TFITS: Time Series Imputation via Dual-Perspective Fusion of Temporal and Feature Views

Junfeng Yuan, Kangyan Li, Baofu Wu, Jian Wan, Jilin Zhang, Yuyu Yin, and Yan Zeng

***Abstract*—In the field of multivariate time series analysis, data completeness plays a crucial role in ensuring the accuracy and reliability of downstream tasks. However, in practical applications, factors such as measurement errors and equipment failures often lead to partial missing values in the data. Therefore, this paper introduces a novel model, focusing on how to efficiently handle missing values in multivariate time series and mitigate the potential negative impact of data incompleteness on subsequent tasks. In previous studies, researchers often adopted a single-perspective approach, separately considering the influence of the temporal dimension and the feature dimension, thereby overlooking the potential of dual-perspective fusion. This paper proposes TFITS, an innovative method for predicting missing values in multivariate time series data. TFITS approaches the problem from both the temporal and feature perspectives of time series data, leveraging the UNetFuse module to fuse feature maps generated from these two perspectives, thereby providing a more comprehensive solution to the missing value problem. Through this approach, TFITS can more effectively address missing values in multivariate time series data. Experimental results demonstrate that the TFITS model not only achieves excellent imputation performance but also exhibits superior and stable performance under different missing rate conditions.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/9732

***Index Terms*—Multivariate time series, Missing value, Dual-Perspective, Feature maps.**

I. Introduction

In multivariate time series data, each time point records different feature information collected by multiple sensors, and the continuous time series data exhibit regular patterns. As a result, multivariate time series play a crucial role in tasks such as data recording and state prediction, and are widely applied in various fields including energy[1], [2], electricity[3], [4], and healthcare[5], [6]. They are extensively used for tasks such as sensor data recovery[7] and network traffic missing data completion[8]. However, due to issues in the data collection process, such as equipment malfunctions, intentional interference, and incorrect data collection methods, time series data often suffer from incompleteness[9].

Imputing missing values not only requires restoring data completeness but also preserves the intrinsic patterns and structures of the data as much as possible to ensure the accuracy of downstream tasks. Methods for handling missing values can be primarily categorized into two types. The first type is the deletion method, which involves directly removing entire records containing missing values. This approach, however, leads to data discontinuity and disrupts the inherent patterns among data points. The second approach is imputation, which involves estimating missing values using observed data. Examples include matrix factorization-based[10], [11] methods, RNN-based methods[12], [13], [14], attention-based methods[15], [16], [17], generative model-based methods[18], [19], [20], and multi-view methods[21], [22]. Most of these methods are developed from a single-view perspective. Although multi-view models exist, they still exhibit limitations. For instance, these models fail to jointly consider the perspectives of temporal and feature, which are crucial for comprehensive analysis. Additionally, their fusion mechanisms lack universality, posing challenges for directly integrating additional perspectives when developing subsequent models.

Predicting missing values using observed data is simultaneously influenced by both the temporal dimension and the feature dimension. Specifically, on the one hand, the patterns exhibited by observed feature values at adjacent time points within the same feature dimension affect the estimation of missing values. On the other hand, the specific values of different features at the same time point also influence the estimation of missing values. However, previous studies have predominantly adopted a single-perspective approach for analysis. For instance, when approaching from the temporal perspective, multiple related feature values are mapped into a feature space, and different time points are embedded into a token that eliminates multivariate correlations. This approach overlooks the intrinsic relationships among different features, leading to insufficient modeling of multivariate correlations[23]. Conversely, when approaching from the feature perspective, the time series is inverted, and each feature variable is embedded into a token that eliminates temporal correlations. This approach ignores the temporal dependencies in the time series, resulting in inadequate modeling of temporal dynamics. Therefore, a single-perspective analysis inevitably leads to incomplete prediction results.

To address the issues, we propose TFITS, an innovative

The associate editor coordinating the review of this manuscript and approving it for publication was Carlos Thomaz (*Corresponding author: Kangyan Li*).

J. Yuan, Kangyan Li, B. Wu, J. Wan, J. Zhang, Y. Yin, and Y. Zeng are with Hangzhou Dianzi University, Hangzhou, 310018, China (e-mails: yuanjunfeng@hdu.edu.cn, kangyanli@hdu.edu.cn, baofu.wu@hdu.edu.cn, wanjian@hdu.edu.cn, jilin.zhang@hdu.edu.cn, yinyuyu@hdu.edu.cn, and yz@hdu.edu.cn).

J. Yuan, Kangyan Li, B. Wu, Z., Y. Yin, and Y. Zeng are also with Key Laboratory of Complex Systems Modeling and Simulation Ministry of Education, Hangzhou, 310018, China.

B. Wu, Y. Yin, and Y. Zeng are also with Zhoushan Tongbo Marine Electronic Information Research Institute of Hangzhou Dianzi University, Zhoushan, 316104, China.

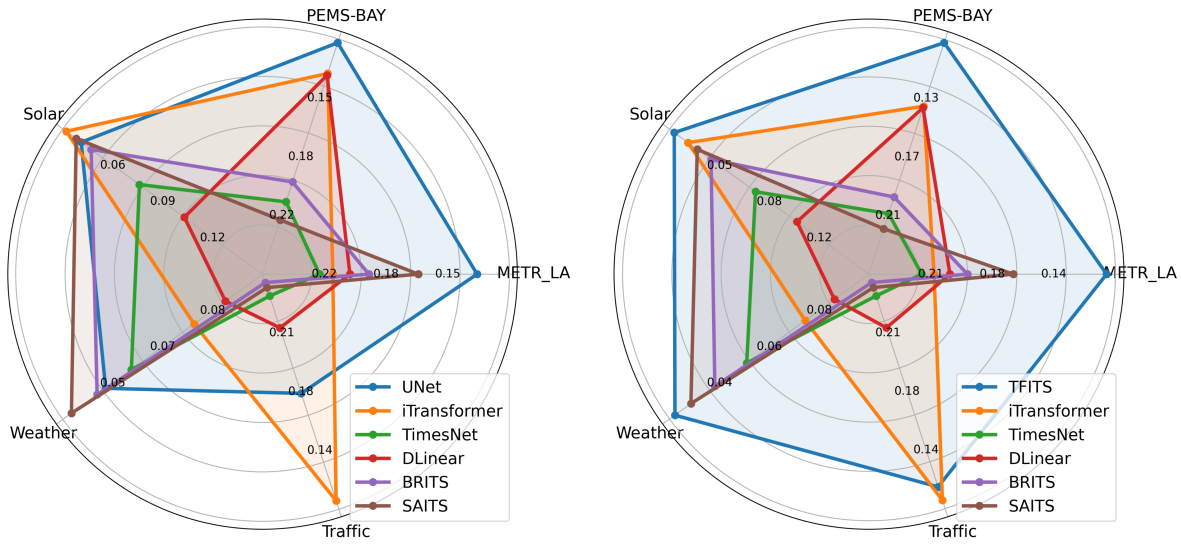


Fig. 1. Compared to other baseline models, using Mean Absolute Error (MAE) as the evaluation metric, the MAE values of the UNet model closer to the outer edge of the graph indicate better performance. From the comparison effect of UNet shown in (a), its overall performance is acceptable, yet it has not achieved the optimal result. However, the overall comparison effect of TFITS presented in (b) indicates that our model has basically achieved the optimal performance.

model that integrates dual perspectives of temporal and feature. TFITS fully exploits the potential synergistic effects between these perspectives, overcoming the limitations of single-view analysis and enabling more comprehensive handling of missing values in multivariate time series. In our research, we observed that applying UNet[24] to this task demonstrates superior imputation performance compared to most existing models (as illustrated in Fig. 1(a)). However, UNet still falls short of achieving optimal performance and exhibits limitations when adapting to time series data, as detailed in the methodology section. Given that UNet serves as a highly effective and universal fusion model in the field of image processing, we designed a universal fusion module, UNetFuse, tailored to the characteristics of time series datasets, to effectively combine feature maps generated from both temporal and feature perspectives. By integrating multi-perspective feature maps, TFITS leverages complementary information from different perspectives to enhance its performance in time series imputation tasks, thereby achieving superior imputation results, as shown in Fig. 1(b).

Building on the above research, the proposed TFITS model can serve as an effective imputation tool. The main contributions of this paper are as follows:

- Starting from the intrinsic nature of time series data, we effectively overcome the inherent limitations of single-perspective analysis by leveraging the dual perspectives of temporal and feature. This enables a more accurate and comprehensive approach to missing value imputation.
- We observed that the UNet model demonstrates good performance in time series imputation tasks and exhibits potential as a universal fusion framework. To adapt to the characteristics of time series data, we designed the UNetFuse module.
- We conducted extensive comparative experiments, and the results indicate that our model not only achieves superior performance but also exhibits strong stability.

II. Related Work

Single-perspective imputation models typically handle time series data from one specific domain, such as the temporal, feature, or frequency domain. Among RNN-based methods, MRNN [12] employs a multi-directional recurrent architecture to gather information from multiple directions for imputation, while BRITS [13] utilizes a bidirectional RNN to model sequences in both forward and backward directions. However, RNN-based approaches are often hindered by vanishing or exploding gradient issues, which limit their capacity to capture long-range dependencies. The Transformer model introduced by Vaswani et al. [25] has not only revolutionized natural language processing but also shown significant potential in time series analysis. Numerous subsequent works have extended the Transformer framework: SAITS [26], for instance, introduces masking into the self-attention weight matrix to prevent each time step from attending to itself. From a feature perspective, iTransformer [23] addresses the performance limitation of standard Transformers in time series forecasting, where they can underperform simple linear models by incorporating transpose operations into the input representation. Meanwhile, building on the two key properties of frequency-domain multilayer perceptrons (MLPs), namely global view and energy compression, the FreTS [27] model demonstrates that MLPs operating in the frequency domain achieve enhanced effectiveness in time series forecasting. SCINet[28], a recursive downsample-convolve-interact architecture, employs multiple convolutional filters in the temporal domain to extract diverse temporal features.

Multi-perspective Imputation Models, such as global-local and temporal-frequency domains, enable a comprehensive consideration of feature information across different scales and dimensions, thereby more effectively capturing the intrinsic

patterns of complex time series data. For example, Time-Transformer[29] and PSR-GALIEN[21] integrate global and local features by Temporal Convolutional Networks(TCNs) and Transformer. PatchTST [30] divides the time series into non-overlapping patches, treating each patch as an individual input unit while preserving both global and local dependencies. TF-Net[31], a temporal-frequency domain feature fusion network for multivariate time series classification, consists of two modules: the temporal domain module and the frequency domain module. It adopts a feature fusion strategy to integrate these two different types of features. Based on the dual perspectives of the temporal and frequency domains, TFDNet[32] effectively integrates temporal and frequency features through multi-scale time-frequency encoding and trend/seasonal component separation mechanisms. He et al. [33] proposed an ARIMA-LSTM hybrid framework wherein linear features extracted from time series data by the ARIMA model are integrated as auxiliary inputs into a multivariate LSTM, thus enabling the capture of complex nonlinear dependencies.

III. Methods

In the field of time series prediction, single-perspective approaches exhibit significant limitations. To overcome these limitations and enhance the accuracy of time series prediction, we propose a novel model, TFITS, which achieves precise analysis and processing of complex data through a unique architectural design. The overall architecture is illustrated in Fig. 2. The model primarily consists of three core components working in synergy: the Feature Perspective Module (FBlock), the Temporal Perspective Module (TBlock), and the Fusion Module (UNetFuse). These components will be described in detail in the following subsections.

A. Problem Definition

In a multivariate time series, the dataset contains time series with multiple variables, which is denoted as $T = \{t_0, t_1, ..., t_{N-1}\} \in R^{N\times D}$. where t_i represents a time point of the time series data, and N also indicates the length of the time series. Each time point is composed of a specified number of features, that is,$t_i = \{x_i^0, x_i^2, ..., x_i^{D-1}\} \in R^D$, where D represents the number of features at this time point. Since some values may be missing, we introduce a mask $M \in \{0,1\}^{N\times D}$, and identify the observed values in the data through Equation:

$$M = \begin{cases} 0, & otherwise \\ 1, & if\ x_i^d\ is\ observed \end{cases} \tag{1}$$

Our goal is to make the predicted values of the model as close as possible to the true values, as shown in Equation 2, where θ represents the model parameters, and ℓ_e is an error function used to measure the gap between the predicted values and the true values. In this paper, we adopt the Mean Absolute Error (MAE) as the loss function, as shown in Equation 3.

$$\mathcal{L}oss(\theta) = \ell_e(T, \mathcal{F}_\theta(T)) \tag{2}$$

$$\text{MAE}(X, Y, M) = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{D-1} |X_{ij} - Y_{ij}| \cdot M_{ij}}{\sum_{i=0}^{N-1}\sum_{j=0}^{D-1} M_{ij}} \tag{3}$$

B. Feature Perspective Module

The Feature Perspective Module (FBlock) draws inspiration from the inversion processing strategy of iTransformer[23], representing time series data from the feature dimension. Specifically, FBlock independently embeds the continuous time series corresponding to each feature variable into a token, where the embedded token reflects the global dynamic changes of that feature. These feature tokens further capture dependencies among features through a self-attention mechanism, thereby more thoroughly exploiting the feature correlations in multivariate time series.

Given the potential divergence in time-varying trends across distinct features, a channel-independent processing methodology is employed in the embedding phase. More precisely, distinct linear mapping layers are implemented for individual features to preserve their intrinsic characteristics, as shown in Equation (4). Here,$W_i \in R^{N\times d_k}$, $b_i \in R^{d_k}$, and d_k represent the dimensions of the mapping space.

$$\text{emb}_f = \left\{ X_{0:N}^i \cdot W_i + b_i \middle| 0 \le i < D \right\} \tag{4}$$

To effectively capture the feature correlations in time series data, the self-attention architecture, with its inherent advantage of global information perception ability, becomes an ideal choice for this task. Multi-head attention(MHA) is a key component introduced in the Transformer architecture that enables the model to simultaneously focus on different aspects of the input sequence by attending to information from multiple representational subspaces. The module adopts linear projections to get $Q, K, V \in R^{H\times N\times d_k}$, where d_k is the dimension of the feature space. and H is the number of attention heads. With denotation of $q_i, k_j \in R^{d_k}$ as the specific query and key of one token, we notice that each entry of the pre-Softmax attention scores is formulated as $A_{i,j}^h = (QK^T/\sqrt{d_k})_{i,j} \propto \frac{q_i^T k_j}{\sqrt{d_k}}$. The attention matrix A reveals the correlation between time points. Consequently, highly correlated time points will be more weighted for the interaction with values V.So the final output is $out = Softmax(\frac{QK^T}{\sqrt{d_k}})V$.

C. Temporal Perspective Module

The Temporal Perspective Module (TBlock) maps the multivariable features corresponding to each time point into a high-dimensional feature space, generating representative vector representations.

To further highlight the data correlations in the temporal dimension, we have innovatively designed an embedding method based on the combination of the frequency domain and the temporal domain. This method utilizes the Discrete Fourier Transform (DFT) and its Inverse Discrete Fourier Transform (IDFT) to incorporate temporal information into the feature vectors in a more compact and efficient manner.Through the

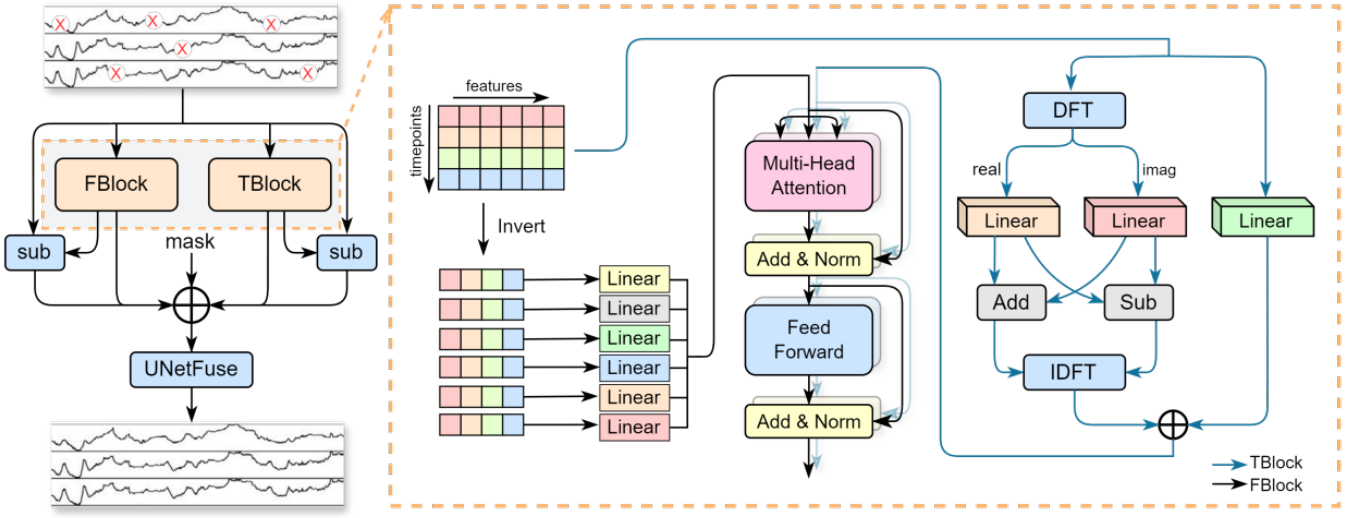


Fig. 2. The TFITS framework consists of three main modules: the Temporal Perspective Module (TBlock), the Feature Perspective Module (FBlock), and the Fusion Module (UNetFuse). The left part corresponds to the overall structure of the model. In it, the TBlock and the FBlock start from the temporal dimension and the feature dimension respectively (inside the orange boxes). The blue lines correspond to the data flow of the temporal perspective, and the black lines correspond to the data flow of the feature perspective. Taking the extracted multi-perspective feature maps as the input, the UNetFuse Module integrates multi-dimensional information through a deep fusion strategy, thereby generating the final prediction result.

DFT, the time series data is transformed into the frequency domain, yielding a series of complex values. These complex values are composed of a real part and an imaginary part, which respectively contain the amplitude and phase information of different frequency components. Here,k represents the k-th frequency domain coefficient, and in practical applications, only the first $N/2$ frequency components are used. This can be expressed as follows:

$$X_k = \sum_{n=0}^{N-1} X_n e^{-i\frac{2\pi kn}{N}} \tag{5}$$

$$X_k = Re(X_k) + j \cdot Im(X_k) \tag{6}$$

$$Re(X_k) = \sum_{n=0}^{N-1} X_n cos\frac{2\pi kn}{N} \tag{7}$$

$$Im(X_k) = -\sum_{n=0}^{N-1} X_n sin\frac{2\pi kn}{N} \tag{8}$$

The core of the frequency domain encoding lies in the learnable update of the real and imaginary parts of the complex values in the frequency domain. The specific update rules follow Equations (9) and (10), which involve multiple linear layers f used for performing linear transformations and dimensional mapping on the frequency domain information. After completing the frequency domain encoding, we use the Inverse Discrete Fourier Transform (IDFT) to transform the encoded frequency domain information back into the temporal domain, obtaining the frequency domain encoding part. In order to avoid losing the continuity and dynamic characteristics of the time series during the encoding process, we obtain the temporal domain encoding result by passing the original time series data through a simple linear layer, and then concatenate it with the frequency domain encoding result to form the final encoding representation, as shown in Equation (11). This fusion method fully combines the advantages of the temporal domain and the frequency domain, enabling the final encoding result to not only retain the local detailed information of the time series but also capture its long-term periodic and trend characteristics.

$$X.real = f_1(Re(X)) - f_2(Im(X)) \tag{9}$$

$$X.imag = f_3(Re(X)) + f_4(Im(X)) \tag{10}$$

$$\text{emb}_t = \text{Linear}(X) \oplus IDFT(X.real, X.imag) \tag{11}$$

Similar to the output of the FBlock module, the encoded result will pass through the self-attention mechanism to obtain a new feature representation. This result performs better in capturing the dependencies between sequence elements and context information.

D. UNetFuse: Dual-Perspective Fusion Module

The UNet architecture, widely recognized for its robust capability in extracting deep features, has achieved significant popularity in the field of image processing and has shown promising potential for feature extraction in time series data. From the dual perspectives of temporal and feature, this architecture can effectively handle multi-feature maps and is well-suited for feature fusion tasks. However, in the processing of time series data, there is a significant difference between the lengths of the feature dimension, which poses challenges to the dimensionality reduction operation. Conventional methods often struggle to continue the dimensionality reduction due to the excessively small feature dimension. To address this issue, we have improved the shape of the convolution kernel and the

feature fusion mechanism, and designed the UNetFuse module to better adapt to the characteristics of time series data, as shown in Fig. 3.

To adapt to the characteristics of time series data, we adjust the convolutional kernel from 3×3 to 3×1. The core rationale behind this adjustment is as follows: time series data exhibit locality in the temporal dimension, so the convolutional kernel retains the size of 3 in the temporal dimension to capture temporal dependencies. In contrast, since different features exhibit correlations but typically lack spatial locality, the kernel size is adjusted to 1 in the feature dimension to avoid unnecessary convolutional operations. Through this improvement, UNetFuse can better adapt to the characteristics of time series data during the dimensionality reduction process, preventing excessive dimensionality reduction caused by the small feature dimension and ensuring smooth dimensionality reduction operations. As shown in Fig. 3(b), during the downsampling and upsampling processes of UNetBase, the temporal dimension of the time series data is gradually reduced or restored through pooling or transposed convolution operations, while the feature dimension size remains unchanged. This design not only effectively captures temporal locality but also preserves feature correlations.

However, this adjustment introduces a new issue: during the convolution process, the 3×1 convolutional kernel can only extract local features in the temporal dimension but fails to effectively capture correlations between feature dimensions. To address this problem, we introduce a feature weight update module between the two convolutional layers of the DoubleConv module, as shown in Fig. 3(a). Specifically, the input data first undergoes max pooling and average pooling separately in the update dimension. The result of max pooling is then subtracted by the result of average pooling to obtain the feature dispersion degree for each channel. A higher dispersion degree indicates that the channel contains more unique information and is assigned a higher weight, while a lower dispersion degree suggests relatively redundant or less important information, resulting in a correspondingly reduced weight. Based on the dispersion degree of features, the input features are dynamically weighted and updated, thereby uncovering deeper correlations between feature dimensions. This design not only effectively captures temporal locality but also preserves feature correlations.

To enrich the feature map information, we introduce the observation mask M as an additional source of feature maps, providing the model with prior knowledge of missing patterns. During the fusion stage, to more reasonably allocate weights to the outputs of the temporal perspective module and the feature perspective module, we propose an adaptive weight allocation mechanism based on reconstruction differences. Specifically, we calculate the reconstruction differences between the outputs of the temporal perspective module X_t and the feature perspective module X_f with the model input X, as shown in Equation (12). This serves as additional guidance information. The difference metric quantifies the deviation of the outputs from the two perspectives relative to the true observed values, thereby automatically allocating weights when fusing the dual-perspective feature maps of temporal and features to clearly distinguish the performance of the two modules. Consequently, the final feature map is constructed as shown in Equation (13).

$$\sigma(X,Y,M)=\left\{\frac{(X_{ij}-Y_{ij})\cdot M_{ij}}{\sum_{i=0}^{N-1}\sum_{j=0}^{D-1}M_{ij}}\middle|0\le i<N,0\le j<D\right\} \tag{12}$$

$$\text{out}=\text{concat}(\sigma(X,X_t,M),\sigma(X,X_f,M),X_t,X_f,M) \tag{13}$$

IV. Experiment

To validate the generalizability of the proposed TFITS model, we conducted comprehensive evaluations on multiple time series imputation tasks. Additionally, through ablation experiments, we thoroughly investigated the importance of each component of TFITS. Furthermore, to intuitively demonstrate the predictive advantages of TFITS, we employed visualization techniques to present its performance. The best-performing results are highlighted in bold, while the second-best results are in italic.

A. Experimental Setup

Datasets: We conducted experiments on seven real-world datasets to evaluate the performance of the proposed TFITS. Table I provides detailed information on the processing of these datasets in the imputation experiments. The datasets are as follows:

- **METR_LA**: This dataset includes traffic speed information from March 1, 2012, to June 30, 2012, covering four months. It records traffic data collected by 207 sensors across Los Angeles County.
- **PEMS-BAY**: This dataset spans from January 1, 2017, to June 30, 2018, covering six months of vehicle detection reports. It contains real-time traffic information from over 325 sensors in the Bay Area.
- **Solar**: This dataset records the solar power production of 137 photovoltaic plants in 2006, with samples taken every 10 minutes.
- **Traffic**: This dataset collects hourly road occupancy rates measured by 862 sensors on highways in the San Francisco Bay Area from January 2015 to December 2016.
- **AirQuality**: The data was collected from March 1, 2013, to February 28, 2017 (48 months in total), covering hourly air pollutant data from 12 monitoring sites in Beijing. For each monitoring site, 11 continuous time series variables were measured. We aggregated the variables from the 12 sites, resulting in a dataset with 132 features.
- **Exchange**: This dataset collects daily exchange rate data from eight countries between 1990 and 2016.
- **Weather**: This dataset includes 21 meteorological factors collected every 10 minutes from a weather station at the Max Planck Institute for Biogeochemistry in 2020.

Evaluation Metrics: The evaluation metrics include Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), with their specific formulas shown in Equations (3) and (14).

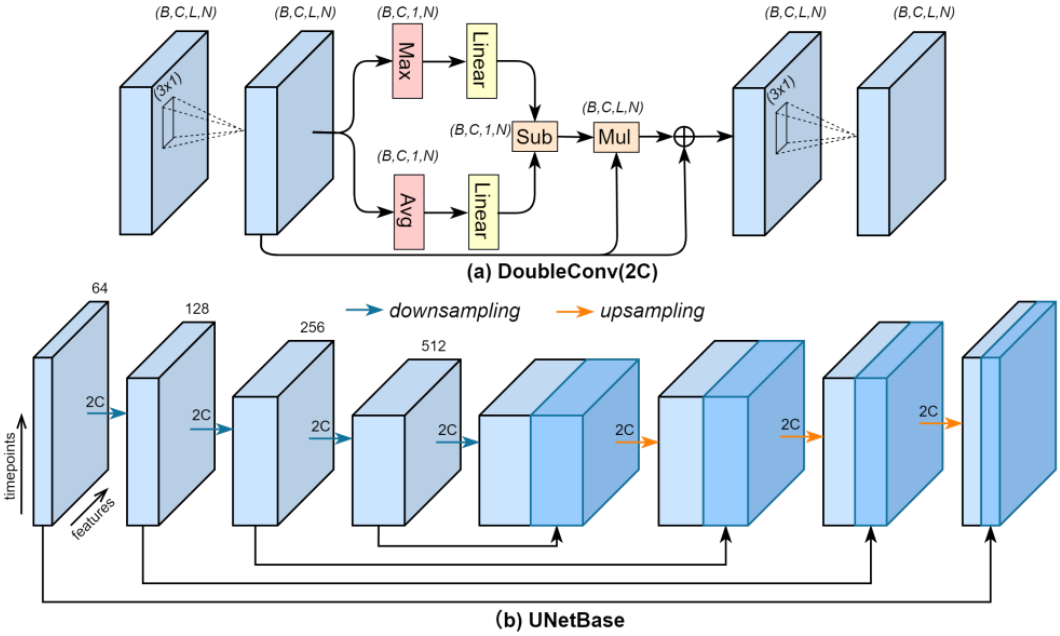


Fig. 3. Framework of the UNetFuse Module. The module consists of two main components: UNetBase and DoubleConv. UNetBase adopts the UNet architecture as its foundation, comprising two symmetric processes: downsampling and upsampling. During the downsampling phase, high-level features are progressively extracted, and spatial resolution is reduced through convolutional operations. In the upsampling phase, spatial resolution is gradually restored, and multi-level features are fused through deconvolutional operations. Before each upsampling or downsampling step, the DoubleConv module is applied. This module extracts local temporal correlations via (3×1) convolutional operations and incorporates a feature weight update strategy to explore feature dependencies.

TABLE I
DETAILED INFORMATION OF IMPUTATION EXPERIMENTS ON SEVEN PUBLIC DATASETS

	METR_LA	PEMS-BAY	Traffic	Solar	AirQuality	Weather	Exchange
length	288	288	24	24	24	144	24
features	207	325	862	137	132	21	8

Lower values of MAE and RMSE indicate predictions that are closer to the true values, reflecting higher accuracy.

$$\text{RMSE}(X, Y, M) = \sqrt{\frac{\sum_{i=0}^{N-1}\sum_{j=0}^{D-1}(X_{ij} - Y_{ij})^2 \cdot M_{ij}}{\sum_{i=0}^{N-1}\sum_{j=0}^{D-1} M_{ij}}} \tag{14}$$

Baseline: We have carefully selected six widely recognized imputation models as benchmarks, including: (1) Models based on the Transformer architecture: SAITS[26] and iTransformer[23]; (2) Model based on the linear network: DLinear[34]; (3) Models based on the Recurrent Neural Network (RNN): MRNN[12] and BRITS[13]; (4) Model based on convolution: TimesNet[35]; (6) Model based on the fusion of the temporal and frequency domains: TFDNet[32].

B. Comparative Experiments

To validate the reliability of TFITS in predicting missing values in multivariate time series, this study conducted comparative experiments. The experiments were divided into two main parts: first, the overall performance of the model was validated by comparing its missing value prediction results with those of various baseline models. Second, the model's performance under different missing rates was specifically analyzed to assess its adaptability.

1) Baseline Comparison Experiments: Table II presents the overall performance of our proposed model, TFITS, compared to baseline models across seven public datasets. From the overall experimental results, the TFITS model demonstrates strong competitiveness. TFITS exhibits excellent performance on the majority of datasets, with the exception of the Traffic dataset, where its performance did not meet expectations.

Specifically, on the remaining six public datasets excluding Traffic, TFITS outperforms all other baseline models. Notably, on the Exchange dataset, TFITS achieves the most significant performance improvements over the second-best model, with enhancements of up to 38.2% and 40.7% in MAE and RMSE, respectively. On the AirQuality dataset, although the performance improvement is the smallest, TFITS still achieves gains of 2.3% and 0.3% in MAE and RMSE, respectively. On average, TFITS improves performance by 24.8% and 22.3% in MAE and RMSE, respectively. Although TFITS underperforms on the Traffic dataset, experiments in the next subsection will demonstrate that it still plays a non-negligible role on this dataset.

2) Sensitivity Experiment to Missing Rate: To further verify the reliability of the model under different missing rates, we set the missing rates at 20%, 40%, 60%, and 80% respectively on the test dataset. The experimental results show that TFITS

TABLE II
COMPARISON RESULTS OF DIFFERENT MODELS ON SEVEN PUBLIC DATASETS

Method	METR_LA		PEMSBAY		Traffic		Solar		AirQuality		Weather		Exchange	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SAITS	*0.168*	*0.349*	0.220	0.461	0.242	0.654	0.038	0.088	0.106	0.377	*0.033*	0.166	0.210	0.247
MRNN	0.297	0.536	0.350	0.638	0.309	0.697	0.094	0.184	0.241	0.425	0.145	0.310	0.874	1.140
BRITS	0.194	0.402	0.199	0.413	0.245	0.635	0.047	0.099	0.106	0.382	0.042	*0.163*	0.235	0.321
DLinear	0.204	0.372	0.140	0.279	0.218	0.547	0.103	0.156	0.177	0.432	0.087	0.196	0.113	0.157
iTransformer	0.213	0.385	*0.139*	*0.268*	**0.115**	**0.352**	*0.032*	*0.084*	*0.088*	*0.233*	0.076	0.196	*0.034*	*0.054*
TimesNet	0.220	0.406	0.210	0.434	0.237	0.678	0.076	0.141	0.146	0.380	0.054	0.185	0.070	0.100
TFDNet	0.235	0.418	0.154	0.304	0.239	0.528	0.066	0.128	0.182	0.437	0.066	0.200	0.050	0.074
TFITS	**0.115**	**0.224**	**0.097**	**0.198**	*0.123*	*0.400*	**0.023**	**0.066**	**0.085**	**0.210**	**0.027**	**0.148**	**0.021**	**0.032**

performs excellently under various missing rates. As shown in Fig. 4, compared with the SAITS model, TFITS improves the average performance by 37.9% on the Traffic dataset, by 48.1% on the METR_LA dataset, and by 61.7% on the Solar dataset. When compared with the iTransformer model, although TFITS does not perform well under a 20% missing rate, it significantly outperforms iTransformer when the missing rate is 40% or higher. The overall average performance is improved by 7.4%. Specifically, on the METR_LA dataset, the average performance is improved by 53.7%, and on the Solar dataset, the average performance is improved by 48.1%. In addition, the error of TFITS shows a stable increasing trend under different missing rates, without the explosive growth or prediction collapse phenomena commonly seen in other models. Specifically, on the Solar dataset, the average increase in error is only 0.011; on the Traffic dataset, it is 0.038; and on the METR_LA dataset, it is 0.026. In conclusion, TFITS not only maintains the optimal prediction performance under various missing rates but also demonstrates significant advantages in terms of stability.

C. Ablation Experiments

To comprehensively and thoroughly validate the effectiveness of each module in TFITS, ablation experiments were conducted, which are primarily divided into two parts. First, the importance of the multi-perspective module was investigated by adding or removing feature maps from specific perspectives to observe the resulting effects. Second, the improvements made to the UNetFuse module based on UNet were rigorously evaluated to determine whether they contribute positively, specifically examining whether the enhanced fusion module demonstrates superior performance.

1) Multi-perspective Ablation Experiments: By removing the temporal or perspective module, we aim to verify that the results of the multi-perspective module are superior to those of the single-perspective module. The results are shown in Table III. TBlock represents the temporal perspective module, and FBlock represents the feature perspective module. The first row of Table III indicates the removal of both the temporal perspective and the feature perspective. When these two perspectives are removed, the model cannot operate properly due to the lack of feature map inputs. Therefore, we directly input the original input values as a feature map into the model. The results of the ablation experiments show that without the feature maps generated by both the temporal and feature perspectives, we observe that the average performance of MAE and RMSE across the entire benchmark decreases by 22.2% and 18.9%, respectively. Without the temporal perspective, MAE and RMSE decrease by 15.4% and 23.9%, respectively. Without the feature perspective, MAE and RMSE decrease by 11.5% and 12%, respectively. In conclusion, as can be seen from the results of the ablation experiments, the temporal perspective module and the feature perspective module in TFITS have a significant impact on the performance of the model. Their absence will lead to varying degrees of decline in performance indicators such as MAE and RMSE.

2) Ablation Experiments of the Fusion Module: To verify the effectiveness of UNetFuse as a fusion module, we respectively used UNet, UNet++[36], and ECANet[37] as the fusion modules to conduct fusion experiments on multi-perspective feature maps. The experimental results are shown in Table IV. The experiments show that UNetFuse performs optimally in the fusion task. Moreover, due to the fact that the Exchange dataset only has eight features, the UNet-series models are unable to complete the downsampling process, while UNetFuse has no such limitation. Compared with the second-best model, UNetFuse improves the MAE and RMSE indicators by 6.7% and 3.7%, respectively. In contrast, although UNet++ is also an improvement of UNet, its complex structure does not bring performance gains in the current scenario, and its performance is far inferior to that of UNetFuse. ECANet, which is based on channel attention, also fails to achieve the same effect as UNetFuse. In conclusion, designed based on UNet, UNetFuse not only overcomes the inherent limitations of UNet but also significantly improves the performance, demonstrating its superiority and effectiveness in the task of multi-perspective feature fusion.

D. Robustness Experiments

All experiments are implemented in PyTorch[38] and conducted on a single NVIDIA A6000 48GB GPU. To systematically evaluate the robustness of TFITS, we analyze the sensitivity of model performance from two complementary perspectives: hyperparameter perturbation and random seed variation. This design aims to examine whether the reported performance is stable under common sources of randomness in model training.

We first conduct a random sampling experiment on several core hyperparameters, where parameter combinations are randomly selected within predefined ranges: learning rate

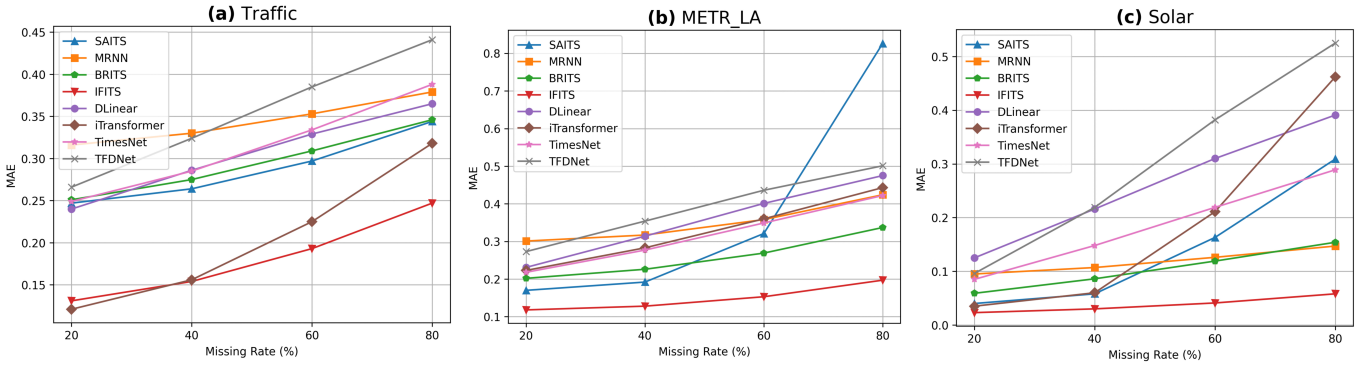


Fig. 4. Visualization effects of three datasets under different missing rates (20%, 40%, 60%, 80%) are presented. Specifically, (a) shows the visualization results of the Traffic dataset, (b) represents the visualization results of the METR_LA dataset, and (c) displays the visualization results of the Solar dataset.

TABLE III
ABLATION EXPERIMENTS OF TEMPORAL AND FEATURE PERSPECTIVES

TBlock	FBlock	METR_LA		PEMS-BAY		Traffic		Solar		AirQuality		Weather		Exchange	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
×	×	*0.119*	*0.237*	*0.099*	*0.202*	0.152	*0.410*	0.032	0.078	0.113	0.379	0.039	0.159	*0.023*	*0.037*
×	√	0.133	0.277	0.100	0.210	*0.129*	0.415	*0.026*	*0.068*	0.119	0.383	0.029	0.155	0.026	0.046
√	×	0.124	0.247	0.101	0.224	0.145	0.423	0.032	0.078	*0.089*	*0.224*	*0.028*	*0.151*	0.024	0.041
√	√	**0.115**	**0.224**	**0.097**	**0.198**	**0.123**	**0.400**	**0.023**	**0.066**	**0.085**	**0.210**	**0.027**	**0.148**	**0.021**	**0.032**

TABLE IV
ABLATION EXPERIMENTS OF THE UNETFUSE FUSION MODULE

Methods	METR_LA		PEMS-BAY		Traffic		Solar		AirQuality		Weather		Exchange	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UNet	*0.124*	*0.234*	*0.099*	*0.199*	*0.138*	*0.437*	*0.027*	*0.071*	*0.087*	*0.213*	*0.028*	*0.149*	-	-
UNet++	0.426	0.787	0.372	0.929	0.528	0.921	0.584	0.689	0.555	0.850	0.140	0.275	-	-
ECANet	0.239	0.421	0.183	0.368	0.144	0.438	0.064	0.137	0.123	0.325	0.068	0.174	*0.035*	*0.052*
UNetFuse	**0.115**	**0.224**	**0.097**	**0.198**	**0.123**	**0.400**	**0.023**	**0.066**	**0.085**	**0.210**	**0.027**	**0.148**	**0.021**	**0.032**

$lr \in [0.0005, 0.0008]$, network layers $layer \in \{1, 2, 3\}$, and model dimension $d_{\text{model}} \in \{256, 512\}$. These ranges cover typical configurations used in practice while avoiding extreme or degenerate settings. The corresponding results are illustrated in Fig 5(a). Across the Exchange, Weather, Solar, and AirQuality datasets, TFITS exhibits relatively small performance variations, with MAE standard deviations consistently below 0.0015, indicating that the model is not overly sensitive to reasonable changes in hyperparameter configurations.

To further assess the impact of randomness introduced by model initialization and training stochasticity, we fix the hyperparameter configuration and vary only the random seed during training, with ($seed \in \{399, 400, 401, 402, 403\}$). The results are shown in Fig 5(b). We adopt five random seeds, which is consistent with robustness evaluation practices commonly used in recent time-series imputation and forecasting studies. Under different random seeds, TFITS demonstrates highly stable performance, with MAE standard deviations remaining below 0.0001 across all evaluated datasets. This suggests that random initialization introduces only negligible fluctuations within a reasonable range.

E. Visualization Results

In real-world scenarios, issues such as equipment failures can lead to data missing, and this kind of missing data often exhibits temporal continuity. To more realistically simulate this actual situation, we conducted an experiment on six datasets. Specifically, we set a continuous time period of a certain feature in the dataset samples to a missing state. The length of this continuous time period is one-third of the total length and is located in the middle position. Subsequently, we input the data samples with continuous missing values into three models, namely iTransformer, SAITS, and TFITS, respectively, to perform the task of predicting continuous missing values. Fig. 6 shows the visualization results of the models' predictions, and the MAE evaluation results compared with the true values are provided in the lower right corner. Through careful observation and comparison of the visualization results, it can be found that the predicted values of our proposed TFITS model are closer to the true values compared with those of other models. This indicates that the TFITS model also demonstrates excellent performance in handling continuous missing values.

In addition to qualitative visualization of imputed values, Fig 5 provides complementary visual evidence regarding the robustness of TFITS. As shown in Fig 5(a), the MAE curves corresponding to different hyperparameter configurations exhibit limited fluctuations across all datasets. The relatively narrow error bars indicate that the performance variations remain small even when the model is trained under different

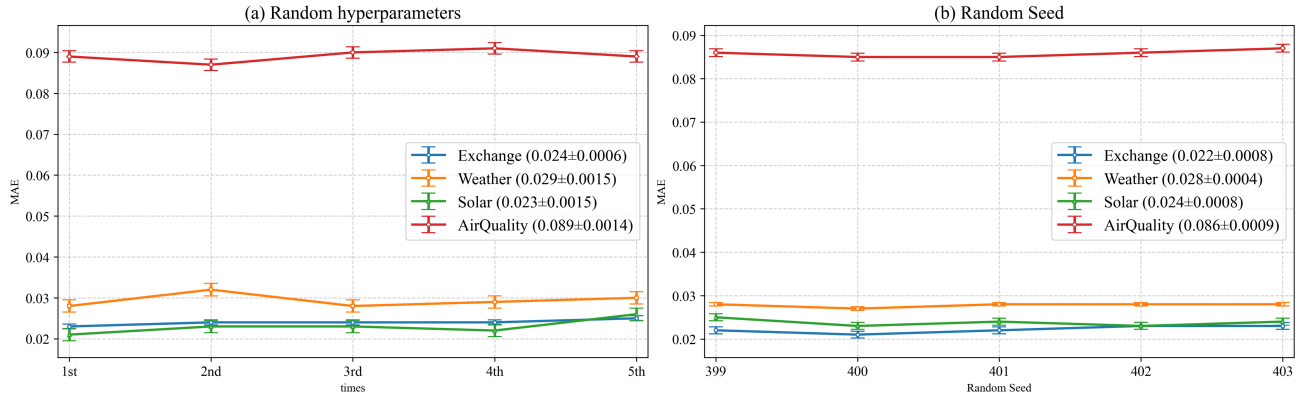


Fig. 5. Model robustness visualization. (a) Hyperparameter randomization, $lr \in [0.0005, 0.0008]$, $layer \in \{1, 2, 3\}$, $d_{\text{model}} \in \{256, 512\}$; (b) Random seed variation,$seed \in \{399, 400, 401, 402, 403\}$. Error bars denote mean MAE ± standard deviation.

reasonable hyperparameter settings. This observation suggests that the predictive performance of TFITS is not highly sensitive to specific hyperparameter choices.

Fig 5(b) further illustrates the effect of random seed variation. Despite changes in random initialization, the MAE curves remain stable and closely clustered, with very small standard deviations across datasets. This visual pattern is consistent with the quantitative results reported in the robustness experiments and indicates that the overall performance trends are preserved under different random seeds.

Taken together, the robustness visualizations suggest that the performance gains reported for TFITS are stable and reproducible, and are not driven by particular random initializations or isolated hyperparameter configurations.

V. Conclusion

In the field of time series imputation, single-perspective methods struggle to comprehensively capture the intrinsic features and changing patterns of data. This limitation leads to insufficient performance and stability when dealing with the task of imputing missing values. To address this issue, we propose the TFITS model. This model innovatively approaches the problem from both temporal and feature perspectives, and designs the UNetFuse fusion module. It brings new ideas and methods to time series imputation, effectively overcoming the limitations of single-perspective approaches. As a result, it enables more accurate imputation of missing values in time series.To evaluate the performance of the TFITS model, we focus on the metrics of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), conducting a comprehensive and systematic assessment. We also compare it with various well-known baseline methods in the industry. The results show that, in most experimental scenarios, the TFITS model significantly outperforms existing baseline methods in terms of performance. Specifically, in terms of the MAE metric, the TFITS model has an average improvement of 20.3%, and in terms of the RMSE metric, it has an average improvement of 17.2%, which fully demonstrates its excellent performance in the task of missing value imputation. In the future, the TFITS model can be combined with more feature maps to further expand its analysis dimensions and explore its application potential in more complex scenarios.

Acknowledgments

Thanks to all who have contributed to the paper. This research was supported by the National Natural Science Foundation of China (NSFC) under Grant No.62302133; the Key Research and Development Program of Zhejiang Province under Grand No. 2024C01104; the Yangtze River Delta Project under Grant No.2023ZY1068; the Zhejiang Province High-Level Talent Special Support Program-Leading Talent of Technological Innovation under No. 2022R52043.

References

[1] H. Xu, Z. Liu, H. Wang, C. Li, Y. Niu, W. Wang, and X. Liu, "Denoising diffusion straightforward models for energy conversion monitoring data imputation," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 10, pp. 11 987–11 997, 2024, doi:10.1109/TII.2024.3413349.

[2] C. Fu, M. Quintana, Z. Nagy, and C. Miller, "Filling time-series gaps using image techniques: Multidimensional context autoencoder approach for building energy data imputation," *Applied Thermal Engineering*, vol. 236, p. 121545, 2024, doi:https://doi.org/10.1016/j.applthermaleng.2023.121545.

[3] F. Chen, L. Yu, J. Mao, Q. Yang, D. Wang, and C. Yu, "A novel data-characteristic-driven modeling approach for imputing missing value in industrial statistics: A case study of china electricity statistics," *Applied Energy*, vol. 373, p. 123854, 2024, doi:https://doi.org/10.1016/j.apenergy.2024.123854.

[4] O. Duarte, J. E. Duarte, and J. Rosero-Garcia, "Data imputation in electricity consumption profiles through shape modeling with autoencoders," *Mathematics*, vol. 12, no. 19, 2024, doi:10.3390/math12193004.

[5] M.-C. Cheng, Y.-H. Hsieh, T.-C. Hsu, T.-H. Su, and C. Lin, "Deep sti: Deep stochastic time-series imputation on electronic health records," in *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2024, pp. 1–4, doi:10.1109/EMBC53108.2024.10782239.

[6] M. Kazijevs and M. D. Samad, "Deep imputation of missing values in time series health data: A review with benchmarking," *Journal of Biomedical Informatics*, vol. 144, p. 104440, 2023, doi:https://doi.org/10.1016/j.jbi.2023.104440.

[7] T. Decorte, S. Mortier, J. J. Lembrechts, F. J. R. Meysman, S. Latré, E. Mannens, and T. Verdonck, "Missing value imputation of wireless sensor data for environmental monitoring," *Sensors*, vol. 24, no. 8, 2024, doi:10.3390/s24082416.

[8] X. Chen, Z. Cheng, H. Cai, N. Saunier, and L. Sun, "Laplacian convolutional representation for traffic time series imputation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6490–6502, 2024, doi:10.1109/TKDE.2024.3419698.

[9] J. Wang, W. Du, Y. Yang, L. Qian, W. Cao, K. Zhang, W. Wang, Y. Liang, and Q. Wen, "Deep learning for multivariate time series imputation: A survey," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, J. Kwok, Ed. International Joint Conferences on Artificial Intelligence

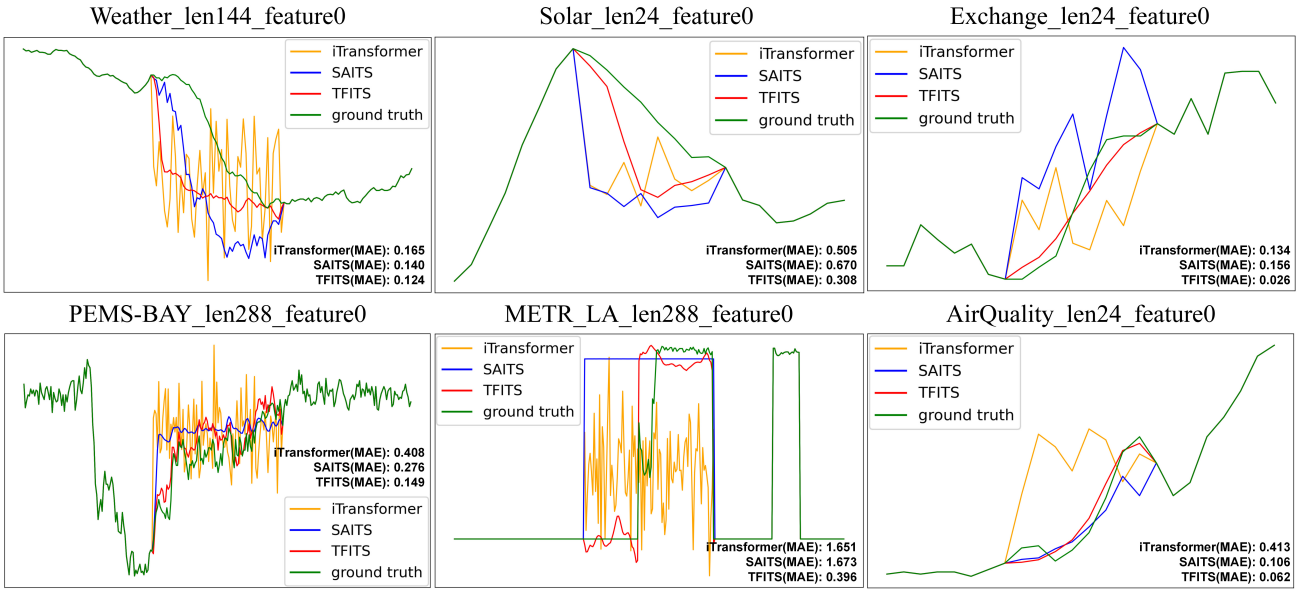


Fig. 6. Visualization results of the missing values in continuous time periods in six public datasets

Organization, 8 2025, pp. 10 696–10 704, survey Track. [Online]. Available: https://doi.org/10.24963/ijcai.2025/1187

[10] X. Jia, X. Dong, M. Chen, and X. Yu, "Missing data imputation for traffic congestion data based on joint matrix factorization," *Knowledge-Based Systems*, vol. 225, p. 107114, 2021, doi:https://doi.org/10.1016/j.knosys.2021.107114.

[11] H. Liu and L. Li, "Missing data imputation in gnss monitoring time series using temporal and spatial hankel matrix factorization," *Remote Sensing*, vol. 14, no. 6, 2022, doi:10.3390/rs14061500.

[12] 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, doi:10.1109/TBME.2018.2874712.

[13] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018, doi:https://doi.org/10.48550/arXiv.1805.10572.

[14] S. Pachal and A. Achar, "Sequence prediction under missing data: An rnn approach without imputation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, ser. CIKM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 1605–1614, doi:10.1145/3511808.3557449.

[15] A. Y. Yıldız, E. Koç, and A. Koç, "Multivariate time series imputation with transformers," *IEEE Signal Processing Letters*, vol. 29, pp. 2517–2521, 2022, doi:10.1109/LSP.2022.3224880.

[16] C. Liu, Z. Zhu, W. Hao, and G. Sun, "Heterogeneous multivariate time series imputation by transformer model with missing position encoding," *Expert Systems with Applications*, vol. 271, p. 126435, 2025, doi:https://doi.org/10.1016/j.eswa.2025.126435.

[17] Y. Zhang and P. J. Thorburn, "A dual-head attention model for time series data imputation," *Computers and Electronics in Agriculture*, vol. 189, p. 106377, 2021, doi:https://doi.org/10.1016/j.compag.2021.106377.

[18] Y. Wang, X. Xu, L. Hu, J. Fan, and M. Han, "A time series continuous missing values imputation method based on generative adversarial networks," *Knowledge-Based Systems*, vol. 283, p. 111215, 2024, doi:https://doi.org/10.1016/j.knosys.2023.111215.

[19] R. Qin and Y. Wang, "Imputegan: Generative adversarial network for multivariate time series imputation," *Entropy*, vol. 25, no. 1, 2023, doi:10.3390/e25010137.

[20] Y. Zhang, B. Zhou, X. Cai, W. Guo, X. Ding, and X. Yuan, "Missing value imputation in multivariate time series with end-to-end generative adversarial networks," *Information Sciences*, vol. 551, pp. 67–82, 2021, doi:https://doi.org/10.1016/j.ins.2020.11.035.

[21] Z. Tang, T. Ji, J. Kang, Y. Huang, and W. Tang, "Learning global and local features of power load series through transformer and 2d-cnn: An image-based multi-step forecasting approach incorporating phase space reconstruction," *Applied Energy*, vol. 378, p. 124786, 2025, doi:https://doi.org/10.1016/j.apenergy.2024.124786.

[22] Y. Chen, S. Liu, J. Yang, H. Jing, W. Zhao, and G. Yang, "A joint time-frequency domain transformer for multivariate time series forecasting," *Neural Networks*, vol. 176, p. 106334, 2024, doi:https://doi.org/10.1016/j.neunet.2024.106334.

[23] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," 2024, doi:https://doi.org/10.48550/arXiv.2310.06625.

[24] W. Weng and X. Zhu, "Inet: Convolutional networks for biomedical image segmentation," *IEEE Access*, vol. 9, pp. 16 591–16 603, 2021, doi:10.1109/ACCESS.2021.3053408.

[25] V. Ashish, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, p. I, 2017, doi:https://doi.org/10.48550/arXiv.1706.03762.

[26] W. Du, D. Côté, and Y. Liu, "Saits: Self-attention-based imputation for time series," *Expert Systems with Applications*, vol. 219, p. 119619, 2023, doi:https://doi.org/10.1016/j.eswa.2023.119619.

[27] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu, "Frequency-domain mlps are more effective learners in time series forecasting," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 76 656–76 679. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/f1d16af76939f476b5f040fd1398c0a3-Paper-Conference.pdf

[28] M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "Scinet: Time series modeling and forecasting with sample convolution and interaction," 2022, doi:https://doi.org/10.48550/arXiv.2106.09305.

[29] Y. Liu, S. Wijewickrema, A. Li, C. Bester, S. O'Leary, and J. Bailey, "Time-transformer: Integrating local and global features for better time series generation," in *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 2024, pp. 325–333, doi:https://doi.org/10.48550/arXiv.2312.11714.

[30] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022, doi:https://doi.org/10.48550/arXiv.2211.14730.

[31] T. Lei, J. Li, and K. Yang, "Time and frequency-domain feature fusion network for multivariate time series classification," *Expert Systems with Applications*, vol. 252, p. 124155, 2024, doi:https://doi.org/10.1016/j.eswa.2024.124155.

[32] Y. Luo, S. Zhang, Z. Lyu, and Y. Hu, "Tfdnet: Time–frequency enhanced decomposed network for long-term time series forecasting," *Pattern Recognition*, vol. 162, p. 111412, 2025, doi:https://doi.org/10.1016/j.patcog.2025.111412.

[33] C. He, "A hybrid model based on multi-lstm and arima for time

series forcasting," in *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2023, pp. 612–616, doi:https://doi.org/10.1109/ICSP58490.2023.10248909.

[34] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128, doi:10.1609/aaai.v37i9.26317.

[35] Y. Wang, K. Yi, X. Liu, Y. G. Wang, and S. Jin, "ACMP: Allen-cahn message passing with attractive and repulsive forces for graph neural networks," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=4fZc_79Lrqs

[36] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds. Cham: Springer International Publishing, 2018, pp. 3–11. [Online]. Available: https://doi.org/10.1007/978-3-030-00889-5_1

[37] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, doi:10.1109/CVPR42600.2020.01155.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

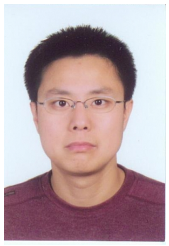
Junfeng Yuan is pursuing his Ph.D. degree in computer science at the School of Computer Science and Technology, Hangzhou Dianzi University, China. He specializes in deep learning, parallel computing, and data mining. His work focuses on promoting the effective utilization of data, particularly in the application and research of time series data.

Kangyan Li is a graduate student at the School of Computer Science and Technology, Hangzhou Dianzi University, China. He specializes in deep learning and data processing.

Baofu Wu is pursuing his Ph.D. degree in computer science at the School of Computer Science and Technology, Hangzhou Dianzi University, China. He specializes in edge computing, parallel computing, and Cooperative Vehicle Infrastructure Systems(CVIS).

Jian Wan received the Ph.D. degree in Computer Application Technology from Zhejiang University, Zhejiang, China, in 1989. His research interests include grid computing, service computing, and cloud computing. He is currently a Professor in software engineering with the School of Computer Science and Technology, Hangzhou Dianzi University, and the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education.

Jilin Zhang received the Ph.D. degree in Computer Application Technology from University of Science Technology Beijing, Beijing, China, in 2009. His research interests include high-performance computing and cloud computing. He is currently an Associate Professor with the Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, School of Computer Science and Technology, Hangzhou Dianzi University.



Yuyu Yin received the Ph.D. degree in computer science from Zhejiang University, in 2010. He is currently a Professor with the College of Computer, Hangzhou Dianzi University. He is also a Supervisor of master's students with the School of Computer Engineering and Science, Shanghai University, Shanghai, China. He has published more than 40 articles in journals and refereed conferences, such as Sensors, Entropy, IJSEKE, Mobile Information Systems, ICWS, and SEKE. His research interests include service computing, cloud computing, and business process management. He is also a member of the China Computer Federation (CCF) and the CCF Service Computing Technical Committee. He has organized more than ten international conferences and workshops, such as FMSC 2011–2017 and DISA 2012 and 2017–2018. He has served as a Guest Editor for the Journal of Information Science and Engineering and International Journal of Software Engineering and Knowledge Engineering and a Reviewer for the IEEE Transaction on Industry Informatics, the Journal of Database Management, and Future Generation Computer Systems.

Yan Zeng PhD, is currently an Associate Professor in the Computer & Software School at Hangzhou Dianzi University. In 2016, she received her Ph.D. degree from the Institute of Software, Chinese Academy of Sciences. Her research interests include distributed and parallel computing, distributed machine learning, and big data.