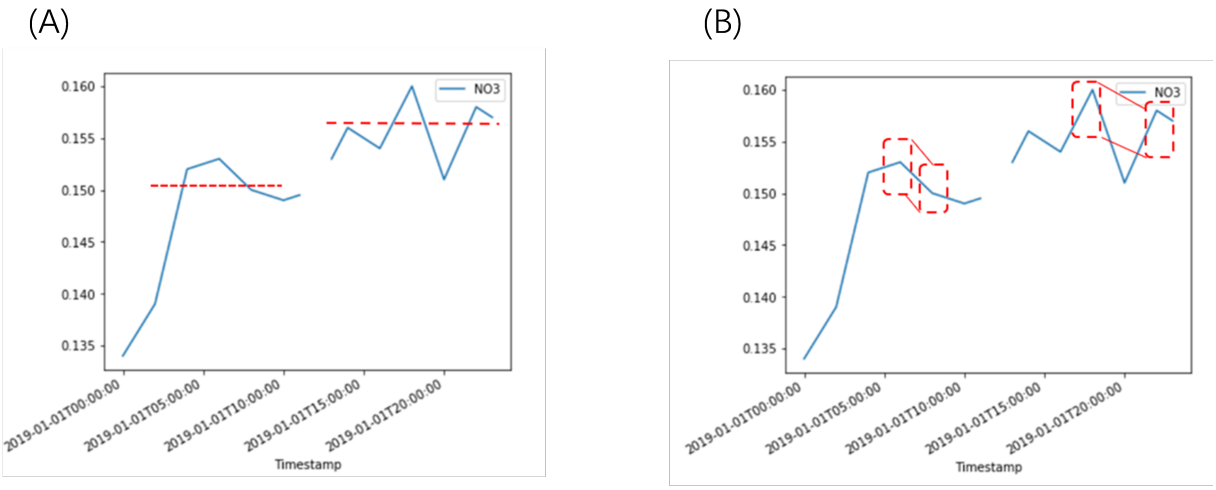

FIGURE 2. Australian water quality dataset data for a certain time period. (A) Attentional focus without convolutional layers (B) Attentional attention part after adding convolutional layers

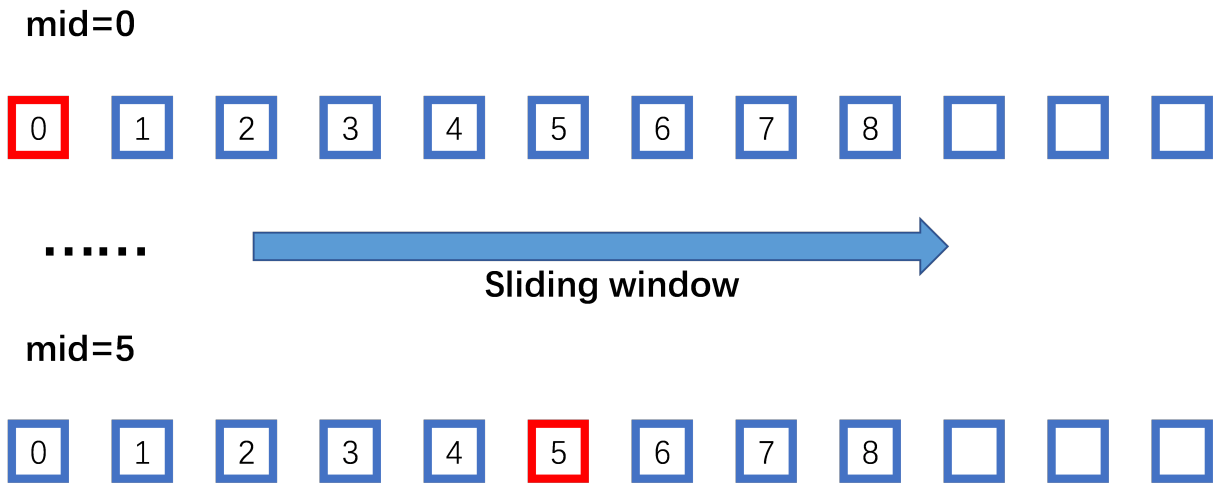

FIGURE 3. Missing value construction algorithm with 1 step size and 1 sliding window

After constructing the missing data set, the whole data set will be divided into past and future data. When the missing position is 5, for example, there are 5 data points in the past moment and 19 data points in the future moment of the missing data in Figure 3.

This method generates a large amount of training and validation data, which is essential for training deep learning models. The large sample size allows the models to learn more effectively how to exploit historical and upcoming data sets in the presence of diverse missing places.

4. Experiment.

4.1. Datasets and pre-processing. We process and experiment on the groundwater resources time series dataset obtained from the Australian government's water quality monitoring program and the Beijing air pollution time series data in order to assess

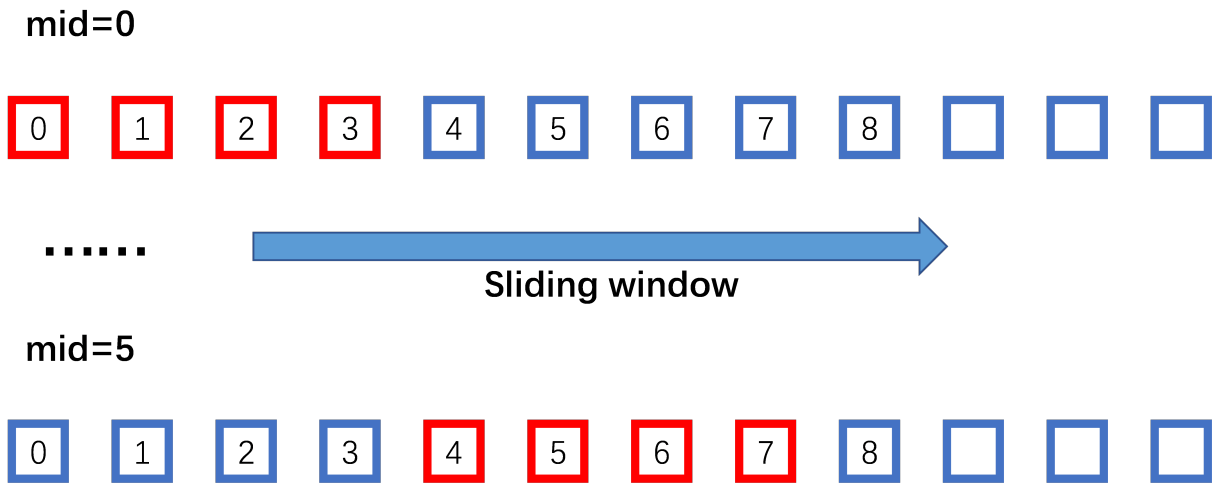

FIGURE 4. Missing value construction algorithm with 4 step size and 4 sliding windows

the efficacy of the created model. These two datasets contain multi-source features and are typical time series datasets. The training and validation datasets are comprehensive datasets that are used to check the precision of the imputation performed in this paper.

For the Australian Government's water quality multi-source time series data, the data included sediment and nutrient monitoring at 43 sites in 20 major catchments. In this paper, we only use the measured nitrogen (NO_3) as the missing feature, and other variables are auxiliary features. Other features affect the missing features, so we input them into the model together as auxiliary features, so that the model can find the relationship between the auxiliary features and the missing features, and thus fill them in better. When the data volume is limited, we therefore utilize a total of 8736 time series data points from January 1 to October 1 as the training set and data from October 1 to December 31 as the validation set. Due to the small amount of data, the model cannot be trained and evaluated adequately if the original data is utilized directly for the experiment. Using the K-fold missing data construction algorithm, the data is divided into a group of 24 hours by day, and the missing data of the experiment is constructed by sliding window of 1 with a step size of 1. A set of time data will be used 24 times, so it is expanded from the original 8736 data to 209664 data for training. As a result, the amount of data increases and the model can be trained better.

For the Beijing air pollution multi-source time series data, the source is the hourly weather and air pollution indices collected by the U.S. Embassy in Beijing for a total of five years from 2010 to 2014. The dataset includes dates, PM2.5 concentration, dew point, temperature, wind direction, wind speed, cumulative hourly snow, and cumulative hourly rainfall. In this paper, we use PM2.5 as the missing feature. Then we utilize the data from January 2, 2010 to January 1, 2013 (a total of 26,304 time series data) as the training set, and the data from December 31, 2012 to December 31, 2014 as the validation set (a total of 17,544 data). This dataset has a huge amount of data, which allows the model to be better trained and give better results.

In the above two time series datasets, we apply the proposed K-fold missing construction algorithm to accomplish 1-fold missing construction and 4-fold missing construction, respectively. The 1-fold missing construction makes each position produce a missing, which can make the model focus more on learning the information of the missing value position. The 4-fold missing construction is focused on solving the case of continuous

missing data. Specifically, when the missing occurs at the very beginning and the very last moment of the time series, only the data from the future or past moments is used to complete the data, respectively. In summary, the missing data processed in this paper contains all possible locations where the missing may occur and the possibility of continuous missing, which is more applicable and scalable.

4.2. **Experimental setup.** In the experiments, we implement the TimeCT model for training and experimental validation of the constructed missing dataset using PyTorch. The hyperparameters of the model are determined by the performance of the model on the validation set. We employ the Adam optimizer with an initial learning rate of 0.001 during the training phase and adjust the learning rate using StepLR of Pytorch, setting the adjustment round interval to 100 rounds, with each adjustment 0.1 times. The batch size is 64. 500 rounds of training are conducted simultaneously, and the training is terminated when the validation set error does not drop in 100 consecutive rounds. And when the validation set error is reduced, the hyperparameter is the hyperparameter of the model. We constructed 1 missing, 2, 4, 12, and 24 consecutive missing for experiments to verify the performance of the model.

4.3. **Comparison with Current Data Imputation Methods.** We compare the TimeCT model with the following baseline used for data imputation:

• MEAN: The mean fill algorithm fills in the missing position data by calculating the mean of the data. This is the most basic interpolation method, which is very fast to compute, but has the disadvantage that mean interpolation reduces the variance difference of the data.

• LOCF algorithm: Last observation carried forward (LOCF), fills with the last observation before the missing one.

• Kalman filtering: The Kalman filtering algorithm effectively estimates the system state from the system input and output values by using the state equation of a linear system. We use Kalman smoothing for missing value estimation.

• RF algorithm: Multiple decision trees are built by Random Forest to fill in the missing values, resulting in random and uncertain data. This better reflects the true distribution of this unknown data, further ensuring the accuracy and reliability of the obtained filled values. We use a training set of samples without missing values for all data (including historical and future data) with missing positions removed, and a test set of samples with missing values, and use the random forest algorithm for regression modeling to predict the missing values.

• MICE algorithm: Multiple interpolation of chained equations, this method fills in the missing data in the dataset through a series of iterations of the predictive model. In each iteration, each specified variable in the dataset is estimated using other variables in the dataset, and these iterations continue to run until convergence is satisfied. The default fill strategy taken is Predictive Mean Matching (PMM). It has the advantage of being able to interpolate appropriate values when the assumption of normality does not hold, and the disadvantage is that it is difficult to determine the random error term.

• RNN: Recurrent neural networks perform well in time series prediction, so we use RNN to forecast missing location data based on the remaining complete data as past and future data.

• LSTM: Long and short-term memory networks are a particular form of RNN that can learn long-term temporal dependence in prediction tasks. Similarly, we use the past data to predict the missing positions and the future data to predict the missing position data in reverse.

As evaluation measures, we employ the mean absolute error (MAE), root mean square error (RMSE), and symmetric mean absolute percentage error (SMAPE):

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}|y - y_{hat}| \tag{12}$$

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(|y - y_{hat}|)^2} \tag{13}$$

$$SMPE = \frac{100}{n}\Sigma_{i=1}^{n}\frac{|y - y_{hat}|}{|y| + |y_{hat}|}\% \tag{14}$$

where y and y_{hat} are the true and predicted values, respectively. The smaller the value of the evaluation index we chose above, the smaller the model filling error and the better the filling effect.

Table 1. Performance comparison of different data imputation methods on the constructed Australian water quality missing data set

Model	MAE	RMSE	SMAPE
MEAN	23.31	61.00	116.66
MICE	8.68	34.96	51.35
LOCF	5.58	23.89	29.54
Kalman filtering	3.28	15.11	23.71
RF	16.76±0.46	45.06±1.65	57.27±0.36
RNN	6.99±0.07	25.88±0.16	48.20±0.13
LSTM	6.69±0.08	25.87±0.17	47.58±0.17
TimeCT	2.82±0.11	11.85±0.13	21.50±0.10

Table 2. Performance comparison of different data imputation methods on the constructed missing data set of Beijing air pollution

Model	MAE	RMSE	SMAPE
MEAN	20.42	48.68	119.96
MICE	1.65	6.31	28.58
LOCF	1.16	4.86	17.89
Kalman filtering	0.87	3.44	14.38
RF	3.28±0.24	11.43±0.46	542.65±0.41
RNN	1.07±0.10	3.82±0.07	20.19±0.17
LSTM	0.97±0.13	3.69±0.05	18.91±0.10
TimeCT	0.63±0.12	2.60±0.12	10.38±0.10

As demonstrated in Tables 1 and 2, the TimeCT model performs best in terms of evaluation metrics on the missing dataset produced using the 1-fold missing construction strategy. The missing case constructed at K = 1 simulates the possibility of random missingness in time series data at any point in time. The model has a good filling effect in this case indicating that the model can handle such missing cases well.

It should also be addressed whether the model is still viable in the presence of continuous missing data. To validate the model, we created a missing dataset with 2 versus 4 consecutive missing points at a time using the 2-fold and 4-fold missing construction

Table 3. Comparison of model performance in the Australian water quality dataset at K=2 and K=4

	K=2			K=4		
Model	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
MEAN	23.35	78.86	113.28	24.38	38.30	117.32
MICE	10.20	35.90	53.14	12.79	28.89	61.66
LOCF	9.20	45.08	41.97	16.42	7.64	66.04
Kalman filtering	10.90	41.58	66.05	17.58	62.97	122.12
RF	22.86±0.72	52.57±0.32	74.36±0.33	19.65±0.59	32.62±1.39	68.64±2.23
RNN	6.21±0.59	28.24±1.39	36.47±2.23	10.52±0.11	24.08±0.17	51.83±0.16
LSTM	5.81±0.10	26.85±0.17	30.11±0.18	9.75±0.10	23.02±0.17	50.03±0.18
TimeCT	4.89±0.34	19.28±0.42	28.47±0.72	8.29±0.19	20.79±0.30	48.76±1.53

Table 4. Comparison of model performance in Beijing air pollution data set at K=2 and K=4

	K=2			K=4		
Model	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
MEAN	19.79	49.31	114.39	20.11	29.78	117.75
MICE	2.06	7.53	0.71	2.23	4.98	36.02
LOCF	1.96	10.53	26.17	2.99	6.83	39.02
Kalman filtering	4.12	13.52	56.63	6.94	22.09	116.87
RF	4.71±0.41	13.45±0.52	57.42±0.12	4.28±0.12	9.05±0.34	54.76±0.61
RNN	1.80±0.19	4.69±0.22	16.76±0.31	.08±0.20	5.05±0.21	29.53±0.52
LSTM	1.43±0.21	4.33±0.34	15.59±0.55	1.81±0.32	8.79±0.41	28.97±0.62
TimeCT	0.99±0.15	4.20±0.21	14.93±0.42	1.43±0.21	3.93±0.33	21.26±0.44

Table 5. Comparison of model performance in the Australian water quality dataset at K=12 and K=24

	K=12			K=24		
Model	MAE	RMSE	SMAPE	MAE	RMSE	SMAPE
MEAN	24.60	133.06	133.52	28.00	197.17	155.16
MICE	12.51	97.68	62.25	22.22	165.50	91.48
LOCF	19.47	150.16	974.68	23.51	183.41	98.04
Kalman filtering	22.94	171.70	164.84	23.71	125.37	176.13
RF	33.03±0.62	190.95±1.22	108.49±1.32	34.94±0.71	234.73±1.29	119.86±1.46
RNN	13.53±0.31	67.20±0.42	57.25±0.44	22.08±0.10	75.05±0.11	89.53±0.72
LSTM	13.13±0.33	6.13±0.60	56.79±0.42	20.91±0.50	74.24±0.66	8.92±0.32
TimeCT	13.05±0.35	66.02±0.43	56.32±0.61	18.63±1.01	67.59±0.79	8.03±0.44

algorithms on both datasets. The results are shown in Table 3 and Table 4. In order to verify the maximum number of consecutive absences that the model can fill, we again took K = 12 and K = 24 for experiments on the Australian water quality dataset. In the experiment, in order to use more data to predict the missing values more accurately, we expanded the time length to 72 hours and used data from the past day versus the future

day to predict the intermediate missing parts. The results are shown in Table 5. It can be seen that even in the case of continuous missing, the TimeCT can still produce a good filling effect compared with other models. However, as the number of consecutive missing increases, the difficulty of filling grows, resulting in a decrease in model performance.

From all the above results, it is clear that the majority of machine learning and conventional statistical methods are not particularly good at solving the issue of missing time series data. Among them, the Kalman filter algorithm produces better results for a small number of missing, but the performance decreases significantly as the number of missing increases with increasing K values. In contrast, recurrent neural networks in deep learning have better results for filling time series. However, for LSTM, although improvements have been made to RNN with improved performance, the gradient disappearance and explosion problems are still not solved for long sequences, resulting in LSTM not being able to obtain global information easily, which can be solved by transformer architecture. The transformer has a stronger long-term dependency modeling capability, so it is easy to obtain global structural information. At the same time, different heads in the transformer's multi-head attention can focus on different patterns, so that long-term dependencies and short-term dependencies can be modeled simultaneously. The transformer with added convolutional layers places more emphasis on local information, allowing the head to focus not only on global information, but also on local contextual information. The experiment results demonstrate that the TimeCT model can successfully capture the local information of the context in multivariate time series complementation and produce superior prediction of the missing location data, outperforming LSTM.

4.4. **Ablation experiments.** To further validate the model, two variants were created to test the effect of convolutional layers on the model. Variant 1 uses the traditional transformer model to fill in the constructed Australian water quality missing data set using the fill method above. Variant 2 operates on the same dataset as the model after adding a convolutional layer to variant 1 (i.e., the TimeCT model designed in this paper).

Variant 1 (Transformer): The basic model includes a traditional transformer encoder and a DNN decoder.

Variant 2 (Conv+Transformer): Add convolutional layers to the traditional transformer, which is the model TimeCT for time series data imputation.

Table 6. Ablation experiment

Variant	MAE	RMSE	SMAPE
Variant 1 (Transformer)	3.66	13.93	27.98
Variant 2 (Conv+Transformer)	2.71	11.77	21.44

Doing so was able to demonstrate that the added convolutional layers were effective in the data filling process and were not a result of the performance of the self-attentive mechanism in Transformer. As shown in Table 6, the convolutional layer in TimeCT is effective in the imputation of the time series data. In addition, we also compare the performance of variant 1 with that of LSTM in Table 1. The traditional transformer makes processing time series data better than LSTM because of its multi-headed attention mechanism. And by ablation experiments, we can get that, for the imputation of time-series data, the convolutional layer helps to focus on the trend of past and future data, thus enabling more accurate prediction of missing location data. Rather than focusing only on the association between a certain data point and the traditional transformer, TimeCT enhances the ability to model the local context, providing sufficient information

for the complementary task. The ablation experiment, as shown above, demonstrates the model's validity on time series data imputation.

5. **Conclusions.** When handling data from real-world applications, dealing with missing values and incomplete time series is a labor-intensive, unpleasant, and unavoidable task. Most of the imputation models in the current study lack the ability to model the local information of the time-series data, and they all fill in the discovered missing locations with low robustness. In this paper, we design a model (TimeCT) to deal with the missing time series data problem. It is based on the transformer model and performs convolutional operations on time series data to extract local features as an effective complement to global features. In addition, we propose a dataset processing method named K-fold missing construction by constructing a sizable number of missing data points for training and testing. The task of filling in the missing time series data is then transformed into a prediction problem using the contextual information of the missing positions. Numerous experimental results demonstrated that the TimeCT model outperformed previous imputation techniques while also strengthening the model's emphasis on the change trend of the time series data itself and improving the correctness of the imputation data. In order to more precisely forecast the missing values, this method collects characteristics from both global and local information using deep mining techniques. However, the analysis using only the temporal dimension is biased, and in the future, the analysis can be merged with the spatial dimension at the same time to fill in the gaps in many dimensions. Among them, the use of graph neural networks to extract deeper spatial features and how to capture the continuity between time and space are also directions that we need to consider in the future. In addition to this, we should explore more the stochastic nature of the missing data, making the designed model sufficiently stable.

REFERENCES

[1] J. W. Graham, "Missing data analysis: Making it work in the real world," *Annual Review of Psychology*, vol. 60, pp. 549–576, 2009.

[2] I. A. Abdellaoui, and S. Mehrkanoon, "Deep multi-stations weather forecasting: explainable recurrent convolutional neural networks," *ArXiv*, 2020. [Online]. Available: http://arxiv.org/abs/2009.11239.

[3] H. Yan, Z. Li, W. Li, C. Wang, and C. Zhang, "ConTNet: Why not use convolution and transformer at the same time?," *ArXiv*, 2021. [Online]. Available: http://arxiv.org/abs/2104.13497.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[5] D. Povey, H. Hadian, P. Ghahremani, K. Li, L. Jones, A. N. Gomez, Ł. Kaiser, and S. Khudanpur, "A Time-Restricted Self-Attention Layer for ASR," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing* , ICASSP, 2018, pp. 5874-5878.

[6] M. Khayati, A. Lerner, Z. Tymchenko, and P. Cudré-Mauroux, "Mind the gap: an experimental evaluation of imputation of missing values techniques in time series," *Proceedings of the VLDB Endowment*, vol. 13, no. 5, pp. 768–782, 2020.

[7] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?," *International Journal of Methods in Psychiatric Research*, vol. 20, no. 1, pp. 40, 2011.

[8] C.-M. Chen, S. Lv, J.-R. Ning, and J. M.-T. Wu, "A Genetic Algorithm for the Waitable Time-Varying Multi-Depot Green Vehicle Routing Problem," *Symmetry*, vol. 15, pp. 124, 2023.

[9] S. Tak, S. Woo, and H. Yeo, "Data-Driven Imputation Method for Traffic Data in Sectional Units of Road Links," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1762–1771, 2016.

[10] H.-B. Chen, S. Grant-Muller, L. Mussone and F. Montgomery, "A study of hybrid neural network approaches and the effects of missing data on traffic forecasting," *Neural Computing and Applications*, vol. 10, no. 3, pp. 277–286, 2001.

[11] D. J. Stekhoven, and P. Bühlmann, "Missforest-Non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112–118, 2012.
[12] B. Lim, and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, pp. 20200209, 2021.
[13] B. K. Reddy, and D. Delen, "Predicting hospital readmission for lupus patients: An RNN-LSTM-based deep-learning methodology," *Computers in Biology and Medicine*, vol. 101, pp. 199–209, 2018.
[14] W. Cao, H. Zhou, D. Wang, Y.-T. Li, J.-A. Li and L. Li, "BRITS: Bidirectional recurrent imputation for time series," *Advances in Neural Information Processing Systems*, vol. 2018-December, pp. 6775–6785, 2018.
[15] R.-H. Jiang, X. Song, Z.-P. Fan, T.-Q. Xia, Q.-J. Chen, S. Miyazawa and R. Shibasaki, "DeepUrbanMomentum: An online deep-learning system for short-term urban mobility prediction," *32nd AAAI Conference on Artificial Intelligence*, AAAI, 2018, pp. 784–791.
[16] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, and T.-Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network," *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17–29, 2019.
[17] L.-C. Li, J. Zhang, Y.-G. Wang, and B. Ran, "Missing value imputation for traffic-related time series data based on a multi-view learning method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2933–2943, 2019.
[18] J. Yoon, J. Jordon, and M. Van Der Schaar, "GAIN: Missing data imputation using generative adversarial nets," *35th International Conference on Machine Learning*, ICML, 2018, pp. 9042–9051.
[19] Y.-G. Luo, Y. Zhang, X.-R. Cai, and X.-J. Yuan, "E2GaN: End-to-end generative adversarial network for multivariate time series imputation," *IJCAI International Joint Conference on Artificial Intelligence*, IJCAI, 2019, pp. 3094–3100.
[20] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*, vol. 10614 LNCS, pp. 3094–3100, 2017.
[21] K. Fauvel, T. Lin, V. Masson, É. Fromont, and A. Termier, "Xcm: An explainable convolutional neural network for multivariate time series classification," *Mathematics*, vol. 9, no. 23, pp. 1–21, 2021.
[22] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.
[23] G.-C. Guo, S.-M. Liu, Y.-P. Wu, J.-Y. Li, R. Zhou and X.-Y. Zhu, "Short-Term Water Demand Forecast Based on Deep Learning Method," *Journal of Water Resources Planning and Management*, vol. 144, no. 12, pp. 1–11, 2018.
[24] S.-J. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *ArXiv*, 2018. [Online]. Available: http://arxiv.org/abs/1803.01271.
[25] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "AWaveNet: A Generative Model for Raw Audio," *ArXiv*, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499.
[26] F. Zhang, T. Y. Wu, Y. Wang, R. Xiong, and L. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, no. 99, pp. 104555–104564, 2020.
[27] Q.-S. Wen, T. Zhou, C.-L. Zhang, W.-Q. Chen, Z.-Q. Ma, J.-C. Yan, and L. Sun, "Transformers in Time Series: A Survey," *ArXiv*, 2022. [Online]. Available: http://arxiv.org/abs/2202.07125.
[28] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case," *ArXiv*, 2020. [Online]. Available: http://arxiv.org/abs/2001.08317.
[29] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A Transformer-based Framework for Multivariate Time Series Representation Learning," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 2114–2124.
[30] X.-M. Yu, Y.-M. Rao, Z.-Y. Wang, Z.-Y. Liu, J.-W. Lu, and J. Zhou, "PoinTr: Diverse Point Cloud Completion with Geometry-Aware Transformers," *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 12478–12487.
[31] Z.-Y. Wan, J.-B. Zhang, D.-D. Chen, J. Liao, "High-Fidelity Pluralistic Image Completion with Transformers," *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 4672–4681.

[32] W.-J. Du, D. Cote, and Y. Liu, "SAITS: Self-Attention-based Imputation for Time Series," *ArXiv*, 2022. [Online]. Available: http://arxiv.org/abs/2202.08516.
[33] H.-Y. Mei, G.-H. Qin, and J. Eisner, "Imputing missing events in continuous-time event streams," *36th International Conference on Machine Learning*, ICML, 2019, pp. 7861–7885.
[34] M. Gupta, and R. Beheshti, "Time-series Imputation and Prediction with Bi-Directional Generative Adversarial Networks," *ArXiv*, 2020. [Online]. Available: http://arxiv.org/abs/2009.08900.
[35] J. Yoon, W. R. Zame, and M. Van Der Schaar, "Estimating Missing Data in Temporal Data Streams Using Multi-Directional Recurrent Neural Networks," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1477–1490, 2019.
[36] V. Lioutas, and Y.-H. Guo, "Time-aware large kernel convolutions," *37th International Conference on Machine Learning*, ICML, 2020, pp. 6128–6139.
[37] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.