SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2026, Vol. 69, Iss. 2, 122303:1–122303:17
https://doi.org/10.1007/s11432-024-4619-y

RFDR: a retransmission-free data reconstruction framework for emergency response networks

Yayong SHI[1], Weidang LU[2], Nan ZHAO[3], Haiyan ZHU[4], Yuan YAO[4], Rui WANG[1*] & Yuan GAO[5,6*]

[1]*School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100853, China*
[2]*College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China*
[3]*School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China*
[4]*First Medical Center, Chinese PLA General Hospital, Beijing 100036, China*
[5]*Academy of Military Science of the PLA, Beijing 100084, China*
[6]*The PLA Information Support Force, Beijing 100011, China*

Received 31 October 2024/Revised 29 May 2025/Accepted 29 September 2025/Published online 13 January 2026

Abstract In emergency rescue scenarios, mobile ad hoc networks (MANETs) with dynamic topology often experience severe data missing. This compromises the reliability of observed data, which is critical for supporting real-time decision-making. Sparse inference methods exhibit significant degradation in reconstruction accuracy under high data missing rates, while retransmission-based methods introduce cascading delays and may even trigger network avalanche effects. To overcome the limitations of both methods, we propose a retransmission-free data reconstruction framework (RFDR). The framework consists of two modules: (i) a soft-sliced dynamic time warping (SS-DTW) module that identifies the most similar offline reference sample for missing value imputation; and (ii) a time-series interpolation module that captures richer contextual correlations to refine the imputation, using the reference sample and the observed sparse data as an initial estimate. To validate the effectiveness of the framework, we developed a hardware-in-the-loop (HIL) testbed that emulates dynamic wireless channel conditions. Experimental results demonstrate that RFDR achieves a 17.3% gain in transmission efficiency and a 63% reduction in median latency over standard retransmission protocols. Compared to baselines, RFDR exhibits consistently superior reconstruction accuracy across simulated random data missing rates ranging from 35% to 85%. These attributes underscore its resilience under extreme data missing conditions and indicate promising applicability in emergency settings that demand high reliability and low-latency communication.

Keywords end-edge collaboration, multivariate time series, transmission outage, MANETs, disaster response

Citation Shi Y Y, Lu W D, Zhao N, et al. RFDR: a retransmission-free data reconstruction framework for emergency response networks. Sci China Inf Sci, 2026, 69(2): 122303, https://doi.org/10.1007/s11432-024-4619-y

1 Introduction

In emergency rescue scenarios, there is a critical need for the rapid deployment of temporary communication infrastructure to restore essential communication services [1–5]. Mobile ad hoc networks (MANETs) serve as a flexible infrastructure to support short-term applications (including environmental sensing, medical monitoring, personnel localization, and traffic management) by interconnecting various Internet of Things (IoT) devices [6]. For instance, injury assessment can be conducted through an end-edge collaborative architecture [7]. End devices incorporate biomedical sensors within wearable or implantable medical systems, which collect vital signs including electrocardiogram, respiratory rate, body temperature, heart rate, and blood pressure [8]. The collected data are transmitted to edge devices, such as mobile command vehicles, portable computing devices, or other edge devices—for real-time monitoring and analysis. Nevertheless, degraded communication links in disaster areas frequently lead to sparse samples, primarily for two reasons.

• Variable missing: Complete failure or damage of a sensor results in a variable-dimension mismatch between the online measurements and the offline repository sample.

• Point-level values missing: Intermittent packet loss and various sampling rates cause irregular data sparsity, which degrades the accuracy of downstream inference tasks.

* Corresponding author (email: wangrui@ustb.edu.cn, yuangao08@tsinghua.edu.cn)

The performance of most computational and perception models is highly correlated with data sparsity and often deteriorates with increased sparsity, as they are typically trained on full-dimensional feature vectors [9]. A common solution is retransmission, which triggers the sender to resend data upon detecting loss or corruption, continuing until an acknowledgment is received. While studies in routing, network deployment, and data slicing have sought to optimize this process, their improvements are often insufficient for real-time control applications. For these systems, delays exceeding strict millisecond-scale thresholds can cause catastrophic system failure [10]. For example, in telemedicine, latency can render surgical commands useless and dangerous; in disaster monitoring, network congestion causes irreplaceable alerts (e.g., for earthquakes) to be lost, forfeiting intervention windows; in industrial IoT, missing sensor data directly reduces control accuracy and product quality [7]. The interpolation of missing data often relies on the low-rank property of the underlying data, a property exploited by established techniques such as low-rank matrix and tensor completion [11]. Recently, neural network-based models for multivariate time-series interpolation have emerged as a prominent class of methods. These models integrate global inter-variable correlations with local temporal statistics to estimate missing values [12]. A key limitation, however, lies in their heavy reliance on short-term temporal patterns, which hinders their performance under long-range missingness and renders them ineffective when an entire variable is absent.

To address the challenge of unusable sparse samples caused by data missing in low-communication-computation environments, this paper begins by identifying two distinct states of data missing and systematically analyzes the limitations of existing methods. Then, focusing on the distribution differences of data features in the target scenario, we propose a retransmission-free data reconstruction framework (RFDR). The framework consists of two modules, each designed to tackle one of the two missing states with targeted strategies, thereby achieving low-error data reconstruction. Finally, comprehensive experiments are conducted on a hardware-in-the-loop (HIL) testbed to evaluate the overall performance of the framework. The contributions of this study are summarized as follows.

- To address the issues of data loss and sparse data reconstruction caused by low-quality communication environments in rescue scenarios, this paper proposes RFDR. RFDR retrieves from an offline database the reference samples most similar to the currently received sparse sample and utilizes them to model usable data correlations, thereby achieving low-error reconstruction of the sparse sample.
- To improve the accuracy and efficiency of sample similarity computation, this paper proposes a soft-sliced dynamic time warping method (SS-DTW). This approach divides time series into partially overlapping segments, allowing the overlapping windows to incorporate contextual information from adjacent segments. This enhances computational efficiency while maintaining global consistency.
- To improve the accuracy of missing value imputation, this paper proposes a time series interpolation method capable of simultaneously modeling bidirectional spatiotemporal dependencies across samples, between variables, and within variables. By incorporating additional contextual constraints to narrow the prediction range of missing values, this method reduces the reconstruction error of the predicted values.
- We implement and evaluate RFDR on a HIL testbed. RFDR improves packet throughput under low-quality wireless links and surpasses baselines in reconstruction accuracy across a wide range of data missing rates, maintaining robust performance even under severe loss.

The remainder of this paper is organized as follows. Section 2 reviews related work; Section 3 formulates the problem and introduces preliminaries; Section 4 details the two core modules of RFDR; Section 5 presents experimental results; and Section 6 concludes the paper.

2 Related work

2.1 Loss and retransmission

To mitigate congestion, packet loss, and other transmission anomalies, a variety of complementary approaches have been developed. Examples include resource allocation protocols [13–15] for dynamic optimization of channel usage and maximized bandwidth utilization; network topology optimization [16] to reconfigure connectivity and alleviate bottlenecks; transmission job scheduling [17] for organizing data traffic to enhance resource efficiency; and volatile environment sensing [18] to enable real-time adaptation to dynamic channel conditions. At the link layer, pTunes for IEEE 802.15.4 [19] and AdaptiveLoRa for LoRaWAN [20] are prominent examples. StarCure [21] employed time-varying graphs to model temporal topology, converting fault-induced topology changes into traffic variations and applying an adaptive hybrid-routing scheme to handle unexpected failures. Wang et al. [22] recasted the minimization of data delivery time as a maximum-flow problem, whereas Chen et al. [23] proposed a pyramid-based job-scheduling model that exploits the parallelism of communication and computation. The distributed scheduler

Table 1 Comparison of methods.

Method	Venue	Category	Interpolation uncertainty	Architecture	Missing mechanism
BRITS [25]	NeurIPS	Predictive	×	RNN	MCAR
GRIN [26]	TITS	Predictive	×	GNN	MCAR/MAR
DCRNN [27]	ICLR	Predictive	×	RNN	MCAR/MAR
PoGeVon [28]	KDD	Predictive	×	VAE	MCAR/MAR/MNAR
DMSTG [29]	TMC	Predictive	×	GNN, Attention	MCAR/MAR/MNAR
PSW-I [30]	ICLR	Predictive	✓	Optimal Transport	MCAR/MAR/MNAR
GRU-GAN [31]	TNNLS	Generative	✓	GAN, RNN	MCAR
E2GAN [32]	IJCAI	Generative	✓	GAN, RNN	MCAR
SSGAN [33]	NeurIPS	Generative	✓	GAN, RNN	MCAR
SGMCAI-DiT [34]	TKDE	Generative	✓	Diffusion	MCAR
Terra [35]	NeurIPS	Generative	✓	LoRA-based	MCAR/MAR/MNAR
MOMENT [36]	ICML	Large model	✓	Foundation model	MCAR
Timer [37]	ICML	Large model	✓	Foundation model	MCAR
Timemixer++ [38]	ICLR	Large model	✓	Foundation model	MCAR
Gpt4mts [39]	AAAI	Large model	✓	Large language model	MCAR/MAR/MNAR
LLM2Trsf [40]	NeurIPS	Large model	✓	Large language model	MCAR/MAR/MNAR
AutoTimes [41]	NeurIPS	Large model	✓	Large language model	MCAR/MAR/MNAR
TimeCMA [42]	AAAI	Large model	✓	Large language model	MCAR/MAR/MNAR

TapFinger [24] further improves flexibility in the use of scarce edge resources. Nevertheless, these solutions often consume significant bandwidth, which can cause channel congestion and buffer overflow, ultimately degrading throughput for subsequent traffic.

2.2 Sparse inference methods

Table 1 provides a systematic categorization and comparison of multivariate time-series imputation methods, highlighting their key characteristics and differences. A critical limitation of these methods, however, is their impracticality for emergency-rescue scenarios, which require efficient operation on resource-constrained micro-computing devices. In addition to architectural differences, these methods can be classified by their handling of uncertainty [12]. We distinguish between predictive models, which yield a single imputation, and generative models, which produce multiple plausible outputs to capture data stochasticity. For instance, BRITS [25], based on recurrent neural networks (RNNs), handles missing values by modeling temporal correlations. TimesNet [43], which leverages convolutional neural networks (CNNs), extracts features by transforming time series into two-dimensional representations to capture multi-periodicity. Meanwhile, GRIN [26], built on attention mechanisms, aggregates information via spatiotemporal graph attention. Generative models also play a significant role in this domain: PoGeVon [28] employs variational autoencoders (VAEs), SSGAN [44] operates under the generative adversarial network (GAN) framework, and DCRNN [27], a graph neural network (GNN)-based approach, utilizes diffusion convolution to model spatial correlations but is deterministic in nature. Together, these diverse methods demonstrate the effectiveness and innovation of different architectures in addressing the challenge of missing spatiotemporal data.

Large pre-trained models have recently emerged as powerful and versatile solutions, achieving state-of-the-art (SOTA) performance across numerous domains. However, their substantial computational and storage requirements render them impractical for resource-constrained devices, such as those used in emergency rescue scenarios. Although existing interpolation methods have proven effective at reconstructing missing data and significantly enhancing the performance of downstream prediction tasks [12], several critical limitations remain: (i) an over-reliance on temporal continuity, which results in unreliable temporal cues and degraded reconstruction quality during prolonged variable missingness; (ii) a persistent efficiency-accuracy trade-off, where current architectures fail to achieve high interpolation accuracy without exceeding the tight computational budgets of typical edge devices; and (iii) insufficient cross-variable modeling, as most techniques lack explicit methods to capture inter-variable correlations over long gaps, further limiting overall performance.

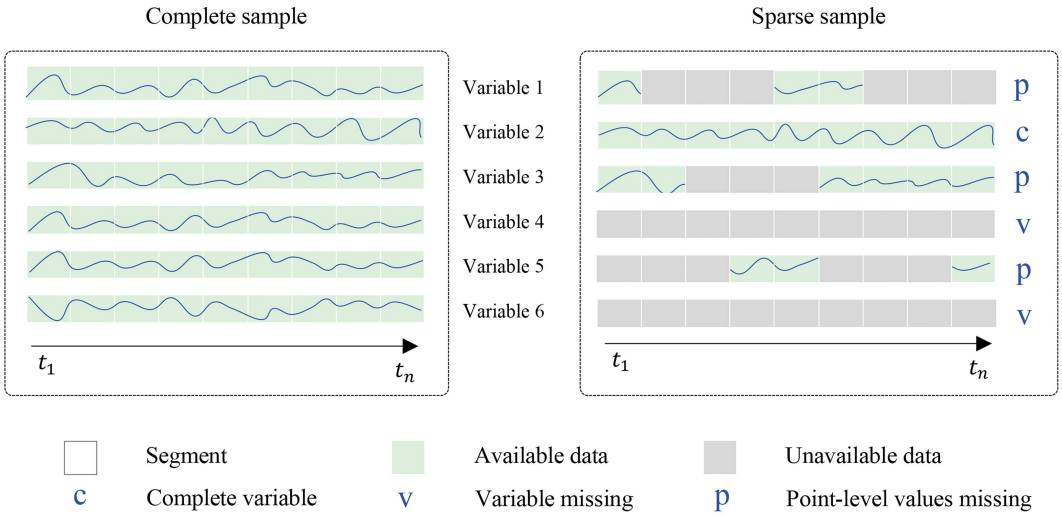

Figure 1 (Color online) Various states of missing data.

3 Problem formulation and preliminaries

3.1 Problem statement

The input to the downstream perception model must have both a fixed temporal resolution and feature dimensionality. This is typically achieved by preprocessing steps or prior layers (e.g., an embedding layer). However, in emergency response deployments that rely on temporary MANETs, packet loss occurs frequently. Although existing sparse-reconstruction methods can mask sporadic losses, they inevitably introduce reconstruction errors that often exceed tolerable limits when losses persist. Conventional systems therefore resort to retransmission by requesting the source node to resend the missing packets. While retransmission restores data availability, it also consumes bandwidth that would otherwise be allocated to subsequent traffic, leading to channel congestion and buffer overflows.

Given the stringent power and spectrum budgets in emergency rescue scenarios, repeated retransmissions significantly reduce network efficiency and can cause systemic failures. Thus, neither inference-based compensation nor retransmission-based recovery alone ensures reliable computing under these constraints.

Edge-collected samples, which are aggregated variables from multiple sensors, exhibit three possible data states for any given variable: (i) variable missing, (ii) point-level values missing, or (iii) complete variable. As shown in Figure 1, we consider a system with N end-deployed sensors. Each sensor transmits its time-series variable to the edge device at its own sampling rate. The edge device then aggregates these variables into a multivariate sample S.

3.2 Framework description

To overcome the limitations of both methods, we propose RFDR (Figure 2). The framework consists of two modules: the SS-DTW module and the time-series interpolation module. The SS-DTW module addresses variable missing. This design is motivated by the observation that although emergency-rescue networks suffer from high packet loss and volatile links, the multivariate temporal patterns are not unique to disaster scenarios and often find analogs in historical data from stable conditions or other scenarios. Therefore, the module identifies the most similar complete sample S_c from an offline repository based on the observed variables in S. It then extracts the values for the missing variables from S_c and imputes them to form a sample S' with a complete variable set (Steps 2–4). However, S' may still contain point-level values missing within individual variables. These are subsequently filled by the time-series interpolation module, which learns latent temporal dynamics to reconstruct the missing values, yielding a fully imputed sample (Step 5).

4 RFDR

RFDR consists of two core modules: the SS-DTW module and the time-series interpolation module, as illustrated in Figure 2.

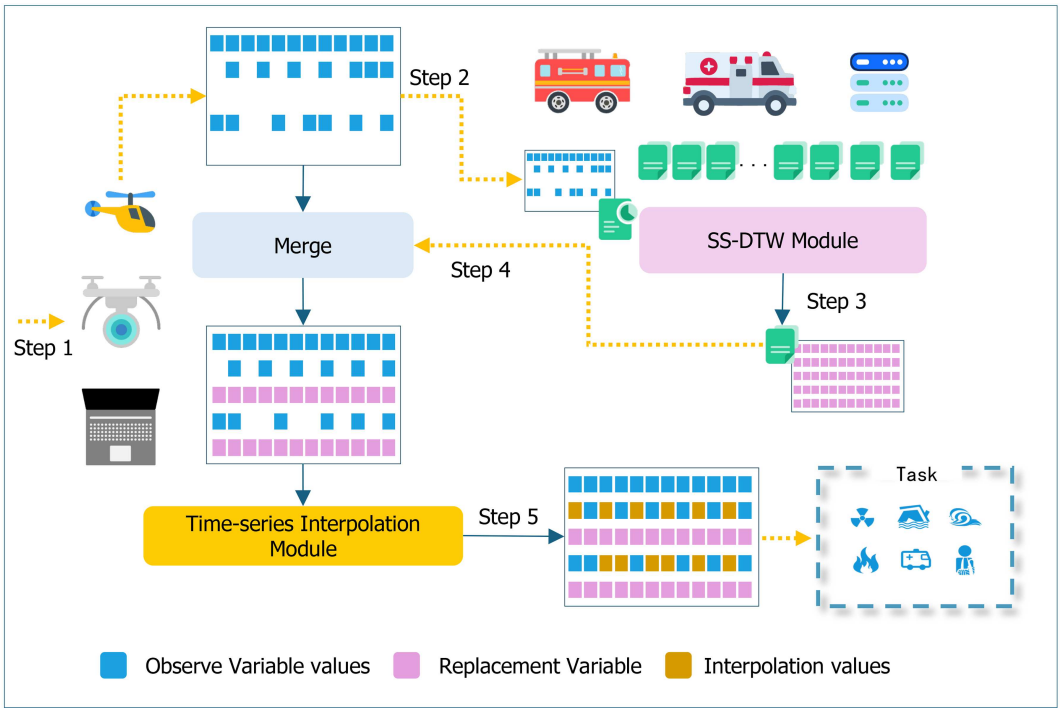

Figure 2 (Color online) Overview of RFDR (step 1: data collection from end devices; step 2: forwarding of the sparse sample to the offline data center; step 3: similarity computation and retrieval of the most relevant historical sample; step 4: interpolation of entirely missing variables from the retrieved sample; step 5: inference of remaining sporadic missing values).

- The SS-DTW module divides time series into partially overlapping segments. This allows the overlapping windows to incorporate contextual information from adjacent segments, which enhances computational efficiency while maintaining global consistency.
- The time-series interpolation module is capable of simultaneously modeling bidirectional spatiotemporal dependencies across samples, between variables, and within variables. By incorporating additional contextual constraints to narrow the prediction range of missing values, this method reduces the reconstruction error of the predicted values.

To improve the accuracy of missing value imputation, this paper proposes a time series interpolation method.

4.1 SS-DTW module

4.1.1 *Main idea*

In multi-sensor emergency rescue systems, the raw data are collected from d different sensors, each generating a univariate time series. A complete data sample is formed by synchronizing and combining all sensor readings over a fixed time period $[1, T]$. Mathematically, we define a full sample as a multivariate time series $S \in \mathbb{R}^{T\times d}$:

$$S = \{s_1, s_2, \ldots, s_T\}^\top,$$

where each vector $s_t \in \mathbb{R}^d$ measures the readings from all d sensors at timestamp t. However, in streaming scenarios, S is not available a priori. Instead, data are transmitted continuously as fixed-length segments. The full sequence is partitioned into l consecutive and potentially overlapping segments $\{S_1, S_2, \ldots, S_l\}$, each covering a short time interval of length k (where $k \ll T$). Each segment $S_i \in \mathbb{R}^{k\times d}$ is itself a multivariate time series slice, containing the synchronized readings from all d sensors over its respective time window:

$$S_i = \{s_{(i-1)\cdot p+1}, s_{(i-1)\cdot p+2}, ..., s_{(i-1)\cdot p+k}\}^\top.$$

This ensures that every data chunk preserves the multivariate structure of the source, enabling real-time processing. DTW is a widely used method for measuring similarity between time series [45], but it has two limitations in such settings: (i) DTW is designed for global alignment of complete series. However, when applied to short data segments, its performance degrades because the segmentation process itself disrupts the original patterns, causing the same pattern to appear across different segments; (ii) its $O(N^2)$ time complexity leads to high computational overhead

in edge computing scenarios, making it difficult to meet real-time requirements. To address these limitations, SS-DTW employs an overlapping window strategy for processing streaming data, reduces local similarity errors through adaptive alignment, significantly lowers computational complexity, and incorporates a dedicated early-stopping criterion tailored for large-sample databases.

4.1.2 *Preliminaries*

Definition 1 (Time series alignment)**.** Let $S_c = (S_{c,1}, S_{c,2}, \ldots, S_{c,N})$ and $S = (S_1, S_2, \ldots, S_M)$ be two time series of length N and M, respectively. An alignment path between S_c and S is a sequence $\pi = (\pi_1, \ldots, \pi_K)$, where each $\pi_k = (i_k, j_k)$ denotes a correspondence between the element S_{c,i_k} at index i_k in S_c and the element S_{j_k} at index j_k in S. The path must satisfy the following conditions.

- Boundary conditions: $\pi_1 = (1,1)$ and $\pi_K = (N, M)$.
- Monotonicity: $i_{k+1} \geqslant i_k$ and $j_{k+1} \geqslant j_k$ for all $k \in \{1, \ldots, K-1\}$.
- Step size constraints: $\pi_{k+1} - \pi_k \in \{(1,0), (0,1), (1,1)\}$ for all $k \in \{1, \ldots, K-1\}$.

The Euclidean distance between two time series of the same length assumes a strict temporal alignment with $\pi_k = (k, k)$. In contrast, DTW optimizes an elastic alignment via dynamic programming. Let $\Delta \in \mathbb{R}^{N \times M}$ be the pointwise distance matrix where $\Delta_{i,j} = |S_{c,i} - S_j|^p$ for a given L^p-norm (typically $p = 1$ or 2). The cumulative distance matrix $D \in \mathbb{R}^{N \times M}$ is computed recursively with proper initialization:

$$
\begin{aligned}
&D_{0,0} = 0, \\
&D_{i,0} = \infty, \quad \text{for } i = 1, \ldots, N, \\
&D_{0,j} = \infty, \quad \text{for } j = 1, \ldots, M, \\
&D_{i,j} = \Delta_{i,j} + \min\left\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\right\}, \quad \text{for } i = 1, \ldots, N,\ j = 1, \ldots, M.
\end{aligned}
$$

The optimal warping path P^* is the path that achieves the minimum cumulative distance, and the DTW distance between S_c and S is defined as

$$
\mathrm{DTW}(S_c, S) = D_{N,M} = \min_{P \in \mathcal{P}} \sum_{(i,j) \in P} \Delta_{i,j}, \tag{1}
$$

where $\mathcal{P}$ denotes the set of all valid warping paths.

Definition 2 (Subsequence alignment)**.** Given a reference subsequence $S_c^{(k:p)} = (S_{c,k}, S_{c,k+1}, \ldots, S_{c,p})$, and $S^{(q:r)} = (S_q, S_{q+1}, \ldots, S_r)$, the local DTW distance between them is computed by constraining the dynamic programming to the indices $i \in [k, p]$ and $j \in [q, r]$. The recurrence is initialized and computed as follows:

$$
\begin{aligned}
&D_{k-1,j} = \infty, \quad \text{for all } j, \\
&D_{i,q-1} = \infty, \quad \text{for all } i, \\
&D_{k,q} = \Delta_{k,q}, \\
&D_{i,j} = \Delta_{i,j} + \min\left\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\right\}, \quad \text{for } i \in [k+1, p],\ j \in [q+1, r].
\end{aligned}
$$

This initialization ensures that the path starts at (k, q), and the recurrence is restricted to the defined subsequence interval, ensuring the path ends at (p, r).

4.1.3 *Details of SS-DTW module*

As illustrated in Figure 3, existing methods typically perform time series alignment as follows. Figure 3(a) shows the reference sample S_c and the observed sample S. S_c contains two characteristic patterns (Pattern 1 and Pattern 2) that are also present in S but are subject to temporal distortions. These distortions, such as shifts, compression, or expansion, frequently occur in low-quality communication environments. Figure 3(d) demonstrates how conventional DTW aligns these patterns by computing a global warping path that accounts for the temporal distortions.

In contrast, the slice DTW method (Figure 3(b)) operates by dividing both time series S_c and S into fixed-length segments and aligning them independently. This approach results in a discontinuous warping path that fails to capture the global temporal structure, thereby distorting the underlying patterns. Figures 3(e)–(g) display the suboptimal alignments produced by the slice DTW method. The limitations of slice DTW are further confirmed by the discrepancy matrices of the optimal warping paths, shown in Figure 3(l), which reveal higher alignment errors compared to the conventional method.

To mitigate the distortion introduced by slice DTW, we designed the SS-DTW module. This method partitions the sequence into overlapping segments and performs a segment-wise alignment that is regularized by overlapping

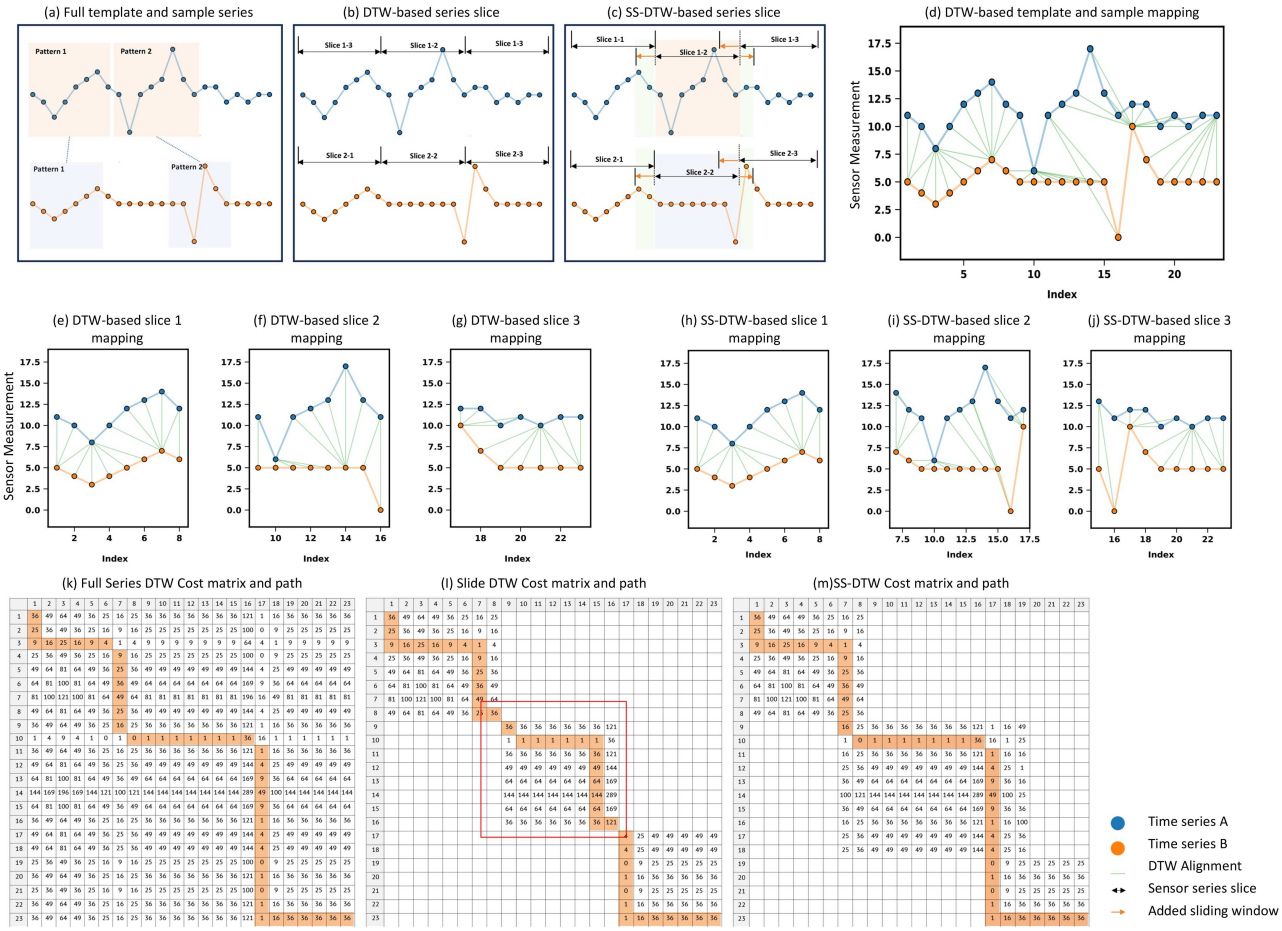

Figure 3 (Color online) The SS-DTW module.

segments to ensure global consistency. The procedure is illustrated in Figure 3(c) and consists of the following steps.

(i) Segmentation and extension: The reference sequence S_c of length n is partitioned into l contiguous segments. The segment boundaries are determined by a target segment length k, resulting in $l = \lceil n/k \rceil$ segments. Each segment Seg_p is then extended by a window of size w in both temporal directions. The first and last segments are handled as boundary cases: the first segment omits the backward extension (as no data exists prior to the start), and the last segment omits the forward extension. The query sequence S is partitioned synchronously using the same segment boundaries derived from S_c and extended identically.

(ii) Local alignment: For each extended segment pair $(\mathrm{Seg}_p(S_c), \mathrm{Seg}_p(S))$, we compute a local cumulative cost matrix C_p and an optimal warping path P_p within the extended window. The recurrence relation for each local matrix is identical to the standard DTW computation but is constrained within the segment's extended boundaries.

(iii) Global path fusion: The final SS-DTW similarity score is obtained by summing the cumulative costs along the optimal path within the *non-extended* core of each segment. Let P_p^* be the segment of the path restricted to the core indices of Seg_p. The global score is

$$\text{SS-DTW}(S_c, S) = \sum_{p=1}^{l} \sum_{(i,j)\in P_p^*} \beta_{i,v}. \tag{2}$$

This approach ensures the alignment is both globally optimal across segments and locally precise within them. Figures 3(h)–(j) demonstrate that this method preserves the holistic pattern structure by maintaining continuous warping paths that respect the natural temporal order of the data. Figure 3(m) confirms the high consistency between the full-sequence warping path and the concatenated soft-sliced paths, as evidenced by the clear diagonal structure in the discrepancy matrix, which is similar to that of conventional DTW.

Definition 3 (Distance matrix)**.** The pairwise local distance matrix $M \in \mathbb{R}^{n\times m}$ is the foundational element for

computing alignment costs. Each entry $M_{i,j}$ is defined using the Minkowski L^p-norm:

$$M_{i,j} = \|\mathbf{S}_{c,i} - \mathbf{S}_j\|_p^p = \sum_{d=1}^{D} |S_{c,i}^{(d)} - S_j^{(d)}|^p, \tag{3}$$

where $p \geqslant 1$ denotes the norm order, D measures the dimensionality of each point, and $S_{c,i}^{(d)}$ denotes the d-th dimension of the i-th point in S_c. This definition applies to both univariate ($D = 1$) and multivariate ($D > 1$) time series.

Definition 4 (Time series similarity)**.** Using the distance matrix M, the time series similarity matrix $C \in \mathbb{R}^{(n+1)\times(m+1)}$ is constructed via accumulated cost matrix. This matrix recursively aggregates local distances to find the optimal alignment path, with its boundary conditions and recurrence relation defined as follows:

$$\begin{aligned}
&C_{0,0} = 0,\\
&C_{i,0} = \infty, \quad \text{for } i = 1, \ldots, n,\\
&C_{0,j} = \infty, \quad \text{for } j = 1, \ldots, m,\\
&C_{i,j} = M_{i,j} + \min\{C_{i-1,j},\ C_{i,j-1},\ C_{i-1,j-1}\}, \quad \text{for } i = 1, \ldots, n,\ j = 1, \ldots, m.
\end{aligned}$$

The final time series similarity between S_c and S is then given by the value at the corner of the accumulated cost matrix:

$$\text{SS-DTW}(S_c, S) = C_{n,m}. \tag{4}$$

4.1.4 *Early termination strategy*

To enhance the computational efficiency of matching a real-time input sample S against a large offline reference dataset $\mathcal{D} = \{S_c^{(1)}, S_c^{(2)}, \ldots, S_c^{(K)}\}$, we propose an early termination strategy leveraging its properties of non-negativity and additivity. The strategy is outlined in Algorithm 1.

Let c be the best-match distance found so far during the search:

$$c = \min_{1 \leqslant i < k} \text{SS-DTW}\big(S_c^{(i)}, S\big). \tag{5}$$

The SS-DTW distance between a reference sample $S_c^{(k)}$ and the sample S is computed as the sum of the distances of their aligned segment pairs:

$$D^{(k)} = \sum_{p=1}^{\ell} d_p^{(k)}, \quad \text{where} \quad d_p^{(k)} = \text{SS-DTW}\big(S_{c,p}^{(k)}, S_p\big). \tag{6}$$

Here, $S_{c,p}^{(k)}$ and S_p denote the p^{th} segments of $S_c^{(k)}$ and S, respectively. Since $d_p^{(k)} \geqslant 0$ for all p, the partial sum $\sum_{p=1}^{q} d_p^{(k)}$ is non-decreasing with q. To enable early termination, we maintain a running lower bound for the total distance. After processing the qth segment, the total distance $D^{(k)}$ can be bounded from below:

$$\hat{D}_q^{(k)} = \sum_{p=1}^{q} d_p^{(k)} + \sum_{p=q+1}^{\ell} \delta_p^{(k)}, \tag{7}$$

where $\delta_p^{(k)} \geqslant 0$ is a cheap-to-compute lower bound estimate for the future distance $d_p^{(k)}$. A simple and effective choice is to set $\delta_p^{(k)} = 0$, which leverages only the non-negativity of the distance. Early termination for reference k is triggered whenever

$$\hat{D}_q^{(k)} \geqslant c \quad \text{for any } q < \ell. \tag{8}$$

4.2 Time-series interpolation module

The observed sample simultaneously suffers from two types of issues: variable missing and point-level values missing. The SS-DTW module addresses the issues of variable missing by extracting the complete data of the corresponding variable from the reference sample and transferring them to the observed sample, thus generating sample S'.

Algorithm 1 Early termination for SS-DTW reference matching.

Input: Reference database $\mathcal{D} = \{S_c^{(1)}, \ldots, S_c^{(K)}\}$, query sample S, target segment length k, stride s, extension window w.
Output: Optimal reference index k^*, minimum distance c_{best}.

Initialization:
$c_{\text{best}} \leftarrow \infty$;
$k^* \leftarrow -1$; // Initialize to an invalid index;
for $k \leftarrow 1$ ***to*** K **do**
  // Determine segment boundaries for reference $S_c^{(k)}$ using k and s;
  $\ell \leftarrow$ number of segments in $S_c^{(k)}$;
  // Segment query sample S synchronously using the boundaries from $S_c^{(k)}$;
  $D^{(k)} \leftarrow 0$; // Accumulated distance for reference k;
  $q \leftarrow 1$; // Segment index;
  while $q \leqslant \ell$ ***and*** $D^{(k)} < c_{best}$ **do**
    // Compute lower bound for remaining segments (using $\delta_p^{(k)} = 0$ as simple estimate);
    $\hat{D}^{(k)} \leftarrow D^{(k)} + 0$; // $\sum_{p=q+1}^{\ell} \delta_p^{(k)} = 0$ when $\delta_p^{(k)} = 0$;
    if $\hat{D}^{(k)} \geqslant c_{best}$ **then**
      break // Early termination: skip the rest of reference k;
    // Compute the expensive segment alignment;
    $d_q^{(k)} \leftarrow \text{SS-DTW}(S_{c,q}^{(k)}, S_q, w)$;
    $D^{(k)} \leftarrow D^{(k)} + d_q^{(k)}$;
    $q \leftarrow q + 1$;
  if $D^{(k)} < c_{best}$ **then**
    $c_{\text{best}} \leftarrow D^{(k)}$;
    $k^* \leftarrow k$;
return k^*, c_{best}.

However, S' may still contain point-level values missing inherited from the raw sparsity sample caused by low-quality communication environments. During inference, we leverage the available observations to capture data correlations. The accuracy of interpolation critically depends on the number of observed entries—a greater number of observations strengthens the correlations to missing values and leads to higher reconstruction accuracy. As illustrated in Figure 4, we model two types of correlations: bidirectional intra-sensor correlations and inter-sensor correlations.

- Bidirectional intra-sensor modeling: To capture bidirectional temporal correlations, we reverse each time-series variable and apply an attention mechanism that connects timestamp t to both $t - \Delta t$ and $t + \Delta t$. This allows an observation at t to capture correlations with both its past $(t - \Delta t)$ and future $(t + \Delta t)$ contexts, thereby enhancing the discovery of temporal correlations.
- Inter-sensor correlations modeling: We employ a graph-based network, in which the hidden states of nodes are updated through message-passing layers with interpolation methods. Observations from neighboring nodes propagate spatiotemporal constraints, resulting in more accurate estimates of the missing values.

4.2.1 *Formal definition*

For each sample S_i, we construct a directed weighted graph $G_i = (V_i, E_i)$, termed the sensor correlation graph. The node set V_i corresponds to sensors, and the edge set E_i encodes pairwise correlation relations. Each edge from node u to node v is associated with a weight $e_{i,uv} \in [0, 1]$, representing the strength of the relationship. If $e_{i,uv} \neq 0$, sensor u can pass neural messages to sensor v; otherwise, the two sensors are considered independent. All graphs are initialized as fully connected directed graphs (i.e., $e_{i,uv} = 1$ for all possible edges) and are adaptively updated during training.

Consider a multivariate time series comprising N sensor variables observed over a time window of length W, which can be represented as a matrix $\mathbf{V} \in \mathbb{R}^{W \times N}$. A corresponding mask matrix $\mathbf{M} \in \{0, 1\}^{W \times N}$ indicates missing entries: $m_{n,t} = 0$ if the value $v_{n,t}$ is missing. Model operations (e.g., variable embedding and inference) are performed only on the observed values (where $m_{n,t} = 1$). To capture complex spatiotemporal correlations, sensors are interconnected via dynamically learned edge weights, enabling effective joint modeling of both spatial and temporal correlations.

4.2.2 *Embedding observations of active sensors*

This section describes the method for generating observation embeddings for any sensor at a given timestamp. Let $x_{i,u}^{(t)} \in \mathbb{R}^d$ denote the preprocessed observation from sensor u in sample i at time t. To enhance model capacity,

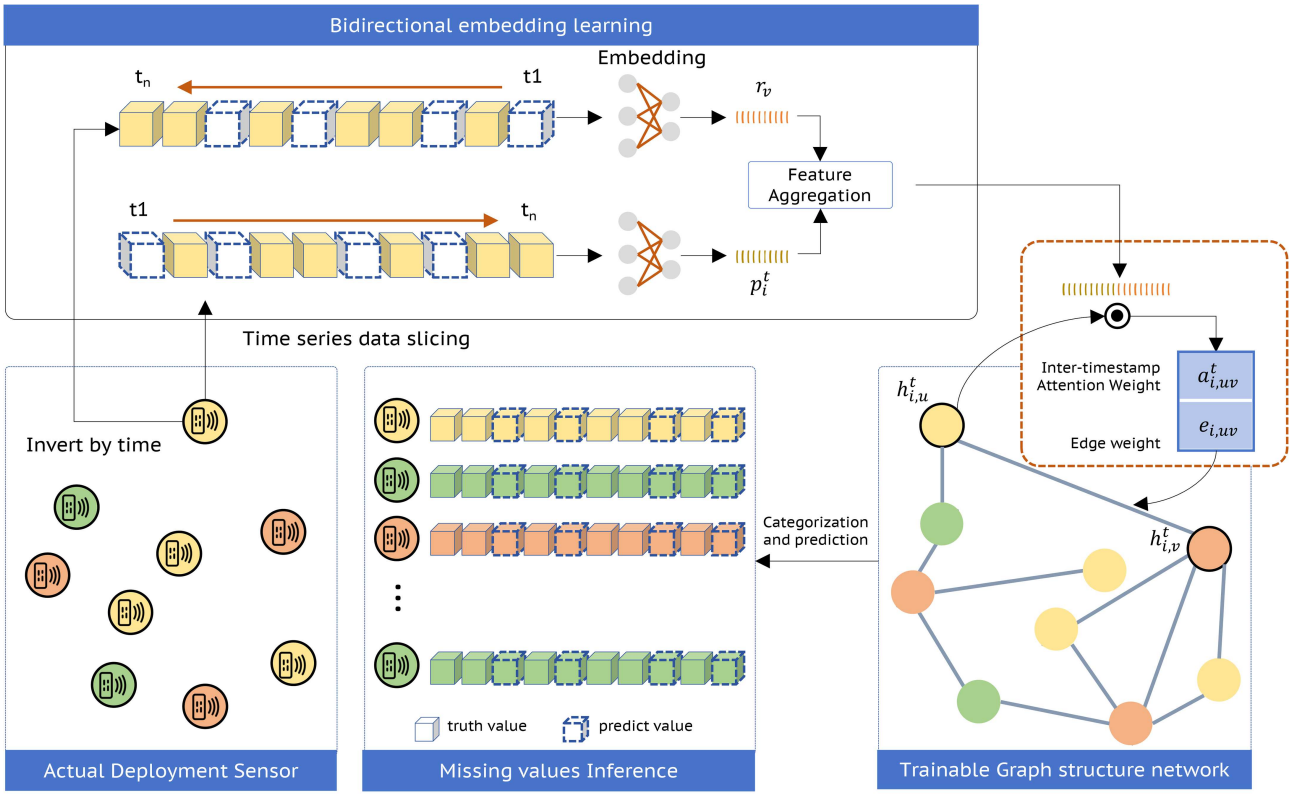

Figure 4 (Color online) Time series interpolation module.

each observation is nonlinearly projected into a high-dimensional space via a sensor-specific transformation:

$$h_{i,u}^{(t)} = \sigma\left(x_{i,u}^{(t)} R_u\right), \tag{9}$$

where $R_u \in \mathbb{R}^{d \times d_h}$ is a trainable weight matrix unique to sensor u (shared across all samples and timestamps), and $\sigma(\cdot)$ is the ReLU activation function. This transformation facilitates the handling of sparse and irregular observations. To capture bidirectional temporal correlations, the reversed observation sequence $\overleftarrow{x}_{i,u}^{(t)}$ is processed independently using a separate set of parameters:

$$\overleftarrow{h}_{i,u}^{(t)} = \sigma\left(\overleftarrow{x}_{i,u}^{(t)} R'_u\right), \tag{10}$$

where $R'_u \in \mathbb{R}^{d \times d_h}$ is a distinct, learned weight matrix for the reverse temporal direction. The forward and backward representations are subsequently concatenated to form a fused representation:

$$\hat{h}_{i,u}^{(t)} = \left[h_{i,u}^{(t)} \parallel \overleftarrow{h}_{i,u}^{(t)}\right] \in \mathbb{R}^{2d_h}. \tag{11}$$

4.2.3 *Message passing for inactive sensor imputation*

To estimate the embedding of an inactive sensor v at time t, our method leverages information from its active neighbors in the graph G_i through a message-passing mechanism. For a neighboring sensor $u \in \mathcal{N}(v)$, an attention weight $\alpha_{i,uv}^{(t)} \in [0,1]$ is computed to quantify the relevance of u to v:

$$\alpha_{i,uv}^{(t)} = \text{sigmoid}\left(\left(h_{i,u}^{(t)} D\right) \cdot \left[r_v \parallel p^{(t)}\right]^\top\right). \tag{12}$$

Here, $r_v \in \mathbb{R}^{d_r}$ is a trainable, sensor-specific embedding vector for the receiver node v, allowing the model to discern the importance of different senders. The vector $p^{(t)} \in \mathbb{R}^{d_t}$ is a temporal encoding that injects timestamp-specific

information, defined using trigonometric functions:

$$\begin{aligned} p_{2k}^{(t)} &= \sin\left(\frac{t}{q^{2k/d_t}}\right), \\ p_{2k+1}^{(t)} &= \cos\left(\frac{t}{q^{2k/d_t}}\right), \end{aligned} \tag{13}$$

where q is a frequency scaling factor set to 10000. The matrix $D \in \mathbb{R}^{d_h \times (d_r + d_t)}$ is a learnable projection matrix. The estimated embedding for an inactive sensor v is then computed as an attention-weighted sum over the embeddings of all its active neighbors:

$$h_{i,v}^{(t)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{i,uv}^{(t)} \cdot e_{i,uv} \cdot \left(h_{i,u}^{(t)} W\right)\right), \tag{14}$$

where $W \in \mathbb{R}^{d_h \times d_h}$ is a shared learnable weight matrix, and $e_{i,uv}$ is the edge attribute between u and v.

4.2.4 *Temporal aggregation into fixed sensor embeddings*

The previous steps yielded a set of observation embeddings $\{h_{i,v}^{(t)} \mid t \in T_{i,v}\}$ for all timestamps where the data were available either at sensor v or within its neighborhood. To aggregate these variable-length sequences into a fixed-length sensor embedding $z_{i,v}$, we employ a temporal attention mechanism. The observation embedding at each time t is first concatenated with its corresponding temporal encoding $p^{(t)}$ to form a composite feature vector $[h_{i,v}^{(t)} \parallel p^{(t)}] \in \mathbb{R}^{2d_h + d_t}$. All such vectors for $t \in T_{i,v}$ are stacked into a matrix $H_{i,v} \in \mathbb{R}^{|T_{i,v}| \times (2d_h + d_t)}$.

To obtain a single vector representation for the entire sequence, we compute the final sensor embedding $z_{i,v} \in \mathbb{R}^{d_z}$ as the weighted sum of the value-projected features, using attention weights derived from a dedicated, learnable query vector $q_{cls} \in \mathbb{R}^{1 \times d_q}$:

$$\beta_{i,v} = \mathrm{softmax}\left(\frac{(H_{i,v} W_K)(q_{cls} W_Q')^\top}{\sqrt{d_k}}\right), \quad z_{i,v} = \sum_{t=1}^{|T_{i,v}|} \beta_{i,v}^{(t)} \cdot \left(H_{i,v}^{(t)} W_V\right), \tag{15}$$

where $W_Q' \in \mathbb{R}^{d_q \times d_k}$, $W_K \in \mathbb{R}^{(2d_h + d_t) \times d_k}$, and $W_V \in \mathbb{R}^{(2d_h + d_t) \times d_z}$ are learned projection matrices, d_k is the dimension of the key/query space, $H_{i,v}^{(t)}$ is the t-th row of $H_{i,v}$, and $\beta_{i,v}^{(t)}$ is the t-th component of the attention weight vector.

4.2.5 *Readout and reconstruction loss*

For each sample S_i, the embeddings of all N sensor-generated embeddings $\{z_{i,v}\}_{v=1}^N$ are aggregated into a sample-level representation $z_i \in \mathbb{R}^{d_z}$ via a readout function g. We implement g as mean pooling:

$$z_i = \frac{1}{N} \sum_{v=1}^{N} z_{i,v}. \tag{16}$$

The sample embedding z_i is then passed through a multi-layer perceptron (MLP) head which projects it into the imputation space and directly produces the imputed values $\hat{Y}_i$ for all sensors and timestamps in the sample.

The model is trained to minimize the reconstruction error over the observed values. For a batch of samples, the loss function is defined as

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \frac{\sum_{t=1}^{T} \sum_{v=1}^{N} m_{i,v}^{(t)} \cdot \ell\left(\hat{y}_{i,v}^{(t)}, y_{i,v}^{(t)}\right)}{\sum_{t=1}^{T} \sum_{v=1}^{N} m_{i,v}^{(t)}}, \tag{17}$$

where $\hat{y}_{i,v}^{(t)}$ and $y_{i,v}^{(t)}$ are the reconstructed and true values, respectively, $m_{i,v}^{(t)} \in \{0, 1\}$ is a binary mask indicating whether the value was observed (1) or missing (0), $\ell(\cdot, \cdot)$ is an element-wise error function, such as mean absolute error (MAE) or mean squared error (MSE), and mean relative error (MRE), B is the number of samples in the batch.

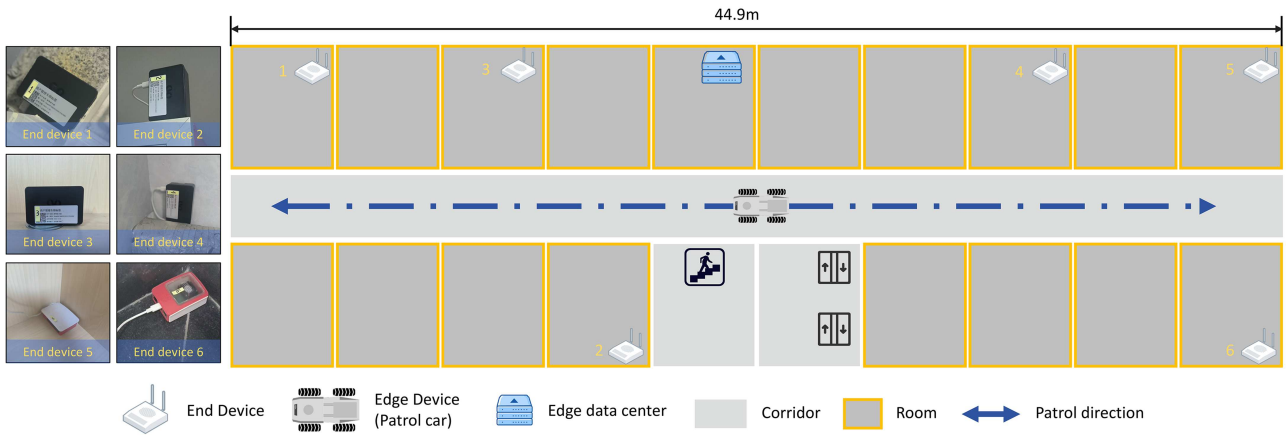

Figure 5 (Color online) The end-edge collaborative wireless communication system.

5 Experiments

5.1 Datasets

We evaluate our method on various public datasets spanning three domains: healthcare, environment, and traffic. Healthcare: The MIMIC open-source database contains anonymized records from 38803 intensive care patients monitored by 36 sensors. Each patient record includes multiple binary labels for common clinical events [46]. Environment: This dataset comprises air quality index measurements from 437 monitoring sites. We focus on $PM_{2.5}$ concentrations defined as AIR, and also use a reduced version with 36 sensors, defined as AIR-36 [47]. Traffic: We employ the PEMS-BAY and METR-LA traffic sensor datasets [27].

5.2 Baselines

We compare our method with the following baselines on the tasks of imputation and forecasting. GRIN [48]: A graph-based imputation method that replaces the standard gating mechanism in RNNs with a message-passing network to handle missing data. TimesNet [43]: A general-purpose time series modeling framework that transforms time series into two-dimensional representations based on temporal periods for analysis. It employs a residual dilated inception block as its convolutional backbone to capture multi-scale temporal patterns. We adapt it for the imputation task. DCRNN [27]: A spatio-temporal GNN designed for traffic forecasting. It captures spatial correlations via bidirectional random walks on a graph and models temporal correlations with an encoder-decoder architecture utilizing scheduled sampling. We apply its paradigm to the imputation problem. PoGeVon [28]: a graph variational autoencoder approach that performs imputation of missing values in both the node-level time series and the underlying graph structure simultaneously. SSGAN [33]: a generative adversarial network for spatio-temporal imputation. It introduces a temporal reminder matrix to guide the generator in producing more accurate estimates of missing values. BRITS [25]: a bidirectional recurrent imputation model that learns missing values in multivariate time series based on both forward and backward directions and the empirical correlations between different channels. DMSTG [29]: a dynamic graph learning model for traffic data imputation. It utilizes a self-attention module to dynamically assign importance weights to different components and a variable-fusion module to integrate multi-view spatio-temporal representations.

5.3 Experimental configuration

We deployed the end-edge HIL testbed consisting of six portable Raspberry Pis (as sensing nodes), one laptop mounted on a patrol car that acted as a portable edge node, and one server functioning as an edge-located, small-scale offline data center (see Figure 5). To simulate dynamic channel conditions, the patrol car moved continuously, and physical obstacles were periodically introduced into the environment to simulate signal blockage. The sensors generated data packets of up to 200 kB. These packets were first buffered in a 1 MB queue on the sensing node. Any undelivered packets were moved to a retransmission queue. Prolonged latency and frequent failures caused packets to accumulate, leading to buffer overflow and data missing. We evaluated the following two schemes.

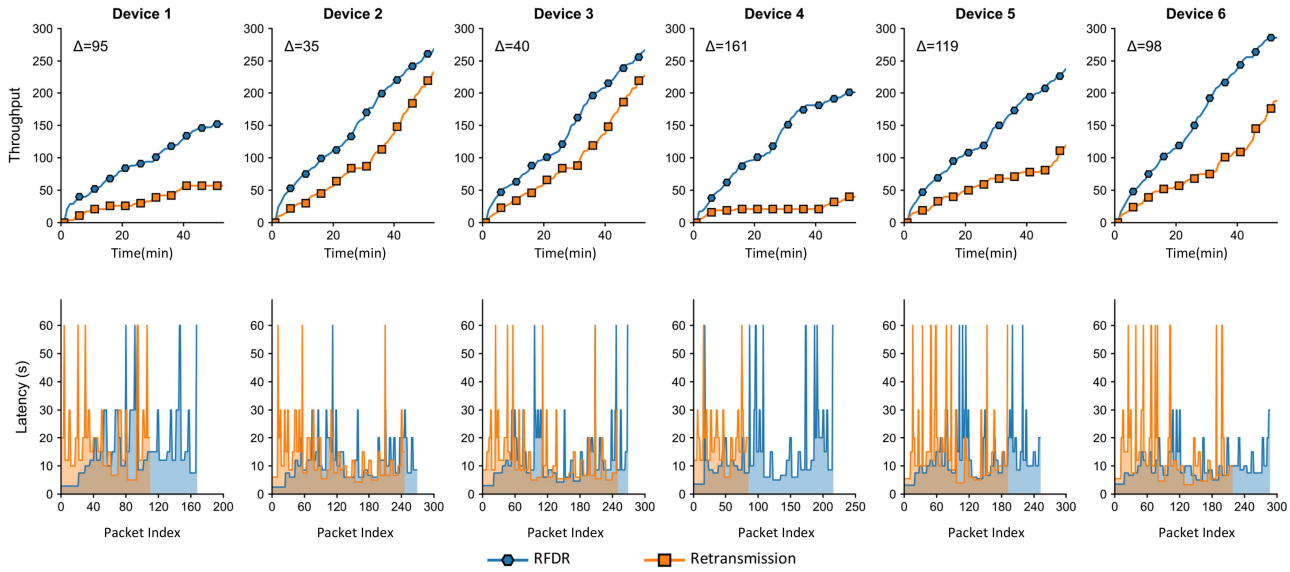

Figure 6 (Color online) Throughput and transmission delay (Δ is cumulative differential values).

• *Retransmission*: This scheme employed a classic ARQ-like recovery mechanism. Upon detecting a packet loss, the receiver sent a retransmission request, prompting the sensing node to resend the packet. A packet delivery is considered a failure if a buffer overflow occurs or if a timeout is triggered before successful transmission.

• *RFDR*: This scheme adopted a loss-tolerant streaming strategy. Upon detecting a loss, it did not request a retransmission. Instead, the edge device queried its offline database for a semantically similar data segment to compensate for the loss, while the system continued to stream the subsequent batch of real-time data.

5.4 Performance metrics

We selected two types of performance metrics for a comprehensive evaluation: transmission performance and reconstruction performance. For the former, we measured throughput, defined as the amount of successfully transmitted data over the test duration, and latency, quantified as the time elapsed from a packet's transmission to its successful receipt, including any delays incurred by loss and retransmission. For the latter, we employed three common metrics to assess imputation accuracy: the MAE, which measures the average magnitude of errors between imputed and ground truth values; the MSE, which calculates the average of squared errors to emphasize larger errors; and the MRE, which normalizes the absolute error by the actual value to provide a relative measure.

5.5 Transmission performance

The first row of Figure 6 compares the real-time throughput of RFDR and the retransmission scheme for six mobile devices during the 60-minute experiment. The second row depicts the corresponding time series of packet transmission latency, where each point represents the delay of an individual packet in chronological order. Owing to the continuous movement of the devices and introduced obstacles, the channel conditions were dynamic, causing both metrics to exhibit significant volatility. We quantify the performance gap as the difference in mean throughput (denoted as Δ) between the two schemes. For Devices 2 and 3, which experienced relatively stable channels, RFDR achieved a mean throughput gain of $\Delta = 37.5 \pm 3.5$. In contrast, for Devices 1, 4, 5, and 6, which suffered from severe channel fluctuations, the performance improvement was more pronounced, with a throughput gain of $\Delta = 118.3 \pm 30.4$ and higher volatility (coefficient of variation = 25.7%). Crucially, RFDR also yielded a substantial reduction in latency. As visible in the second row of Figure 6, the retransmission scheme exhibited frequent latency spikes caused by retransmission timeouts following packet loss. RFDR, by eschewing retransmissions, eliminates these spikes and maintains a consistently lower and more predictable delay profile. In summary, the results demonstrate that RFDR consistently outperforms the conventional retransmission method in both throughput and latency, with its advantages being particularly significant under highly volatile channel conditions. This robust performance underscores the framework's strong potential for application in demanding scenarios such as emergency rescue operations.

The latency distribution of data packets reveals a significant head-of-line blocking issue. This phenomenon is characterized by a few packets consuming a disproportionate share of the transmission time, which stalls the en-

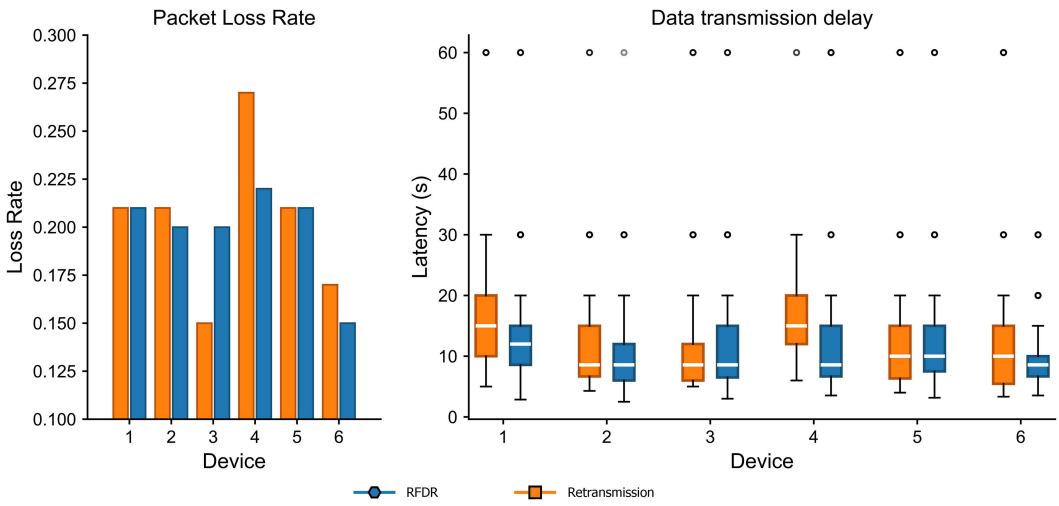

Figure 7 (Color online) Packet loss rate and data transmission delay.

tire queue and induces cascading delays for subsequent packets, thereby significantly reducing overall transmission efficiency. In contrast, RFDR mitigates this problem through a novel method. It abandons conventional retransmission, which intensifies channel contention. Instead, upon detecting missing data, RFDR leverages nearby edge computing nodes to retrieve or infer the missing information from an existing dataset. This strategy bypasses repeated transmissions over the end-edge link, effectively freeing up channel resources. In our experimental evaluation, this method achieved an average improvement in transmission efficiency of 17.32% compared to the retransmission scheme.

Figure 7 compares packet loss rate and latency between the two schemes. The average packet loss rate is similar (20.33% for retransmission vs. 19.83% for RFDR), confirming consistent network conditions. The box plot shows that RFDR achieves lower latency across most devices (mean reduction: 1.81 s) and a smaller interquartile range (IQR reduction: 1.74 s), indicating superior robustness—especially under poor network conditions. The retransmission scheme exhibited higher latency variability, indicating higher sensitivity to network variations.

5.6 Reconstruction performance under various scenarios

Figure 8 provides a systematic comparison with baselines in the field of sparse data reconstruction. To evaluate the generalization capability and robustness of our method, we conducted comprehensive benchmark tests on multiple datasets. Following Zhang et al. [49], we adopted MAE, MSE, and MRE as evaluation metrics. We established two experimental conditions. In-sample evaluation: Training and validation sets originate from the same data distribution. Out-of-sample evaluation: The validation set is sourced from a significantly different data origin than the training set. The experimental results, under the given experimental conditions, show that RFDR significantly outperformed all baselines across all 30 evaluation tasks (5 datasets $\times$ 2 settings $\times$ 3 metrics). Compared to the second-best method, it achieved an average reduction of 49.94% in MAE, 8.86% in MSE, and 16.25% in MRE, demonstrating its superior cross-domain reconstruction capabilities and stability.

5.7 Reconstruction performance under various missing ratios

Through a comprehensive evaluation across various missing ratios (ranging from 35% to 85%; see Table 2), we empirically show that our proposed RFDR achieves consistent improvements in cross-domain sparse data reconstruction tasks. Specifically, RFDR attains optimal performance in all 36 comparative scenarios (6 missing ratios $\times$ 3 metrics $\times$ 2 settings), with average relative improvements of 18.18% in MAE, 8.71% in MSE, and 30.29% in MRE compared to the second- best method. The notably higher improvement in MAE and MRE compared to MSE suggests that our method is particularly effective at reducing common errors, while its advantage in mitigating large outliers is more modest. Notably, the performance gap progressively widens as the missing data ratio increases. The method also exhibits superior generalization in out-of-sample conditions. It achieves a 56.5% reduction in MAE at a 35% missing ratio (7.92 vs. DMSTG's 18.21) and maintains an average MAE improvement of 44.5%, compared to 34.7% in in-sample conditions. This confirms its exceptional generalization capability under distribution shifts. At an extreme missing ratio of 85% in-sample, RFDR achieves an MAE of 27.63 (a 17.3% reduction over DCRNN's

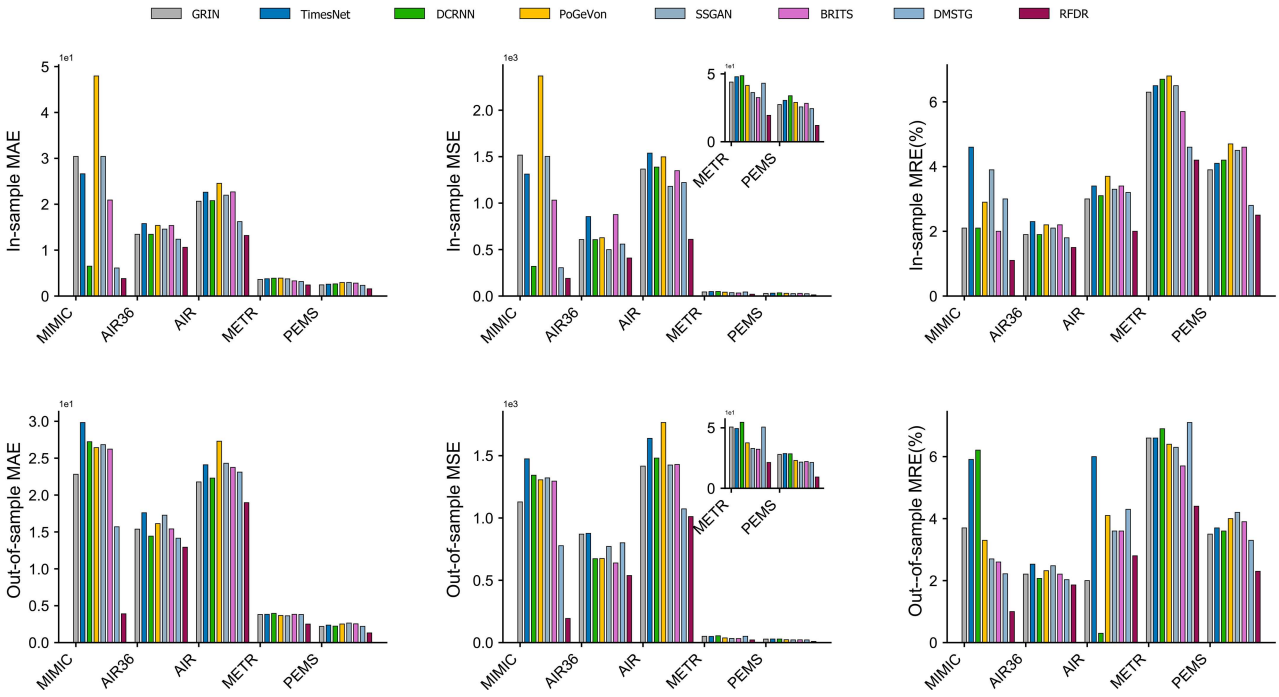

Figure 8 (Color online) Comparison of reconstruction errors across datasets and configurations.

Table 2 Performance comparison across different ratios. The best results are in bold. The second best results are underlined.

Missing ratio	Method	In-sample			Out-of-sample			Missing ratio	Method	In-sample			Out-of-sample		
		MAE	MSE	MRE (%)	MAE	MSE	MRE (%)			MAE	MSE	MRE (%)	MAE	MSE	MRE (%)
35%	GRIN	23.71	1211.13	4.10	26.02	1355.44	4.70	65%	GRIN	45.75	1998.27	5.75	51.18	2398.32	7.42
	TimesNet	30.12	1521.51	5.54	32.89	1769.44	6.61		TimesNet	51.45	2172.66	9.02	57.53	2607.94	11.01
	DCRNN	25.41	1221.21	5.15	30.24	1612.29	7.41		DCRNN	28.23	1455.08	10.04	32.01	1746.34	12.85
	PoGeVon	22.34	1113.13	2.11	29.77	1567.94	4.59		PoGeVon	45.55	2167.52	5.75	50.19	2601.69	7.70
	SSGAN	20.13	1223.11	1.89	30.42	1586.72	3.23		SSGAN	42.34	2142.33	5.31	47.14	2570.87	6.52
	BRITS	18.13	1012.34	1.44	29.06	1555.09	3.22		BRITS	46.35	1943.09	6.04	51.28	2332.69	7.52
	DMSTG	12.45	745.23	1.98	18.21	934.02	2.48		DMSTG	28.61	1311.39	4.52	32.02	1573.84	6.27
	RFDR	**7.05**	**693.79**	**1.10**	**7.92**	**833.00**	**2.10**		RFDR	**24.32**	**1197.53**	**2.94**	**27.45**	**1437.57**	**4.39**
45%	GRIN	30.73	1517.52	4.25	34.40	1821.10	4.90	75%	GRIN	51.05	2198.69	6.66	56.19	2638.54	8.78
	TimesNet	40.04	1975.37	6.98	44.69	2371.29	7.73		TimesNet	53.05	2390.57	10.90	58.66	2869.25	14.02
	DCRNN	26.83	1323.42	7.50	30.04	1588.91	8.83		DCRNN	31.93	1600.89	11.56	35.45	1921.51	14.19
	PoGeVon	33.33	1643.44	4.03	36.79	1972.65	5.00		PoGeVon	49.65	2384.31	6.43	55.03	2861.83	7.75
	SSGAN	33.83	1669.67	3.67	37.93	2004.21	4.60		SSGAN	47.75	2357.05	6.67	53.09	2829.23	8.29
	BRITS	31.73	1565.36	3.84	35.25	1879.35	4.75		BRITS	49.05	2137.72	6.80	54.10	2565.36	8.48
	DMSTG	17.31	853.82	2.69	19.85	1025.39	3.78		DMSTG	30.55	1443.40	5.63	34.10	1732.43	6.95
	RFDR	**15.82**	**780.08**	**2.00**	**18.14**	**936.86**	**2.29**		RFDR	**27.13**	**1317.74**	**3.34**	**30.02**	**1581.75**	**4.01**
55%	GRIN	36.84	1816.31	5.19	40.82	2180.49	6.22	85%	GRIN	55.96	2418.81	7.79	62.19	2903.01	10.05
	TimesNet	40.04	1974.27	7.98	44.44	2369.19	9.39		TimesNet	57.36	2630.44	12.09	63.62	3156.90	15.37
	DCRNN	26.83	1322.72	8.76	29.53	1587.66	10.58		DCRNN	33.43	1761.26	13.13	37.26	2113.84	15.89
	PoGeVon	39.94	1970.17	5.06	44.12	2365.13	6.03		PoGeVon	59.26	2623.20	7.65	66.00	3148.05	9.90
	SSGAN	39.44	1947.25	4.30	43.44	2337.24	5.31		SSGAN	55.06	2593.01	8.10	60.65	3112.56	10.47
	BRITS	35.74	1765.66	4.81	40.00	2119.18	5.81		BRITS	53.25	2351.97	8.05	59.01	2823.35	9.75
	DMSTG	24.16	1191.27	3.77	27.53	1429.86	4.47		DMSTG	37.24	1588.69	6.53	41.11	1906.65	8.36
	RFDR	**22.02**	**1088.39**	**2.61**	**25.06**	**1306.20**	**3.82**		RFDR	**27.63**	**1449.86**	**3.97**	**30.44**	**1740.16**	**5.56**

33.43) and an MRE of 3.97% (a 39.2% reduction over DMSTG's 6.53). These consistent advantages substantiate RFDR's significant advancement in addressing the critical challenge of extreme sparse data reconstruction.

6 Conclusion

This paper proposes RFDR for data reconstruction that consists of two modules: (i) an SS-DTW module for handling missing variables, and (ii) a time-series interpolation module for imputing missing values. Unlike conventional retransmission methods that rely on requesting retransmission data from end devices, SS-DTW leverages an offline historical repository to retrieve the most similar reference sample to the current sparse observed data. By integrating the observed data and the reference sample, the time-series interpolation module achieves accurate reconstruction of the raw data. Validation on a HIL testbed, which emulates an end-edge link, shows that RFDR improves throughput by 17.3% and reduces median latency by 1.81 s compared to the retransmission scheme, while also

outperforming deep learning-based baselines in reconstruction accuracy under 35% to 85% random data missing rates. These results demonstrate its superior performance under severe data missing conditions and its strong potential for emergency scenarios where low latency and high reliability are critical.

Acknowledgements This work was supported in part by Natural Science Foundation of Beijing (Grant No. 62173158), National Natural Science Foundation of China (Grant Nos. 62222121, 62271447), and Zhejiang Provincial Natural Science Foundation of China (Grant No. LR25F010003).

References

1 Wang Y, Yang C, Lan S, et al. End-edge-cloud collaborative computing for deep learning: a comprehensive survey. IEEE Commun Surv Tutorials, 2024, 26: 2647–2683
2 Zhou Y, Jin Z, Shi H, et al. Enhanced emergency communication services for post-disaster rescue: multi-IRS assisted air-ground integrated data collection. IEEE Trans Netw Sci Eng, 2024, 11: 4651–4664
3 Cui J X, Wang Q P, Sun B Y, et al. Learning continuous network emerging dynamics from scarce observations via data-adaptive stochastic processes. Sci China Inf Sci, 2024, 67: 222206
4 Xiao Y, Wang E H, Chen Y F, et al. Communication-centric integrated sensing and communications with mixed fields. Sci China Inf Sci, 2024, 67: 219301
5 Wang C, Rahman A, Durugkar I, et al. N-agent ad hoc teamwork. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 111832–111862
6 Li C, Wang J, Wang S, et al. A review of IoT applications in healthcare. Neurocomputing, 2024, 565: 127017
7 Wang R, Qi J, Chen L, et al. Survey of collaborative inference for edge intelligence. J Comput Res Dev, 2023, 60: 398–414
8 Xu X Z, Suo Y X, Zhao Y, et al. A dry-electrode enabled ECG-on-Chip with arrhythmia-aware data transmission. Sci China Inf Sci, 2025, 68: 122405
9 Mu J, Yang R, Li M, et al. Design and optimization of cooperative MAC protocol for mobile ad hoc network-enabled train-to-train communications. IEEE Trans Veh Technol, 2025, 74: 7278–7291
10 Zhang H, Cui Z, Huang S, et al. QM-RGNN: an efficient online QoS measurement framework with sparse matrix imputation for distributed edge clouds. In: Proceedings of IEEE Conference on Computer Communications, Vancouver, 2024. 191–200
11 Chernozhukov V, Hansen C, Liao Y, et al. Inference for low-rank models. Ann Statist, 2023, 51: 1309–1330
12 Adhikari D, Jiang W, Zhan J, et al. A comprehensive survey on imputation of missing data in Internet of Things. ACM Comput Surv, 2022, 55: 1–38
13 Li X, Xu L D. A review of Internet of Things—resource allocation. IEEE Internet Things J, 2021, 8: 8657–8666
14 Luo Q, Hu S, Li C, et al. Resource scheduling in edge computing: a survey. IEEE Commun Surv Tut, 2021, 23: 2131–2165
15 Hu S, Shi W, Li G. CEC: A containerized edge computing framework for dynamic resource provisioning. IEEE Trans Mobile Comput, 2023, 22: 3840–3854
16 He Q, Wang Y, Wang X, et al. Routing optimization with deep reinforcement learning in knowledge defined networking. IEEE Trans Mobile Comput, 2024, 23: 1444–1455
17 Sun Y, Wu Z, Meng K, et al. Vehicular task offloading and job scheduling method based on cloud-edge computing. IEEE Trans Intell Transp Syst, 2023, 24: 14651–14662
18 Yun J, Eryilmaz A, Moon J, et al. Remote estimation for dynamic IoT sources under sublinear communication costs. IEEE ACM Trans Netwing, 2024, 32: 1333–1345
19 Zlmmerling M, Ferrari F, Mottola L, et al. pTUNES: runtime parameter adaptation for low-power mac protocols. In: Proceedings of the 11th International Conference on Information Processing in Sensor Networks, Beijing, 2012. 173–184
20 Gao W, Zhao Z, Min G. Adaplora: resource adaptation for maximizing network lifetime in LoRa networks. In: Proceedings of the 28th International Conference on Network Protocols (ICNP), Madrid, 2020. 1–11
21 Lai Z, Li H, Wang Y, et al. Achieving resilient and performance-guaranteed routing in space-terrestrial integrated networks. In: Proceedings of IEEE Conference on Computer Communications, New York, 2023. 1–10
22 Wang P, Sourav S, Li H, et al. One pass is sufficient: a solver for minimizing data delivery time over time-varying networks. In: Proceedings of IEEE Conference on Computer Communications, New York, 2023. 1–10
23 Chen Q, Wang K, Guo S, et al. Latency-optimal pyramid-based joint communication and computation scheduling for distributed edge computing. In: Proceedings of IEEE Conference on Computer Communications, New York, 2023. 1–10
24 Li Y, Zeng T, Zhang X, et al. Tapfinger: task placement and fine-grained resource allocation for edge machine learning. In: Proceedings of IEEE Conference on Computer Communications, New York, 2023. 1–10
25 Cao W, Wang D, Li J, et al. Brits: bidirectional recurrent imputation for time series. In: Proceedings of Advances in Neural Information Processing Systems, Montréal, 2018
26 Wei X, Zhang Y, Wang S, et al. Self-attention graph convolution imputation network for spatio-temporal traffic data. IEEE Trans Intell Transp Syst, 2024, 25: 19549–19562
27 Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: Proceedings of International Conference on Learning Representations (ICLR'18), 2018
28 Wang D, Yan Y, Qiu R, et al. Networked time series imputation via position-aware graph enhanced variational autoencoders. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, 2023. 2256–2268
29 Diao Z, Wang X, Zhang D, et al. DMSTG: dynamic multiview spatio-temporal networks for traffic forecasting. IEEE Trans Mobile Comput, 2024, 23: 6865–6880
30 Wang H, Li H, Chen X, et al. Optimal transport for time series imputation. In: Proceedings of the 13th International Conference on Learning Representations, 2025
31 Yang S, Dong M, Wang Y, et al. Adversarial recurrent time series imputation. IEEE Trans Neural Netw Learn Syst, 2020, 34: 1639–1650
32 Luo Y, Zhang Y, Cai X, et al. E2gan: end-to-end generative adversarial network for multivariate time series imputation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019. 3094–3100
33 Liu Y, Yu R, Zheng S, et al. Naomi: non-autoregressive multiresolution sequence imputation. In: Proceedings of Advances in Neural Information Processing Systems, 2019
34 Liu D, Wang Y, Liu C, et al. Scope-free global multi-condition-aware industrial missing data imputation framework via diffusion transformer. IEEE Trans Knowl Data Eng, 2024, 36: 6977–6988
35 Zhuang Z, Zhang Y, Wang X, et al. Time-varying lora: towards effective cross-domain fine-tuning of diffusion models. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 37: 73920–73951
36 Goswami M, Szafer K, Choudhry A, et al. Moment: a family of open time-series foundation models. In: Proceedings of ICML'24, 2024
37 Liu Y, Zhang H, Li C, et al. Timer: generative pre-trained transformers are large time series models. In: Proceedings of the 41st International Conference on Machine Learning, 2024
38 Wang S, Li J, Shi X, et al. Timemixer++: a general time series pattern machine for universal predictive analysis. In: Proceedings of International Conference on Learning Representations (ICLR'25), 2025
39 Jia F, Wang K, Zheng Y, et al. GPT4MTS: prompt-based large language model for multimodal time-series forecasting. AAAI, 2024, 38: 23343–23351

40 Tan M, Merrill M A, Gupta V, et al. Are language models actually useful for time series forecasting? In: Proceedings of Advances in Neural Information Processing Systems, 2024. 60162–60191
41 Liu Y, Qin G, Huang X, et al. Autotimes: autoregressive time series forecasters via large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 122154–122184
42 Liu C, Xu Q, Miao H, et al. TimeCMA: towards LLM-empowered multivariate time series forecasting via cross-modality alignment. AAAI, 2025, 39: 18780–18788
43 Wu H, Hu T, Liu Y, et al. Timesnet: temporal 2D-variation modeling for general time series analysis. In: Proceedings of International Conference on Learning Representations, 2023
44 Ji M, Zhang X, Li X. Ssgan: image generation from freehand scene sketches. In: Proceedings of the 7th International Conference on Electronic Technology and Information Science, 2022. 1–5
45 Wang L, Koniusz P. Uncertainty-dtw for time series and sequences. In: Proceedings of Computer Vision – ECCV 2022. Cham: Springer Nature Switzerland, 2022. 176–195
46 Johnson A E, Pollard T J, Shen L, et al. MIMIC-III, a freely accessible critical care database. Sci Data, 2016, 3: 1–9
47 Zheng Y, Yi X, Li M, et al. Forecasting fine-grained air quality based on big data. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2015. 2267–2276
48 Cini A, Marisca I, Alippi C. Filling the G_ap_s: multivariate time series imputation by graph neural networks. In: Proceedings of International Conference on Learning Representations, 2022
49 Zhang X, Zeman M, Tsiligkaridis T, et al. Graph-guided network for irregularly sampled multivariate time series. In: Proceedings of International Conference on Learning Representations, 2022