CODE$^+$: Fast and Accurate Inference for Compact Distributed IoT Data Collection

Huali Lu, *Graduate Student Member, IEEE*, Feng Lyu, *Senior Member, IEEE*, Ju Ren, *Senior Member, IEEE*, Huaqing Wu, *Member, IEEE*, Conghao Zhou, *Member, IEEE*, Zhongyuan Liu, Yaoxue Zhang, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—In distributed IoT data systems, full-size data collection is impractical due to the energy constraints and large system scales. Our previous work has investigated the advantages of integrating matrix sampling and inference for compact distributed IoT data collection, to minimize the data collection cost while guaranteeing the data benefits. This paper further advances the technology by boosting fast and accurate inference for those distributed IoT data systems that are sensitive to computation time, training stability, and inference accuracy. Particularly, we propose *CODE*$^+$, i.e., Compact Distributed IOT Data CollEction Plus, which features a cluster-based sampling module and a Convolutional Neural Network (CNN)-Transformer Autoencoders-based inference module, to reduce cost and guarantee the data benefits. The sampling component employs a cluster-based matrix sampling approach, in which data clustering is first conducted and then a two-step sampling is performed in accordance with the number of clusters and clustering errors. The inference component integrates a CNN-Transformer Autoencoders-based matrix inference model to estimate the full-size spatio-temporal data matrix, which consists of a CNN-Transformer encoder that extracts the underlying features from the sampled data matrix and a lightweight decoder that maps the learned latent features back to the original full-size data matrix. We implement *CODE*$^+$ under three operational large-scale IoT systems and one synthetic Gaussian distribution dataset, and extensive experiments are provided to demonstrate its efficiency and robustness. With a 20% sampling ratio, *CODE*$^+$ achieves an average data reconstruction accuracy of 94% across four datasets, outperforming our previous version of 87% and state-of-the-art baseline of 71%.**

***Index Terms*—Accurate data inference, CNN- transformer, compact distributed data collection, spatio-temporal data sampling.**

Received 29 November 2023; revised 28 June 2024; accepted 27 August 2024. Date of publication 3 September 2024; date of current version 16 September 2024. This work was supported in part by the National Key Research and Development (K&D) Program of China under Grant 2022YFF0604504, in part by the National Natural Science Foundation of China under Grant 62422216, Grant 62320106006, Grant 62122095, Grant 62432004, and Grant 62072472, in part by 111 Project under Grant B18059, in part by Central South University Innovation-Driven Research Program under Grant 2023CXQD029, and in part by a grant from the Guoqiang Institute, Tsinghua University. Recommended for acceptance by S. Wang. *(Corresponding author: Feng Lyu.)*

Huali Lu and Feng Lyu are with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: huali_lu@csu.edu.cn; fenglyu@csu.edu.cn).

Ju Ren and Yaoxue Zhang are with the Department of Computer Science and Technology, BNRist, Tsinghua University, Beijing 100084, China, and also with Zhongguancun Laboratory, Beijing 100094, China (e-mail: renju@tsinghua.edu.cn; zhangyx@tsinghua.edu.cn).

Huaqing Wu is with the Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada (e-mail: huaqing.wu1@ucalgary.ca).

Conghao Zhou and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: c89zhou@uwaterloo.ca; sshen@uwaterloo.ca).

Zhongyuan Liu is with the High School Affiliated to Renmin University of China, Beijing 100080, China (e-mail: zyliu121@163.com).

Digital Object Identifier 10.1109/TPDS.2024.3453607

I. Introduction

As ONE of the most important impetuses to push forward the construction of smart city, large-scale distributed Internet of Things (IoT) systems have been widely deployed permeating the fields of environment monitoring, transportation, communication, and more [2], [3], [4]. Over time, the distributed IoT systems can collect extensive data in terms of system operation, user profiles, targeted status, etc., based on which a lot of intelligent services to government, business, and users, can be provisioned by leveraging the advanced technologies of Big Data and deep learning [5], [6], [7], [8]. However, collecting and storing the large-scale IoT data is costly and usually prohibited when considering the system efficiency and robustness. On the one hand, as the system scale increases, the data size grows explosively, posing significant communication and storage overhead for data collection. On the other hand, the distributed IoT sensors are usually power- and communication-restrained, being prohibited to support full-size data collection in the long run [9]. Besides, the IoT data may have underlying spatio-temporal correlations, and the full-size data collection can result in information redundancy, restraining the system efficiency. Therefore, how to reduce the data collection cost without losing data benefits has become an urgent yet challenging problem for ubiquitous distributed IoT systems.

To deal with the issue, sparse sensing can be leveraged to reduce the data collection cost, via which the system can infer the missing data based on partial sampled data [10]. As only partial data are collected, the communication and storage cost can be largely reduced. However, for most IoT applications, such as trajectory prediction [11], anomaly detection [5], and evolution analysis [12], they are highly dependent on the full-size data, and thus the accuracy of data completion/inference/reconstruction becomes crucial for the system. Therefore, in recent years, the technologies of compressive sensing (CS) [13], [14], matrix completion (MC) [15], and tensor completion (TC) [16], have received extensive attention, which can be applied to infer the missing data from the low-rank feature data. However, the

existing sparse sensing technologies mainly focus on designing efficient algorithms to infer the missing data, often assuming that the incomplete data already exists or using the random/uniform sampling approach. These methods cannot work well in general since the sampling process can affect the inference performance significantly. Particularly, random/uniform sampling approaches can perform well for Gaussian distribution data since each data matrix location carries the same amount of information. Unfortunately, they can lose efficiency rapidly when there are information differences among data matrix locations for non-Gaussian distribution data, which has been verified by our pilot experiments. On the other hand, the conventional sparse sensing technologies mainly rely on the data linearity correlations to infer the missing data, which restrains the inference performance without capturing the non-linearity features for complicated IoT data structures.

To bridge this gap, we study the deeply coupled problems of matrix[1] sampling and inference, to enable compact distributed IoT data collection without losing the data benefits, which is challenging for the following reasons. First, given a target matrix, how much data should be sampled is unknown ahead for the system. On the one hand, our goal is to sample as little data as possible to save the data collection cost. On the other hand, without sufficient sampling ratio, regardless of the sampling strategy employed, the required data features that reflect the overall data picture cannot be guaranteed, leading to the matrix recovering failures. Second, given the amount of samples, how to determine their locations within the matrix is another challenge since the sampling quality can affect the data inference performance directly, especially for the non-Gaussian distribution data. Third, with the sampled incomplete matrix, how to design the matrix inference model that can capture both the linearity and non-linearity correlations becomes crucial.

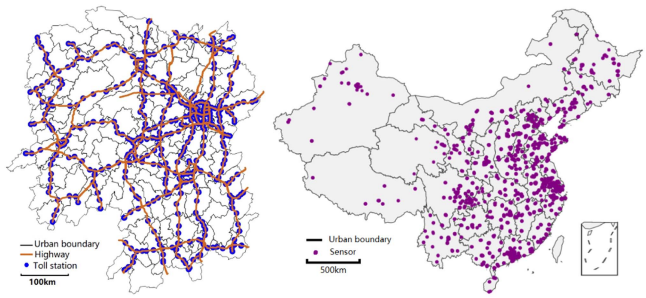
To tackle the above challenges, in our previous work, we have demonstrated the superiority of integrating the matrix sampling and inference for compact distributed IoT data collection. In this paper, we further optimize the technology by boosting fast and accurate inference for those distributed IoT data systems that are sensitive to computation time, training stability and inference accuracy. Particularly, we first conduct an empirical study on typical distributed IoT data systems by disclosing their low-rank features, based on which the matrix completion theory can be leveraged to determine the minimum amount of samples. In accordance with the samples amount constraint, we then propose a systematical framework, named *CODE*$^+$, i.e., Compact Distributed IOT Data CollEction Plus, which consists of two major components, i.e., matrix sampling and matrix inference, to determine sampling locations and conduct matrix inference, respectively. In the matrix sampling component, given the amount of samples $|\Omega|$ to be collected, we devise a cluster-based sampling approach. Particularly, for training matrix samples, the sensing data of each row (in spatial domain) are first clustered based on data values, resulting in a total of K clusters. Then a two-step scheduling process is performed to determine the respective K and $|\Omega| - K$ locations according to the number of clusters and clustering errors. In the matrix inference component, we devise a Convolutional Neural Network (CNN)-Transformer Autoencoder to output the estimated matrix by learning the spatio-temporal correlations of data. Specifically, a CNN-Transformer encoder is designed to extract the underlying features of the sampled matrix, and a lightweight decoder is included to map learned latent features back to original full-size data matrix. These two neural networks are trained simultaneously to minimize data inference loss, which can be used to conduct matrix inference. The merits of *CODE*$^+$ are two-folds: 1) the cluster-based sampling approach can keep pace with the underlying data distribution by sampling data with greater diversity, increased information content, and higher data benefit; 2) the CNN-Transformer Autoencoders-based model can unleash the full potential of spatio-temporal correlations among data for matrix inference, collectively contributing to a superior performance.

The main contributions are summarized as follows.

- We study the deeply coupled matrix sampling and inference problem, both of them are valuable to data collection performance in large-scale distributed IoT data systems. Our investigation can effectively address the one-dimensional control and potential approach defect issues in existing works.
- We conduct an empirical study on typical IoT system datasets, and disclose insightful observations, such as long-tailed distributions and low-rank features, which direct our solutions to address the considered data collection problem.
- To address the challenges, we propose a fast and accurate compact data collection framework, *CODE*$^+$, which includes two key technical components: 1) cluster-based matrix sampling to identify high-information-content sampling locations, and 2) CNN-Transformer Autoencoders-based matrix inference for rapid and precise data reconstruction.
- Extensive experiments have been conducted to demonstrate the effectiveness and efficiency of our proposed *CODE*$^+$ framework. Using three operational large-scale distributed IoT systems and one synthetic Gaussian distribution dataset, we compare *CODE*$^+$ with four peer baselines. The experimental results indicate that *CODE*$^+$ outperforms state-of-the-art baselines in terms of data inference accuracy, data distribution fidelity, and operational efficiency. Furthermore, *CODE*$^+$ can maintain superior performance across different distributed IoT data systems.

The remainder of this paper is organized as follows. The system description and problem definition are given in Section II. We conduct an empirical data analytics in Section III, and elaborate on the design of *CODE*$^+$ in Section IV. Extensive experiments are conducted in Section V, and the related work is reviewed in Section VI. Finally, we conclude the paper in Section VII.

[1]For distributed IoT systems, the data is usually managed in matrix format with temporal and spatial domains.



(a) Provincial ETC system (b) National air quality sensors

Fig. 1. Typical large-scale IoT data systems.

II. System Description and Problem Definition

In this section, we first describe the large-scale distributed IoT data systems, and then define the data collection problem. Finally, we highlight technical challenges that have to be carefully addressed when tackling the defined problem.

A. Large-Scale Distributed IoT Data Systems

In the Big Data era, extensive physical entities are managed by information systems, where large-scale distributed IoT sensors are deployed to collect entity information, monitor the system environment, and act as server providers for nearby users [4], [17], [18]. Examples include transportation ETC (Electronic Toll Collection) systems, air quality sensor networks, cellular communication networks, IoT monitoring systems, etc. By collecting the system data, extensive intelligent services can be enabled based on advanced techniques of Big Data and deep learning. This also enables the government to achieve precise governance capability with system overview, the service providers to deliver cost-efficient services with efficient resource utilization, and the users to enjoy high-quality services with personalization [19], [20], [21]. Fig. 1 shows two typical large-scale distributed IoT data systems, i.e., provincial highway ETC system and national air quality sensor networks, which are two fundamental building blocks for a smart city.[2] Particularly, Fig. 1(a) shows a provincial highway ETC system, in which there are 465 toll stations across $6,800$ km highway roads covering 90 districts in the province. Normally, the daily transition traffic in the system can reach 1.6 millions, and the moving states of these vehicles have to be collected by the system. Fig. 1(b) shows a national air quality sensor network, which is used to monitor the air quality around the whole country. There are about $1,518$ sensors located in 375 cities. Each sensor needs to report the air quality value to the sink node on an hourly basis. Considering the system scales and energy constraints of sensors, full-size data collection becomes the significant bottleneck for efficiency and robustness [22], [23].

[2]Note that, the considered problem and proposed solutions in this paper can be readily applied in general large-scale distributed IoT data systems, where the exemplary scenarios are used to carry out specific verification.

B. Problem Definition

We consider a general data collection system, where each sensor has to collect the sensing data and deliver the data to one sink every time slot. Suppose there are a set of n sensors, $\{s_1, s_2, \ldots, s_n\}$, and m time slots, $\{t_1, t_2, \ldots, t_m\}$, where the length of each time slot can be set in accordance with the system management/granularity requirements. Then, the collected data can be represented by an $n \times m$ matrix $X \in \mathbb{R}^{n \times m}$, and each entry x_{ij} indicates the monitoring data from the sensor i at the time slot j. For instance, in the above mentioned highway ETC system, the row indicates different toll stations and the column indicates different time slots, where each entry in the matrix is a recorded number of traffic flow passing through the station within the time slot. For n sensors in the distributed IoT data system, instead of making each sensor periodically collect and report data to the sink, only a subset of sensors are scheduled to perform the sensing data collection in each time slot.

It is worth noting that the spatio-temporal data usually has similarity among neighboring locations and periodicity among time slots, which has been verified by our pilot experiments. Therefore, we can sample partial data in a systematic way to reduce the data collection cost, and infer other empty data based on them to recover the full data picture. If there is no measurement data for a location at a time slot, the corresponding entry in X is set to be empty. All observed entries are denoted by $\Omega = \{i, j \mid x_{ij} \text{ is known}\}$, which forms an incomplete data sample matrix X_Ω. If the set Ω contains enough data utility, we can use the incomplete data sample matrix X_Ω to reconstruct the complete data matrix $\hat{X}$. The primary challenge addressed in this paper is minimizing data collection costs while preserving data utility, which is a coupled matrix sampling and inference problem. This involves two key questions: 1) How should we calculate the cost of sampling data points? 2) How can we reflect and guarantee the utility of the data? For the first question, we simplify the cost calculation by assuming that the cost of sampling each data point is uniform. Thus, the total cost is directly proportional to the number of sampled data points. For the second question, we quantify data utility by the reconstruction error of the unsampled data, which depends on quality of sampled data and quantity of sampled data. To ensure high data utility while minimizing costs, the focus is on improving the quality of the sampled data. Based on these considerations, we define the problem as a cost minimization problem with a constraint on acceptable reconstruction error:

$$\min_{\{\Omega\}} c|\Omega|$$
$$\text{subject to } \frac{1}{|\bar{\Omega}|} \sum_{\bar{\Omega}} \left| (X - \hat{X}) \star X \right| \leq \theta, \tag{1}$$

where c represents the cost of sampling each data point, and Ω, $\bar{\Omega}$, $|\Omega|$, and $|\bar{\Omega}|$ denote the sets of sampled and unsampled data points, and the number of sampled and unsampled data points, respectively. The term $\frac{1}{|\bar{\Omega}|} \sum_{\bar{\Omega}} |(X - \hat{X}) \star X|$ calculates the mean absolute percentage error (MAPE), indicating the relative error between the reconstructed matrix $\hat{X}$ and the true matrix X for unsampled data. Here, $\star$ denotes element-wise division

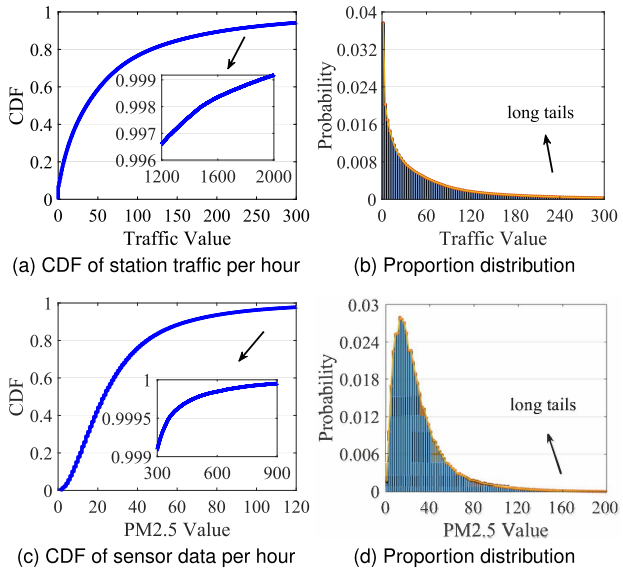


Fig. 2. Long-tailed distribution of observed values.

of the matrices, and θ is the maximum acceptable relative error for the system.

C. Problem Challenges

1) *Determining the Amount of Sampling Data:* To reduce the sampling cost, the first issue is to consider how much data are sufficient to recover the required data picture, that is, how to determine the minimum amount of data (i.e., $|\Omega|$) to be sampled. Based on $|\Omega|$, the system should be able to infer the missing data with an acceptable inference error.
2) *Determining the Locations of Sampling Data:* The amount of samples determines whether the unsampled data can be accurately reconstructed, while the sampling locations (i.e., Ω) determine the quality of sampled data, which can also affect the inference accuracy significantly.
3) *Reconstructing the Unsampled Data:* Sampling can reduce the data collection cost, but almost all data-based services rely on the entire data (i.e., $\hat{X}$). Therefore, how to design the inference model to recover the full-size data is another pivotal building block.

III. Empirical Study on IoT System Data

A. Long-Tailed Distribution of Observed Values

In actual distributed IoT data systems, the distributions of data often differ from ideal assumptions. We reveal this phenomenon by analyzing the data of two typical distributed IoT data systems. In the highway ETC system, each station will record the number of transition vehicles within each time slot. We first investigate the distribution of traffic value, where the time slot is set to be one hour and we adopt the one-year data dating from Jan. 1st, 2019 to Dec. 31th, 2019. Fig. 2(a) shows the cumulative distribution function (CDF) of station traffic, and we can achieve the following two major observations. First, the traffic value can vary widely, e.g., being as small as 0 and as large as 2,000, which can enlarge the information space and pose challenges to data sampling and inference. Second, the observed traffic values follows a clear long-tailed distribution. Particularly, 80 and 90 percents of traffic values are limited within about 110 and 200, respectively, while the remaining 10% of the traffic values can span up to 2,000. Fig. 2(b) shows the proportion of station traffic, which cross-verifies the long-tailed phenomenon. For instance, when the traffic value is larger than 180, the proportion becomes quite small, composing the long tails. In addition, we can observe that the proportions of traffic values are uneven, which generally decreases with the traffic value.

The same phenomenon appears in air quality monitoring data. For the air quality sensor network, each sensor needs to report data every hour. The data contains the concentration of many different pollutants (e.g., PM2.5, PM10, SO_2, and NO_2), together with some meteorological logs (e.g., rainfall, wind speed, and temperature) collected within the country. Among them, the primary pollutant of air quality is PM2.5, and thus we employ its values as the target data. Particularly, we adopt nearly one-year data dating from Jan. 11th to Dec. 11th, 2021 to observe the overall distribution of the sensor data. As shown in Fig. 2(c) and (d), the national air quality sensor data also follows a widely distributed and long-tailed pattern. Both pilot experiments reveal that in real distributed IoT data systems, the data distribution typically deviates from a perfect Gaussian distribution. Therefore, when sampling the same amount of data, the traditional random/uniform approaches cannot well capture the data diversity information.

B. Low-Rank Feature

In this subsection, we examine the low-rank features of the previous two typical IoT datasets. The low-rank feature is the prerequisite to adopt the sparse sampling approach, as without it, the unsampled data cannot be accurately recovered. Particularly, we apply the singular value decomposition (SVD) approach to strictly examine whether the data matrix has a low-rank structure. Particularly, the traffic matrix $X_{n\times m}$ can be decomposed as

$$X = U\Sigma V^T, \tag{2}$$

where $U = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \vdots & \vdots \\ u_{n1} & \cdots & u_{nn} \end{bmatrix}$, $V = \begin{bmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \vdots & \vdots \\ v_{m1} & \cdots & v_{mm} \end{bmatrix}$ are the $n \times n$ and $m \times m$ unitary matrix, respectively. In addition, Σ is an $n \times m$ diagonal matrix with the diagonal elements (i.e., the singular values) in descending order, i.e., $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$. Denote by r the rank of matrix X, which is equivalent to the number of its non-zero singular values. A matrix is low-rank if it holds that $r \ll \min\{n, m\}$. Note that, the above calculation is defined for precise rank, which can be ill-posed for practical data when there exists arbitrary and small perturbations of matrix elements, containing limited data features but affecting the rank calculation significantly. Therefore, instead of calculating the precise rank, we adopt the approximate rank, which is used in

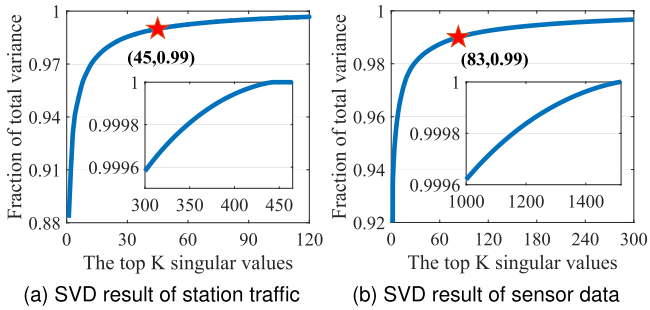


Fig. 3. Low-rank feature of distributed IoT system data.

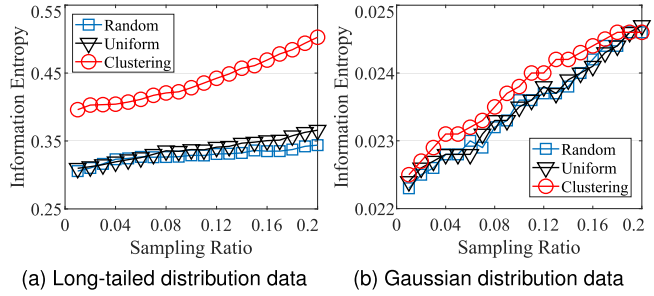


Fig. 4. Pilot sampling experiments.

the literature [24]. Particularly, the matrix X is claimed to have ω-rank k if

$$\inf\{\|X - Y\| : Y \quad \text{has} \quad \text{rank} \quad k\} \leq \omega. \tag{3}$$

where $\|\cdot\|$ represents norm operation used to calculate the distance between X and Y (Y is any matrix with rank k), inf represents the operation of calculating the lower bound, and ω is the upper bound.

The problem of finding an approximate rank k can be formulated in a definite manner, i.e., finding one matrix X_k with rank k, such that there is no other matrix of rank k whose distance from X is less than the distance from X_k to X. A theorem provided by Eckart and Young [25] proves that the error of approximating a matrix X by X_k satisfies $\|X - X_k\|_F^2 \leq \|X - Y\|_F^2$, where Y is any matrix with rank k, and X_k is the truncated SVD of the matrix X with the ω-rank k, i.e., $X_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$, where u_i and v_i are column vectors of U and V, respectively. The ratio $g(k) = \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{r} \sigma_i^2$ represents the fraction of total variance (Frobenius norm) in X that is explained by its approximated matrix with ω-rank k, i.e., X_k. According to the Principal Components Analysis (PCA), if a matrix has low-rank, its top k singular values should occupy the nearly total variance, i.e., $\sum_{i=1}^{k} \sigma_i^2 \approx \sum_{i=1}^{r} \sigma_i^2$. Fig. 3(a) shows the fraction of total variance captured by the top k singular values for traffic data, where we adopt the one year data, i.e., $n = 465$ and $m = 8760$. Fig. 3(b) also shows the fraction of total variance captured by the top k singular values for air quality monitoring data, where $n = 1518$ and $m = 8040$. We can observe that large fractions of total variance can be covered by a few top singular values. Particularly, the respective top 45 and 83 singular values can capture 99% of total variance for the traffic and sensor datasets, indicating that the data matrix usually has an approximate low-rank structure in practice.

According to the matrix completion theory [26], for most matrix $X \in \mathbb{R}^{n \times m}$ with low rank r, if a subset of its entries $X_{ij}, (i,j) \in \Omega$ is known ahead, it can be perfectly recovered by solving the following convex optimization problem,

$$\min_{\{\Omega\}} \|\hat{X}\|_* \quad \text{subject to } \hat{X}_{ij} = X_{ij},\ (i,j) \in \Omega, \tag{4}$$

where $\|\hat{X}\|_*$ is the nuclear norm of the matrix $\hat{X}$. It is the sum of its singular values, i.e., $\|\hat{X}\|_* = \sum_{i=1}^{\min\{n,m\}} \hat{\sigma}_i$, where $\hat{\sigma}_i \geqslant 0$ are the singular values of $\hat{X}$. However, there is a prerequisite for $|\Omega|$ (the amount of sampling data), which should satisfy

$$|\Omega| \geq C g^{6/5} r \log g, \tag{5}$$

where C is a numerical constant and $g = \max\{n, m\}$. Therefore, the principle of (5) [26] can be adopted to guide the determination of the amount of sampling data.

C. Takeaways

With the empirical study, we can conclude that: 1) the IoT data matrix usually has a low-rank feature, which is the fundamental for sampling; 2) when the IoT data has uneven distribution, the traditional random/uniform sampling approach may lose efficiency [27], [28], [29], calling for more efficient methods; and 3) for low-rank matrices, the matrix completion theory can provide the principle for determining $|\Omega|$, but the matrix factorization based inference approach can only extract the linear features among data, leaving optimization room for complicated data structure with non-linear features. Inspired by these insights, in what follows, we propose *CODE*$^+$, which consists of sampling and inference components, to determine the locations of $|\Omega|$ and estimate the matrix $\hat{X}$, collectively optimizing the data collection problem in a fast and stable manner.

IV. Design of *CODE*$^+$

A. Cluster-Based Matrix Sampling

Cluster-Based Sampling Matters: In distributed IoT systems, similarity in the observed data values usually implies that their locations have the underlying correlations. To reduce the data redundancy, we propose to cluster the observed values, since data values in the same cluster are more likely to be correlated, which can be leveraged to guide the location determination. To verify the effectiveness of the clustering method, we then conduct pilot experiments on both Gaussian and non-Gaussian distribution data. Fig. 4 shows the information entropy achieved by different sampling approaches, using the long-tailed distribution data mentioned above, and one Gaussian distribution data in [30]. With the clustering approach, the samples are distributed evenly for each cluster. We can observe that for non-Gaussian distribution data, there is a significant performance gap between the clustering and uniform/random approaches, while for Gaussian distribution data, their corresponding entropy

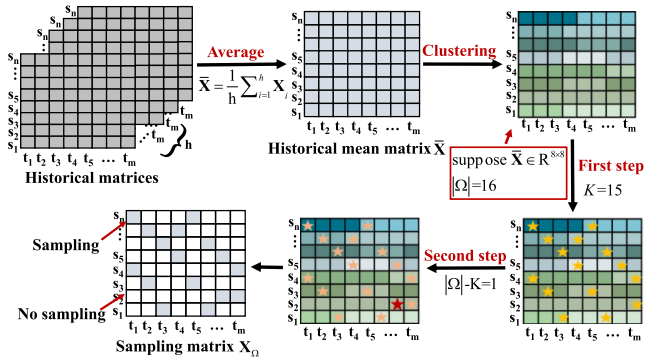


Fig. 5. Sampling scheduling process.

values are quite close. However, in most distributed IoT data systems, the observed data values do not follow perfect uniform or Gaussian distributions. Therefore, in what follows, we cast our cluster-based matrix sampling approach.

The Two-step Sampling Approach: As shown in Fig. 5, the sampling approach works as follows. Given a set of historical $n \times m$ matrices $X_1, X_2, \ldots, X_h$, we first calculate the historical mean matrix $\bar{X} = \frac{1}{h}\sum_{i=1}^{h} X_i$. For the matrix $\bar{X}$, we then conduct the k-means clustering for each row values, i.e., $\{\bar{X}_{i,1}, \bar{X}_{i,2}, \ldots, \bar{X}_{i,m}\}$. Let k_i denote the number of clusters which is independently learned by each row according to its own data distribution, and $g^i = \{g_1^i, g_2^i, \ldots, g_p^i, \ldots, g_{k_i}^i\}$ denote the set of clusters for the i-th row. The $|\Omega|$ locations are determined by the following two-step sampling. In the first step, for each cluster in each row, we randomly choose one sample, and thus a total of $K = \sum_{i=1}^{n} k_i$ samples are selected. For the remaining $|\Omega| - K$ samples, we distribute them according to the cluster error of each cluster, since the larger the clustering error indicates higher diversity of the data within the cluster, calling for more samples. Particularly, all clusters (i.e., $\forall g_p^i$) in the list are ranked by their clustering errors in descending order. Then, one sample is assigned to the first cluster with the largest error, and the cluster is then deleted from the list. The process repeats until all $|\Omega|$ samples are assigned. Note that, during the two-step sampling, when determining the specific (i, j) for one sample in the cluster, the sampling frequency of the time slot index j is used. Specifically, we maintain a list $\{q_1, q_2, \ldots, q_j, \ldots, q_m\}$ to record the frequency with which data in time slot j has been sampled. If one element $\bar{X}_{i,j}$ in g_p^i is sampled, then the count of q_j adds one. After that, when to determine a new sample (i, j) in the cluster g_p^i, the element (i, j') is selected if $q_{j'}$ is the smallest count in the candidate elements. Fig. 5 shows an example of two-step sampling approach with $|\Omega| = 16$ and $K = 15$. Specifically, the first step is to determine the first 15 locations in accordance with each cluster, and the second step is to determine the remaining 1 location based on the clustering errors.

B. CNN-Transformer Autoencoders-Based Matrix Inference

In this subsection, we introduce a novel CNN-Transformer autoencoders model to capture spatio-temporal data correlations for data inference. Specifically, the model consists of a Pre-Completion module to preprocess the inputs for improving of speed and accuracy, a CNN-Transformer encoder to map the sampling samples to a latent representation, and a lightweight decoder to reconstruct the full samples from the latent representation. Fig. 6 shows the overall architecture of the designed CNN-Transformer Autoencoders.

Pre-Completion Matrix Input: In applications sensitive to data volume, like deep learning-based algorithms, performance correlates with the available data. Consequently, the data inference component aims to swiftly and accurately reconstruct full-sized data based on sparse samples. However, the incomplete sampling matrix (i.e., X_Ω) contains minimal information due to missing of most data. Despite this, the clustering information from the previous data sampling component offers valuable insights that are not evident in the incomplete matrix. Drawing inspiration from this, we introduce a data preprocessing module to enhance the input matrix. By pre-completing the matrix, we inject more effective information into it. Therefore, instead of using the incomplete sampling matrix as input, our CNN-Transformer autoencoders model takes the pre-completed sampling matrix (i.e., $X_s \in \mathbb{R}^{n \times m}$) as the model's input so as to reduce the predictive uncertainty and increase reconstruction efficiency. For empty-data locations, we conduct the pre-completion operation based on the clustering results. Particularly, within each cluster, the empty data is replaced by the empirical mean value of the sampled data instead of zero. Although the mean value may not be identical to the true value, it can reflect the basic feature of the original value to some extent, which is more effective than starting training from the value of zero. The pre-completion operation can reduce the training time effectively and improve the inference accuracy accordingly.

CNN-Transformer Autoencoders: To reconstruct the original IoT collection data, we devise a novel autoencoder model with an asymmetric encoder-decoder design, which is proficient in dealing with natural data with heavy spatial and temporal redundancy in aspects of accuracy, speed and stability. Unlike classical autoencoders, we adopt an asymmetric design consisting of two sub-modules, i.e., a CNN-Transformer encoder and a lightweight decoder, for feature extraction from local to global scales and for full-size original data inference. In what follows, we will elaborate on each individual component.

CNN-Transformer Encoder: Similar to image data, the data matrix collected by the distributed IoT data system also has heavy data redundancy, e.g., an unsampled data point can be recovered from neighboring points with little high-level understanding of whole network. Therefore, driven by the huge success of transformer in images, we attempt to address our data inference problem with the transformer structure, which has strong global inductive modeling capability, and its attention structure can effectively extract the correlations between features. However, transformers, despite their success, tend to underperform compared to CNNs of similar size when the training dataset is limited. This is because transformers lack certain inductive biases, which are essentially prior knowledge and assumptions embedded within models. Although CNNs possess two key inductive biases, locality (i.e., similar regions in low-rank

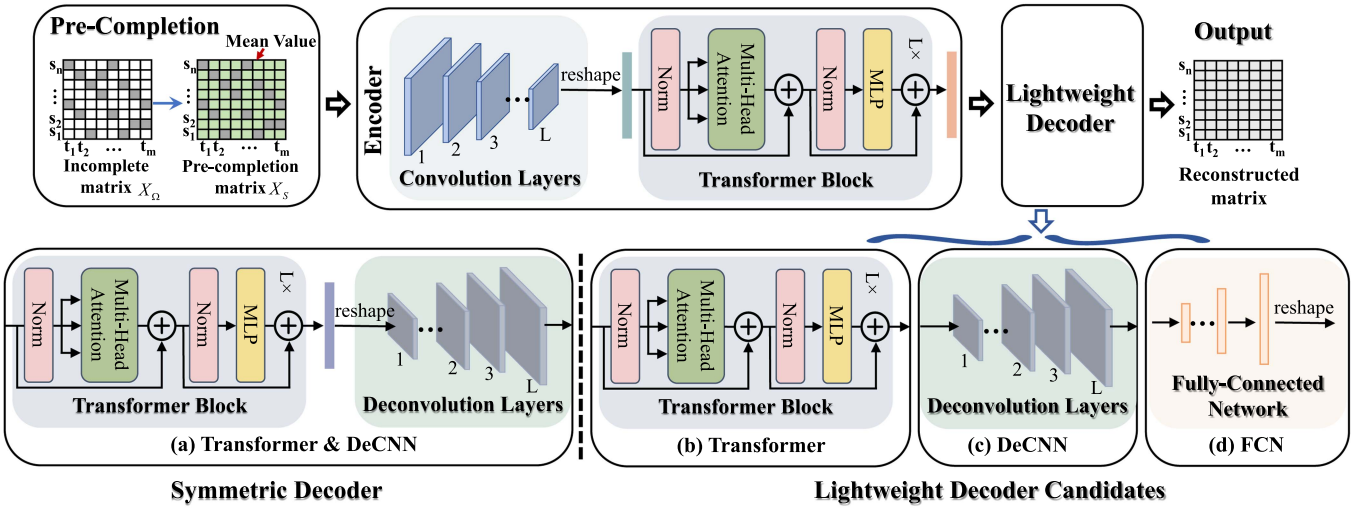


Fig. 6. Architecture of the CNN-Transformer Autoencoders.

matrix data have similar features) and translation invariance (i.e., $f(g(x)) = g(f(x))$, where $g(.)$ denotes convolution operation, and $f(.)$ denotes translation operation), their static parameters hinder them from adapting to massive data effectively. To bridge this gap, we propose a hybrid CNN-Transformer model for our encoder module. By combining CNN's inductive biases with Transformer's global inductive modeling capability and dynamic parameters, our hybrid model optimally adapts to the data. The specific workflow of CNN-Transformer encoder is shown in Fig. 6.

In more details, the pre-completion matrix is first fed into a CNN including multiple convolutional, batch normalization and activation layers to extract the features of the pre-completion matrix input. After the l-th convolutional layer, the l-th extracted feature map X_s^l can be expressed as

$$\begin{aligned} X_s^l &= \mathcal{F}\left(X_s^{l-1} * k^l + b_k^l\right), \quad l = 2, 3, \ldots, L, \\ X_s^1 &= \mathcal{F}\left(X_s * k^1 + b_k^1\right), \end{aligned} \tag{6}$$

where k^l denotes the convolutional filter of the l-th layer, $*$ represents the convolution operator, and b_k^l is a bias. Besides, $\mathcal{F}(\cdot)$ is a non-linear activation function ReLU, which can be mathematically represented by $\mathcal{F}(x) = \max(0, x)$, and L is the total number of convolutional layers.

As previously discussed, convolution layers primarily focus on local feature extraction, often overlooking comprehensive high-quality feature extraction due to the limited receptive field of CNN. To address this limitation and capture long-term dependencies without distance constraints in the input sequence, we incorporate a transformer block into our architecture. Traditionally, a transformer comprises encoder and decoder layers. In our case, we specifically utilize transformer encoder layers to extract global high-level features, in which, each encoder consists of a multi-head attention (MHA) block and a multi-layer perceptron (MLP) block, both preceded by a normalization layer. Fig. 7 shows the detailed architecture of the transformer block.

Therefore, the extracted convolutional feature map, i.e., X_s^l, is then fed into the transformer blocks after the size conversion

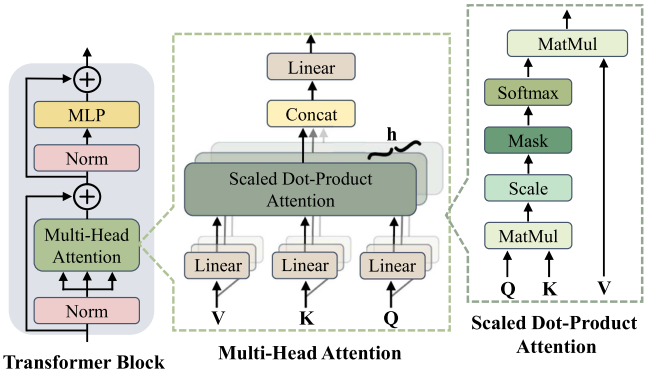


Fig. 7. Architecture of the Transformer Block.

(i.e., V_s). First, the desired size feature map V_s^0 is forwarded into a normalization layer, which denotes as

$$\hat{V}_s = \mathrm{Norm}\left(V_s\right) \tag{7}$$

Then, we compute query (Q), key (K), and value (V) projections according to the normalized $\hat{V}_s$, yielding

$$\begin{aligned} Q &= W^Q \hat{V}_s, \\ K &= W^K \hat{V}_s, \\ V &= W^V \hat{V}_s, \end{aligned} \tag{8}$$

where W^Q, W^K, and W^V are the projection weights for Q, K, and V, respectively.

Thereafter, the input of Q, K, and V will be forwarded into different heads of MHA block. In each head, a scale dot-product attention (SDPA) is used to extract the global feature, the process can be expressed as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V. \tag{9}$$

Specifically, in SDPA, the correlation between Q and K is first calculated through dot product operation, and then $\sqrt{d}$ is used to avoid overly large values of the inner product and extremely small gradients [31], where d is the dimension of K. Then, a masking operation is performed to mask the diagonal of self-attention score matrix to avoid high matching scores between identical vectors of Q and K. Later, we use softmax activation to generate an attention map, which will be used to form the final weighted output.

Subsequently, the output of each head will be concatenated together, followed by a linear layer. This process yields the final output of the MHA, which can be expressed by the following formula,

$$\text{head}_i = \text{Attention}\left(W_i^Q \hat{V}_s, W_i^K \hat{V}_s, W_i^V \hat{V}_s\right), i \in (0, h] \tag{10}$$

$$\text{MHA}(Q, K, V) = \text{Concat}\ (\text{head}_1, \ldots, \text{head}_h)\, W^O \tag{11}$$

where W_i^Q, W_i^K, and W_i^V represent the weights of the linear layers of the i-th head to map Q, K, and V, respectively. W^O is the weight of the last linear layer in MHA.

Afterwards, a skip-connection operation is utilized to avoid gradients vanishing and exploding. A normalization layer and a MLP layer are used to further transform the learning feature of MHA, which can be denoted as

$$V_s^1 = \text{MLP}\left(\text{Norm}\left(\text{MHA}(Q, K, V) + V_s\right)\right) \tag{12}$$

Therefore, after N transformer blocks, the global long-term dependencies will be captured, denoted as V_s^N.

Lightweight Decoder: The autoencoder's decoder, responsible for mapping the latent representation back to the input, varies its role in reconstructing different modal content. In the conventional autoencoders design, the symmetric decoder structure corresponding to our CNN-Transformer encoder is Transformer-DeCNN, which mirrors the encoder, comprising multiple transformer blocks followed by multiple deconvolution layers, as shown in the Fig. 6(a). While this design enhances reconstruction accuracy during training, it significantly impacts efficiency and memory consumption, especially for extensive IoT datasets and large models. To overcome these challenges, we diverge from the classical symmetric design found in traditional autoencoders. Instead, we introduce an asymmetric lightweight decoder. This design offers flexibility by allowing the selection of different implementation structures, balancing the trade-off between running time and accuracy. Our proposed lightweight decoder options include partial components of Transformer-DeCNN or simpler fully connected networks (FCN), as shown in Fig. 6(b)–(d).

The choice for a lightweight decoder design in our task stems from the inherently low semantic level of the model output. In contrast to the high semantic complexity of natural language, IoT data exhibits significant data redundancy and operates at a lower overall semantic level. Consequently, a straightforward decoder suffices for converting the learned latent features into the desired output. Employing a compact decoder in our asymmetric encoder-decoder setup leads to substantial reductions in computation. This efficiency in both time and memory usage renders our model particularly advantageous for training extensive models. In our subsequent experimental evaluations, we systematically explore the use of very compact decoders, outlining the precise trade-offs between reconstruction accuracy and computational efficiency.

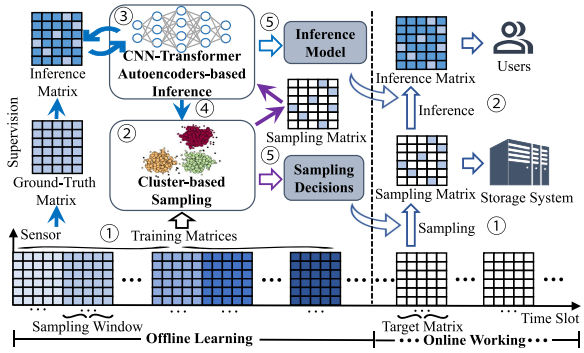


Fig. 8. The workflow of *CODE*$^+$.

Model Training: Unlike the GAN-based inference module adopted in *CODE* [1], which often experiences high output variance due to the unbounded noise of GAN without constraints, our CNN-Transformer Autoencoders structure prioritizes stability and reconstruction accuracy over originality and creativity. *CODE*$^+$ works in an end-to-end manner, reconstructing the input by predicting possible values for each unsampled location. The last layer of the lightweight decoder is a linear projection whose number of output channels equals the size of input data matrix. Our loss function calculates the mean squared error (MSE) between the reconstructed data matrix and the original data matrix, focusing solely on unsampled locations. To enhance model robustness and prevent overfitting, we employ the Adaptive Moment Estimation (Adam) algorithm for optimization. Additionally, widely used techniques such as Dropout, L2 regularization, and a learning rate schedule are applied to mitigate the risk of overfitting [32].

C. *Workflow of* CODE$^+$

In this subsection, we delineate the workflow of *CODE*$^+$ when deployed in operational distributed IoT data systems, encompassing the distinct stages of offline learning and online working, as depicted in Fig. 8.

The offline learning stage, a pivotal phase in *CODE*$^+$'s deployment, involves the following steps: 1) Based on the spatio-temporal analysis of historical data, we set a sampling window, typically on a weekly basis, to synchronize with temporal patterns. Using this sampling window size, we collect full-size historical data matrices, which serve as the foundation for training the two integral models within *CODE*$^+$. 2) Using the historical data matrices obtained in Step 1), we implement our proposed clustering-based matrix sampling module to generate the initial sampling strategies. 3) The historical sampled data matrices derived from the sampling strategies in Step 2) are fed into the CNN-Transformer Autoencoder-based matrix inference module to reconstruct the unsampled data. We iteratively optimize the

inference model by minimizing the error between the inference matrices and the ground-truth matrices. 4) Once the inference model from Step 3) is trained, we evaluate the reconstruction error of the complete data under the current sampling strategies. This error is compared against the system's acceptable error threshold. If the error is below the threshold, we finalize the sampling strategies and the inference model. If the error exceeds the threshold, we return to Step 2), increase the sampling ratio, and regenerate the sampling strategies. We iterate through Steps 2) to 4) until the reconstructed data from the trained inference model meets the system's requirements. 5) Once the model meets the system's requirements, we finalize and output the accurate sampling strategies and the well-trained inference model. These will seamlessly guide data collection during the online working stage. Notably, the inherent randomness in our clustering-based matrix sampling module allows for the generation of multiple high-quality and distinct sampling strategies during the offline learning stage.

The online working stage encompasses the future data collection and processing activities and includes the following aspects: 1) Future data collection is directed by the finalized sampling strategy. During this phase, only partial data is collected according to the sampling positions specified by the strategy, resulting in a data volume significantly smaller than the full-size data, which can be directly stored in the storage system. Additionally, varying sampling strategies can be employed across different time windows. This approach helps to balance the energy consumption of various sensors, thereby extending the operational lifespan of the entire data collection system. 2) When downstream applications that rely on full-size data require it, the system can use the inference model to reconstruct the missing data, providing the complete data set to the applications.

V. Performance Evaluation

In this section, we conduct comprehensive data-driven experiments to evaluate the overall performance of *CODE*$^+$, verify the effectiveness of each individual component, analyze the impact of sampling ratio, and assess *CODE*$^+$'s robustness when deployed in different distributed IoT data systems or handling different distribution data.

A. Evaluation Methodology

Platform: We develop *CODE*$^+$ in Python, and implement the core functions based on PyTorch, which is an open-source machine learning framework. The experiments are carried out on a server with 4 CPUs, each containing 192 Intel(R) Xeon(R) Platinum 8260 processor running at 2.40GHz with 24 cores, along with the utilization of a graphics processing unit card (NVIDIA Tian X) to accelerate the training process.

Setup: In our experiments, the *CODE*$^+$ model has a size of approximately 5.3MB. The encoder component of the matrix inference module comprises three convolutional layers, each with 16 convolution kernels, followed by six transformer blocks, each containing six attention heads. Symmetrically, the decoder mirrors this structure, utilizing transformer blocks and deconvolutional layers of equivalent size. Collectively, this architecture results in a total of approximately 1.37 million parameters.

Datasets: To comprehensively evaluate the performance of our proposed *CODE*$^+$, we employ a diverse range of datasets, including three real-world IoT datasets and one synthetic dataset generated with a Gaussian distribution. Each dataset varies in scale and distribution, providing a robust basis for assessment. The details of these datasets are described as follows.

- *Provincial highway traffic data (A):* This dataset records daily transition traffic within the provincial highway ETC system, encompassing data from 465 stations and approximately 1.6 million daily transitions. For our evaluation, we utilize a four-month period, which translates to 16 weeks of data. The hourly aggregated traffic data results in a traffic matrix of dimensions 465×2688.
- *WiFi system data (B):*[3] This dataset captures public WiFi usage by tracking the number of users connected to access points (APs) on a campus per hourly time slot. The dataset includes data from 3,600 APs, with each time slot representing one hour of usage. For our evaluation, we consider a six-week period, producing a data matrix with dimensions 3600×1008.
- *National air quality data (C):*[4] This dataset contains comprehensive air quality monitoring data collected from 1,518 sensors located in 375 cities nationwide, with hourly data recordings. For evaluation purposes, we utilize a full year's data (52 weeks) to ensure a robust and thorough performance assessment. This results in a data matrix with dimensions 1518×8736.
- *Synthetic Gaussian-distribution data (D):* This dataset is generated to specifically evaluate the performance of *CODE*$^+$ across various data distribution scenarios. For a robust and meaningful comparison, we synthesized data following a Gaussian distribution that mirrors the scale and characteristics of Dataset A. The synthetic data is created with a mean of 96.5 and a variance of 132, ensuring that it closely resembles real-world conditions. Consequently, the total data volume is set at 465×2688.

Our primary evaluation focuses on dataset A, as detailed in Sections V-B, V-C, and V-D. For performance validation with data diversity and distribution heterogeneity, we conduct robust experiments with datasets B, C, and D, as discussed in Section V-E. Given the significant spatio-temporal correlations and the pronounced periodic characteristics observed in the temporal dimension of these datasets, we set the sampling window size to one week. Each hour within this period is treated as an individual time slot, resulting in a total of 168 time slots. This choice of sampling window ensures the capture of a complete temporal cycle, thereby enhancing the model's ability to extract and learn the underlying temporal features effectively. To mitigate the risk of overfitting, we initially divide each dataset into training, validation, and testing sets following a 6:2:2 ratio. Subsequently, we employ a data augmentation technique to artificially increase the size of training set. This is achieved by repeatedly applying the sampling strategy detailed in Section IV-A. Each iteration yields multiple sets of sample data that, while maintaining the same overall data benefit, feature

[3]https://github.com/Intelligent-WiFi/DataSet
[4]https://quotsoft.net/air

TABLE I
THE DESCRIPTION OF USING DATASETS

Name	Time span	Attribute 1	Attribute 2	Original size	Augmented size	Training set	Validation set	Testing set	Source
Dataset A	16 weeks	ETC ID	Time slot	465 × 2688	465 × 26880	465 × 16800	465 × 5040	465 × 5040	privacy
Dataset B	6 weeks	AP ID	Time slot	3600 × 1008	3600 × 10080	3600 × 6720	3600 × 1680	3600 × 1680	public
Dataset C	52 weeks	sensor ID	Time slot	1518 × 8736	1518 × 87360	1518 × 53760	1518 × 16800	1518 × 16800	public
Dataset D	16 weeks	device ID	Time slot	465 × 2688	465 × 26880	465 × 16800	465 × 5040	465 × 5040	synthetic

different sampling locations as dictated by the inherent design of the sampling strategy. This approach effectively enlarges the training set by a factor of 10. To ensure the consistency of the sampling strategy across validation and testing sets, we apply the same augmentation process to these sets. Consequently, the sizes of the training, validation, and testing sets are all increased tenfold. It is crucial to note that the division into training, validation, and testing sets is performed on the original, unaugmented data first. This is followed by the augmentation of each set independently, thereby preserving data integrity and ensuring an accurate estimation of the generalization error. The specifics of each dataset are summarized in Table I.

Metrics: Denote by X_{ij} and $\hat{X}_{ij}$ the raw data and inferred data, at (i,j)-th element of the traffic matrix X and estimated matrix $\hat{X}$, respectively, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Ω is the sampled entries, $\bar{\Omega}$ is the unsampled entries, and $\Omega + \bar{\Omega}$ represents the total data entries. N is the number of unsampled data entries, i.e., $N = |\bar{\Omega}|$. To comprehensively evaluate the performance of our proposed *CODE*$^+$ framework, we employ a diverse set of metrics categorized into three key areas: inference accuracy, data distribution fidelity, and operational efficiency.

- *Inference Accuracy:* This category evaluates the precision of *CODE*$^+$ is reconstructing the original data from the sampled subset. The critical metrics in this category include:
  - *Relative Error (RE):* This metric measures the relative discrepancy between each actual value and its inferred counterpart, providing an intuitive indication of inference accuracy. It is computed by $\frac{|X_{ij}-\hat{X}_{ij}|}{X_{ij}}$ $((i,j) \in \bar{\Omega})$.
  - *Mean Absolute Percentage Error (MAPE):* This metric assesses the average accuracy by comparing the absolute percentage differences between the original and reconstructed values. It provides an overall measure of inference quality and is calculated by $\frac{1}{N}\sum_{i,j\in\bar{\Omega}} |\frac{X_{ij}-\hat{X}_{ij}}{X_{ij}}|$.
- *Data Distribution Fidelity:* Metrics in this category assess how well the reconstructed data preserves the statistical characteristics and distribution of the original dataset. The primary metric is:
  - *Kullback-Leibler Divergence (KLD):* This metric quantifies the divergence between the probability distributions of the original data and the reconstructed data. It reflects how accurately the reconstruction approximates the original distribution and is computed by $\sum_{(i,j)\in\bar{\Omega}} X_{ij}\log(X_{ij}/\hat{X}_{ij})$.
- *Operational Efficiency:* This category evaluates the computational performance of *CODE*$^+$, specifically focusing on the time required for data processing. The main metric is:
  - *Time:* This metric measures the total time taken to process the data, reflecting the efficiency of the data reconstruction process.

In summary, these metrics provide a comprehensive evaluation framework to assess the model performance from multiple dimensions. Notably, lower values across these metrics indicate better performance, affirming the model's effectiveness in accurate data reconstruction, preservation of data distribution, and efficient processing.

Baselines: To evaluate the performance of *CODE*$^+$, the following four baselines are adopted.

- *CODE [1]:* is the state-of-the-art data collection algorithm based on cluster-based matrix sampling and GAN-based matrix inference.
- *NTC [30]:* is an advanced data collection algorithm including random sampling approach and deep learning-based data reconstruction model.
- *MF [33]:* is a conventional data inference algorithm based on low-rank matrix factorization, in which the random sampling approach is adopted.
- *MC-Weather [34]:* is a data collection scheme based on matrix completion theory, in which the UTSCS (i.e., Uniform Time-Slot and Cross Sample) model is designed for matrix sampling, i.e., sampling uniformly in time and crossly in location, and low-rank matrix factorization is adopted for data inference.

B. Performance Comparison

We first evaluate the overall performance on dataset A in terms of model generalization, inference accuracy, data distribution fidelity, operational efficiency, and spatio-temporal performance, where the sampling ratio is fixed to 20%.

Generalization: We first evaluate *CODE*$^+$'s generalization capability to unseen data by monitoring both the training loss and the validation loss during the training process. As shown in Fig. 9, *CODE*$^+$ demonstrates strong generalization ability. As training progresses, both losses decrease, indicating effective learning. The gap between training and validation losses remains relatively small throughout the training period, suggesting good initial generalization. Additionally, the validation loss remains stable without significant fluctuation, indicating an absence of overfitting.

Distribution: Fig. 10 shows the performances of all schemes on reconstructed data distribution fidelity. Obviously, *CODE*$^+$ outperforms all baselines, exhibiting significantly smaller values in the data distribution metric (KLD). Specifically, *CODE*$^+$ achieves an outstanding score of $2.84e^{-4}$, while the scores of *CODE*, *NTC*, *MC-Weather*, and *MF* reach about 7.97×10^{-4},

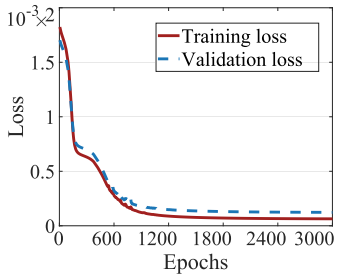


Fig. 9. Generalization of *CODE*$^+$.

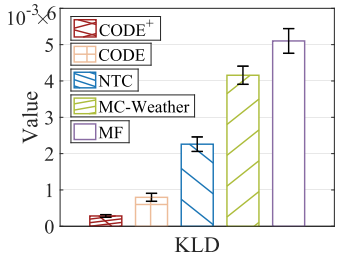


Fig. 10. Comparison of distribution performance.

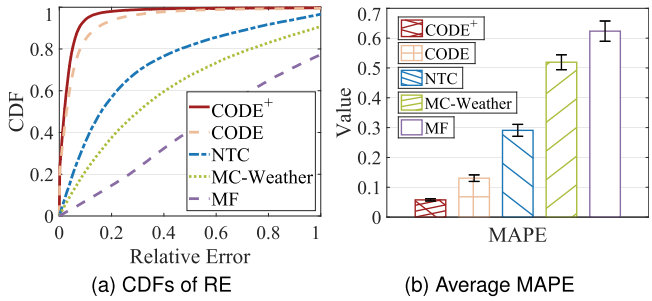


Fig. 11. Comparison of accuracy performance.

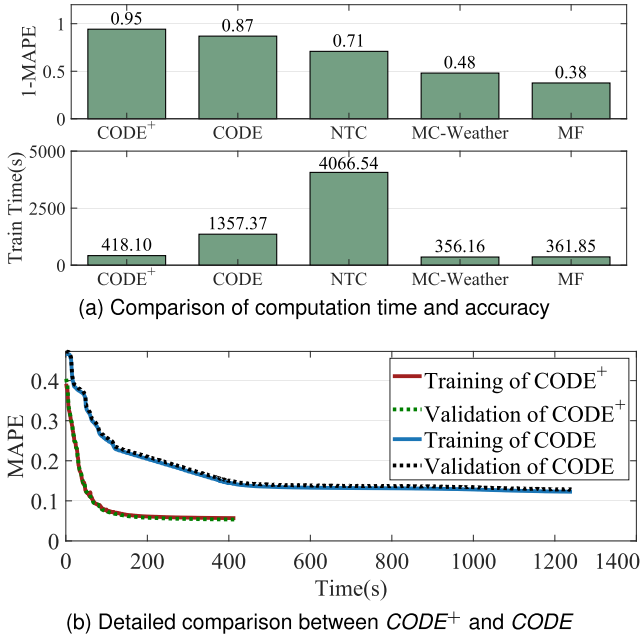


Fig. 12. Comparison of efficiency performance.

2.26×10^{-3}, 4.16×10^{-3}, and 5.10×10^{-3}, respectively. This demonstrates *CODE*$^+$'s superiority, outperforming these baselines by factors of 2.8, 8, 14.6, and 18, respectively. These results underscore the effectiveness of *CODE*$^+$ in reconstructing unsampled data and capturing the distribution of long-tailed skewed data.

Accuracy: Fig. 11 shows the performances of all approaches on accuracy metrics RE and MAPE. It is evident that *CODE*$^+$ consistently outperforms other baselines for both metrics. Specifically, Fig. 11(a) shows the CDFs of RE for all the approaches (i.e., one location in the matrix is a sample), and we can observe that *CODE*$^+$ can outperform other baselines significantly. For instance, at the percentile of 80%, the REs of *CODE*, *NTC*, *MC-Weather*, and *MF* reach about 0.08, 0.46, 0.73, and 1, respectively, while *CODE*$^+$ achieves a score of 0.05, improving the performance by 37.5%, 89.1%, 93.2%, and 95%, respectively. Fig. 11(b) shows the overall average MAPE with error bars (i.e., one-week matrix results are calculated as one sample), and we can make the following two statements. First, *CODE*$^+$ consistently achieves superior performance and the gaps are significant. Specifically, *CODE*$^+$ scores about 0.058 compared to 0.131, 0.291, 0.519, and 0.623 for *CODE*, *NTC*, *MC-Weather*, and *MF*, respectively, improving the performance by 55.8%, 80.1%, 88.2%, and 90.7%, respectively. Second, when observing the error bars, we can find that the deviation achieved by *CODE*$^+$ is much smaller when compared to other baselines, demonstrating its reliability in maintaining performance consistency.

Efficiency: Fig. 12 presents a comparative analysis of the operational efficiency across five algorithms, highlighting two critical observations. First, examining the computation time and reconstruction accuracy for each algorithm reveals that *CODE*$^+$ excels in both metrics, as depicted in Fig. 12(a). Specifically, *CODE*$^+$ completes its training processes in just 418.10 seconds with a remarkable 95% accuracy. This indicates an efficiency performance of 3.25×, 9.73×, 0.85×, and 0.87× compared to *CODE*, *NTC*, *MC-Weather*, and *MF* respectively, and improves accuracy by 9.2%, 33.8%, 97.9%, and 150% correspondingly. Second, a detailed comparison between *CODE*$^+$ and *CODE* reveals that *CODE*$^+$ excels in terms of convergence speed, inference accuracy, and operational efficiency, as shown in Fig. 12(b). Both *CODE*$^+$ and *CODE* exhibit strong generalization capabilities, indicating their robustness across various scenarios. This comparative analysis underscores the stability, efficiency, and effectiveness of *CODE*$^+$ in addressing the challenges of data sampling and reconstruction.

Spatio-Temporal Performance: With the overall performance guarantee, we then check its spatio-temporal performance, i.e., how it performs across different stations over time. Fig. 13 shows the CDF results at stations (i.e., treating data values in each station as one sample) for MAPE and KLD metrics. It can be seen that *CODE*$^+$ can outperform the other four baselines significantly for both metrics. For instance, for 80%

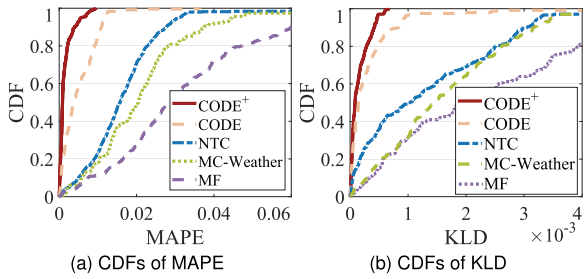


Fig. 13. Spatial performance.

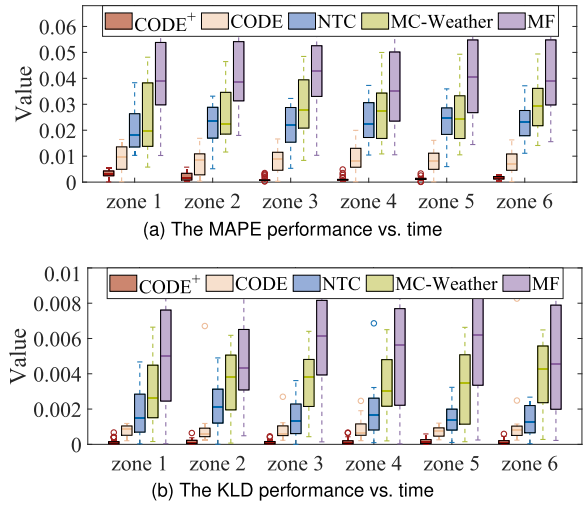


Fig. 14. Temporal performance.

of stations, their MAPE and KLD can be smaller than 0.002 and 2.84×10^{-4} when adopting our proposed scheme of *CODE*$^+$. These values are enlarged to 0.009 and 4.65×10^{-4} with *CODE*, 0.023 and 2.51×10^{-3} with *NTC*, 0.028 and 2.63×10^{-3} with *MC-Weather*, and 0.047 and 3.93×10^{-3} with *MF*, respectively. Fig. 14 shows the temporal box-plots of MAPE and KLD, respectively, where one day is uniformly divided into six temporal zones. We can have the following three major observations. First, under all temporal zones, *CODE*$^+$ can achieve the supreme performance with significant gaps. Second, when observing the 25^{th}, 50^{th}, 75^{th}, and 100^{th} percentiles, their gaps are quite small by adopting *CODE*$^+$ and *CODE*. However, these percentiles can have large deviations when adopting other baselines, demonstrating the stability of *CODE*$^+$ and *CODE*. Finally, with the time evolving, the performance of *CODE*$^+$ and *CODE* remains stable while that of other schemes can fluctuate dramatically, further indicating their superiority.

C. Impact of Sampling Ratio

Then we examine the impact of sampling ratio by varying the sampling ratio from 0.05 to 0.5 with the step of 0.05, and plot the average metric performance by adopting the different schemes in Fig. 15. It is obvious that *CODE*$^+$ can outperform all the baselines under all sampling ratio conditions. In addition, the inference error usually decreases with sampling ratio for

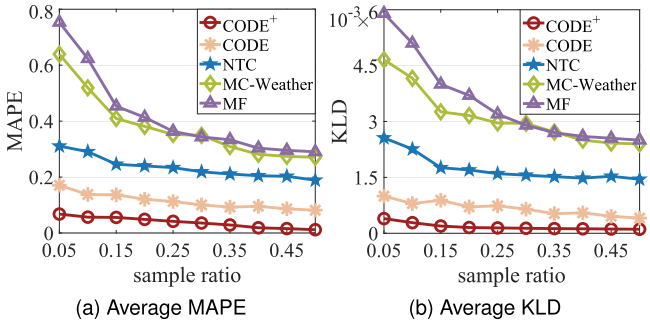


Fig. 15. The average performance versus sampling ratio.

TABLE II
ABLATION EXPERIMENT RESULTS OF DATASET A

Metric / Variant	Accuracy MAPE	Distribution KLD	Efficiency Time (s)
- *Random sampling*	0.1573	1.01×10^{-3}	454.42
- *Uniform sampling*	0.1495	8.48×10^{-4}	494.28
- *Transformer*	0.1064	6.07×10^{-4}	401.83
- *DeCNN*	**0.0501**	$\mathbf{2.42 \times 10^{-4}}$	**335.01**
- *FCN*	0.1216	7.09×10^{-4}	387.16
CODE$^+$	0.0577	2.84×10^{-4}	418.12

all schemes. The difference is that when adopting *CODE*$^+$, the error can be bounded within a small range even with a quite small sampling ratio, while for other baselines, larger sampling ratio is required to reach a satisfied performance. Taking the metric of MAPE as an example, when adopting *CODE*$^+$, the score can be smaller than 0.07 with a sampling ratio as low as 0.05. Conversely, to reach the same target inference accuracy, *CODE* requires a sampling ratio of about 0.5, which is even higher for *NTC*, *MC-Weather*, and *MF*. This means that to reach the same inference accuracy, *CODE*$^+$ can save the data collection cost by more than 90%.

D. Effectiveness

Effectiveness of Sampling Scheme: In *CODE*$^+$, a cluster-based matrix sampling scheme tailored for long-tailed distribution data collection is meticulously designed. The variations in information entropy resulting from different sampling approaches are depicted in Fig. 4, as demonstrated in our earlier analysis. To underscore the impact of these diverse sampling approaches on overall data collection, we conduct an ablation experiment. Specifically, we replace the cluster-based sampling scheme in *CODE*$^+$ with random sampling and uniform sampling schemes. The objective is to assess the final reconstruction performance of the unsampled data, and the results are summarized in Tables II and III. The outcomes are compelling, in which the performances utilizing random sampling and uniform sampling schemes degrade significantly. This degradation underscores that conventional sampling schemes are inadequate for data collection characterized by long-tailed data distribution. In contrast,

TABLE III
ABLATION EXPERIMENT RESULTS OF DATASET C

Variant \ Metric	Accuracy MAPE	Distribution KLD	Efficiency Time (s)
- *Random sampling*	0.0729	1.58×10^{-3}	197.53
- *Uniform sampling*	0.0725	1.57×10^{-3}	194.71
- *Transformer*	0.0668	1.47×10^{-3}	317.31
- *DeCNN*	0.0490	1.02×10^{-3}	303.7
- *FCN*	0.0531	1.24×10^{-3}	390.47
CODE$^+$	**0.0459**	$\mathbf{3.84 \times 10^{-4}}$	**257.7**

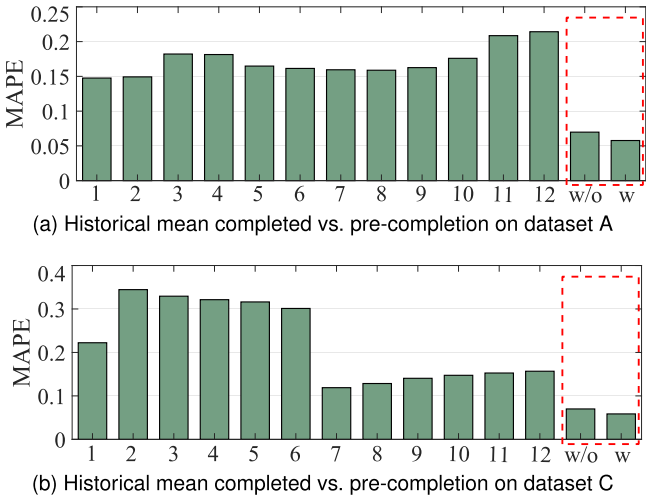


Fig. 16. Effectiveness of Pre-Completion.

our proposed method effectively solves this challenge, highlighting the pivotal role of our tailored cluster-based sampling approach in ensuring accurate and efficient data reconstruction.

Effectiveness of Pre-Completion: To assess the effectiveness of the pre-completion component in *CODE*$^+$, we compare its performance against inference matrices completed using the historical mean over various time ranges. Specifically, we consider 12 different time ranges, extending from the past 1 week to the past 12 weeks, and use these historical means as generic baselines. The generated inference matrices from these baselines are then compared to the performance of *CODE*$^+$ with and without the pre-completion component. The results, shown in Fig. 16, yield three key observations. First, both variants of *CODE*$^+$ significantly outperform all baseline matrices completed using historical means, demonstrating a substantial advantage in terms of accuracy. Second, a direct comparison between *CODE*$^+$ with and without the pre-completion component shows that the inclusion of this component markedly enhances inference accuracy. This indicates the pre-completion component's crucial role in improving performance. Third, the effectiveness of using historical means for completion varies unpredictably across different time ranges and datasets. For instance, there is no consistent pattern to determine which historical mean time range yields the best completion accuracy. This irregularity persists when comparing the performance between datasets A and C.

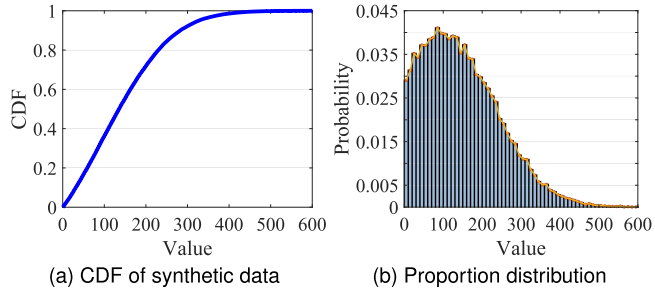
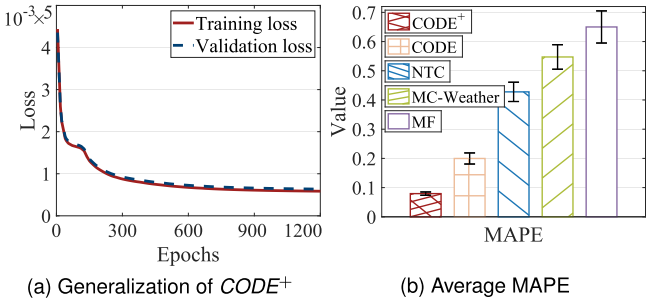


Fig. 17. Gaussian distribution of synthetic values.

Effectiveness of Lightweight Decoder: We proceed to validate the effectiveness of our lightweight decoder, considering the effectiveness and efficiency trade-offs involving three distinct candidates, i.e., *Transformer*, *DeCNN*, and *FCN*. Note that we implement a symmetric decoder structure, *Transformer* & *DeCNN*, corresponding to our CNN-Transformer encoder, within *CODE*$^+$. For practical system deployment, the optimal lightweight technology to implement the decoder should be selected based on the specific characteristics of the dataset in use, as validated by the results shown in Tables II and III. For instance, the *DeCNN* architecture yields the best performance for Dataset A, whereas the *Transformer* & *DeCNN* combination excels for Dataset C. This paper argues that for data completion tasks involving significant spatio-temporal correlation and low-rank characteristics, a lightweight decoder structure is both sufficient and effective. More complex network structures might not only be unnecessary but could also introduce additional time and resource overhead. Hence, depending on the dataset's attributes, one can achieve a balance between training overhead and accuracy by choosing the most appropriate lightweight decoder technology.

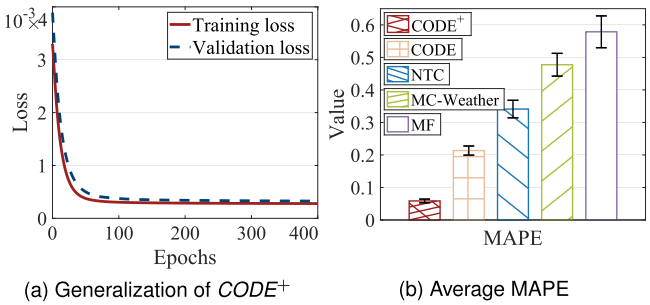
E. *Robustness of* CODE$^+$

To evaluate the robustness of *CODE*$^+$, we conduct experiments on datasets B, C, and D to determine its performance across various distributed IoT data systems and diverse data distributions. Dataset D features a synthetic Gaussian distribution designed to mirror the scale and characteristics of Dataset A, as depicted in Fig. 17. When comparing Figs. 17(b) with 2(b), it is evident that Dataset A exhibits increased complexity due to its deviations from a Gaussian distribution. The results of robustness experiments, as shown in Figs. 18 to 20, reveal three key insights. First, *CODE*$^+$ demonstrates strong generalization capability across all three datasets. This is evidenced by the stable and consistent loss curves shown in Figs. 18(a), 19(a), and 20(a). Second, *CODE*$^+$ consistently outperforms other comparison algorithms across all datasets in terms of inference accuracy and fidelity to the original data distribution. This highlights the algorithm's capability to maintain high performance across diverse data domains. Third, a comparative analysis of *CODE*$^+$ across datasets A, B, C, and D shows that it performs optimally on the Gaussian-distributed dataset D and least effectively on the WiFi system dataset B. For datasets A (highway ETC) and C



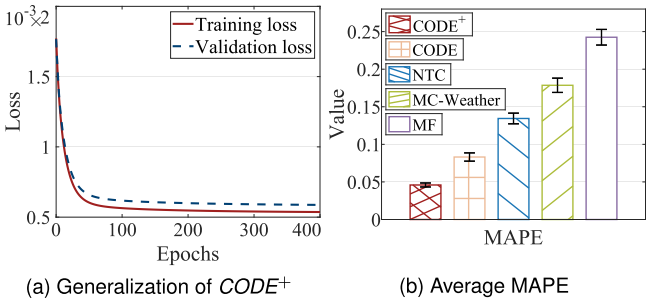
(a) Generalization of *CODE*$^+$ (b) Average MAPE

Fig. 18. Robustness experiments on dataset B.



(a) Generalization of *CODE*$^+$ (b) Average MAPE

Fig. 19. Robustness experiments on dataset C.



(a) Generalization of *CODE*$^+$ (b) Average MAPE

Fig. 20. Robustness experiments on dataset D.

(air quality monitoring), *CODE*$^+$ displays similar performance levels. Specifically, for the MAPE metric, *CODE*$^+$ achieves approximately 0.058, 0.079, 0.059, and 0.046 for datasets A, B, C, and D, respectively. These variations in performance can be attributed to the intrinsic characteristics of each dataset. *CODE*$^+$ performs better with more uniform data distributions and faces challenges with increased data volatility. Notably, *CODE*$^+$ demonstrates superior performance across different target domains, underscoring its robustness and adaptability.

VI. Related Work

A. Data Sampling

The exponential increase in data volume has imposed significant challenges on sensing and storage within distributed IoT data systems, prompting advancements in data sampling methodologies. Existing literature categorizes prevalent sampling schemes into three main types: traditional sampling, active sampling, and online sampling.

Traditional Sampling: Random sampling and uniform sampling are two of the most common sampling approaches used in various fields of research and data analysis. Random sampling was rigorously analyzed by Candès and Recht [26], demonstrating that an $n_1 \times n_2$ matrix with rank r can be accurately recovered by at least $Cn^{6/5}r\log n$ random samples, where $n = max(n_1, n_2)$. Zhang and Aeron [35] later extended this, establishing that a sufficient number of random tubal samples $O(\eta(\mathcal{T})knr\log^2 n)$ enables precise recovery of unobserved entries in 3D tensor completion problem. Subsequent study [36] further reduced the sample complexity to $O(n^{(a+1)})$ under the Bernoulli sampling model, where a is a small constant. Thereafter, numerous works [30], [37] have explored data recovery using random sampling methods. Uniform sampling, involves selecting an equal number of samples from both temporal and spatial domains. For instance, Xie et al. [34] introduced an online data collection scheme using uniform sampling, adapting sampling locations based on the *UTSCS* model.

Active Sampling: This less explored area focuses on selecting optimal sampling locations. For instance, *CCS-TA* [38] devised a leave-one-out and re-sampling principle, employing multiple-step sampling to minimize the total number of samples. Wang et al. [39] introduced the *SPACE-TA* framework, which actively identifies sampling locations with significant uncertainties or errors. Deng et al. [40] proposed an adaptive scheme utilizing leverage scores to reduce measurement costs in large-scale networks. Additionally, He et al. [41] presented a Query-by-Committee-based sampling approach tailored for high-dimensional magnetic resonance imaging (MRI) with limited imaging speed.

Online Sampling: From the relevant literature, it is evident that few studies have focused on online sampling. This is largely due to the inherent challenges of real-time decision-making, sampling bias, and the need for adaptability to concept drift, which are intrinsic to the real-time nature of data collection and analysis. Al Jawarneh et al. have made notable contributions in this area, particularly in contexts where cluster computing resources are limited. In their work outlined in [27], they explored spatial-aware approximate processing techniques for Big Data streams. This research aimed to optimize real-time analysis in geospatial applications by leveraging spatial awareness to improve the efficiency of processing large volumes of data. In [28], they introduced an approach specifically designed for online Big Data sampling within the context of smart cities. This approach emphasized maintaining spatial representativeness in geospatial data streams and dynamically adjusting sampling intensity based on spatial density and distribution patterns. Further expanding on these concepts, their study in [29] proposed a comprehensive framework tailored for geospatial data applications. This framework employed spatially representative sampling and approximate join methods to effectively manage and process high-volume, high-velocity spatio-temporal data.

Traditional sampling and active sampling methods predominantly concentrate on the quantity of samples rather than their

quality. While effective for Gaussian data, these methods are often ineffective with non-Gaussian datasets. In contrast, current online sampling methods prioritize the user's expectations in terms of QoS metrics, such as latency, throughput, and accuracy. These methods make real-time decisions on whether to discard incoming data, aiming to manage the system's immediate performance. However, when handling large volumes of data, this can lead to significant costs and inefficiencies, as data is often discarded reactively rather than preemptively. Moreover, the rapid growth in data necessitates not only improvements in QoS but also substantial reductions in data collection costs. This includes lowering the energy consumption of sensors, transmission costs, and storage expenses. Therefore, how the chosen sampling strategy impacts subsequent data reconstruction processes should be carefully investigated, which is often overlooked in existing sampling schemes.

B. Data Inference

While several studies have explored data inference methods using sparse sensing technologies, such as compressive sensing [37], [42], matrix completion [43], [44], and tensor completion [45], [46], these methods predominantly rely on mathematical modeling grounded in linear features. Deep learning models, with their ability to extract non-linear features, have been applied in this context. For instance, He et al. [47] introduced *NCF*, a framework that combines matrix factorization with neural networks for generalization. Xie et al. [30] proposed a neural tensor completion framework employing 3D CNN to extract hidden features, thereby enhancing missing data inference accuracy. Deng et al. [48] presented a graph neural network approach for network traffic imputation, capturing topological correlations in network traffic. Additionally, a class of data reconstruction techniques [23], [49], [50] based on GAN networks has gained popularity. For example, Tan et al. [50] introduced a projected generative adversarial network to recover complete point clouds from partial and sparse data. Xie et al. [23] presented a deep adversarial tensor completion scheme to infer the skewed distribution of missing data.

In summary, all the mentioned methods have their applicability, but they also come with some limitations. Traditional data completion approaches often overlook non-linear features, leading to suboptimal data reference accuracy. Conversely, deep learning-based methods enhance data inference accuracy but still under the assumption that the sparse data is known ahead. Consequently, these methods focus primarily on improving the inference accuracy of missing data, neglecting the intricacies of data sampling. However, the sample qualities can affect the data inference performance significantly, and there is limited study available to tackle the deep-coupled problem, which motivates our research in this paper. In our previous work [1], we have investigated the deep-coupled problem of matrix sampling and matrix inference, and proposed an efficient compact data collection framework *CODE*. In this work, we further improve it by proposing more comprehensive and effective framework design, presenting a faster and more stable matrix inference module to optimize possible inefficiency and instability arising from the uncontrolled and creative nature of GAN used in *CODE*.

VII. Conclusion and Future Work

In this paper, we propose *CODE*$^+$, a novel framework designed to address the problem of heavy data collection cost in large-scale distributed IoT systems, emphasizing compact, fast, and accurate data sampling and inference. To achieve this, *CODE*$^+$ integrates two key technical components: cluster-based matrix sampling to identify optimal sampling locations, which significantly reduces data collection costs without compromising data benefits, and CNN-Transformer Autoencoders-based matrix inference for fast and accurate data reconstruction, enabling thorough exploration of data for relevant applications. Extensive experiments on four datasets demonstrate that *CODE*$^+$ outperforms four state-of-the-art baselines in terms of data reconstruction accuracy, data distribution fidelity, and computational efficiency. These results confirm that *CODE*$^+$ consistently delivers superior performance across various target domains, highlighting its versatility and robustness.

This work provides valuable insights and directions for future research on improving the temporal and spatial robustness of data collection. Our next step is to explore the development of a fully online version of *CODE*$^+$ to facilitate real-time data collection and processing, enabling prompt responses to new data patterns. Additionally, to address the new challenge of inference error accumulation over time, we aim to investigate a novel method to ensure the system's reliability and robustness over extended periods.

References

[1] H. Lu et al., "CODE: Compact IoT data collection with precise matrix sampling and efficient inference," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2022, pp. 743–753.
[2] Y. Ding, Y. Yang, W. Jiang, Y. Liu, T. He, and D. Zhang, "Nationwide deployment and operation of a virtual arrival detection system in the wild," in *Proc. ACM SIGCOMM Conf.*, 2021, pp. 705–717.
[3] X. Zhang and T. Wang, "Elastic and reliable bandwidth reservation based on distributed traffic monitoring and control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 12, pp. 4563–4580, Dec. 2022.
[4] H. Lu et al., "FL- AMM: Federated learning augmented map matching with heterogeneous cellular moving trajectories," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3878–3892, Dec. 2023.
[5] J. Cui et al., "Collaborative intrusion detection system for SDVN: A fairness federated deep learning approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 9, pp. 2512–2528, Sep. 2023.
[6] C. Shi, J. Liu, H. Liu, and Y. Chen, "WiFi-Enabled user authentication through deep learning in daily activities," *ACM Trans. Internet Things*, vol. 2, no. 2, pp. 1–25, 2021.
[7] Y. Deng et al., "AUCTION: Automated and quality-aware client selection framework for efficient federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1996–2009, Aug. 2022.
[8] S. Yue et al., "Federated offline reinforcement learning with proximal policy evaluation," *Chin. J. Electron.*, vol. 33, no. 6, pp. 1–13, 2024.
[9] K. Li et al., "Fair scheduling for data collection in mobile sensor networks with energy harvesting," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1274–1287, Jun. 2019.
[10] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (Extended version)," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 662–676, Jun. 2012.
[11] P. Tong, M. Li, M. Li, J. Huang, and X. Hua, "Large-scale vehicle trajectory reconstruction with camera sensing network," in *Proc. ACM 27th Annu. Int. Conf. Mobile Comput. Netw.*, 2021, pp. 188–200.

[12] A. M. Avila and I. Mezić, "Data-driven analysis and forecasting of highway traffic dynamics," *Nature Commun*, vol. 11, no. 1, pp. 2090–2105, 2020.
[13] J. Zhang, C. Zhao, and W. Gao, "Optimization-inspired compact deep compressive sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 4, pp. 765–774, May 2020.
[14] M. Lyu et al., "Sensing matrix optimization for random stepped-frequency signal based on two-dimensional ambiguity function," *Chin. J. Electron.*, vol. 33, no. 1, pp. 161–174, 2024.
[15] N. Razin and N. Cohen, "Implicit regularization in deep learning may not be explainable by norms," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21174–21187.
[16] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, and T. Huang, "Low-rank high-order tensor completion with applications in visual data," *IEEE Trans. Image Process.*, vol. 31, pp. 2433–2448, 2022.
[17] S. Duan et al., "MOTO: Mobility-aware online task offloading with adaptive load balancing in small-cell MEC," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 645–659, Jan. 2024.
[18] Z. Wang, K. Liu, J. Hu, J. Ren, H. Guo, and W. Yuan, "AttrLeaks on the edge: Exploiting information leakage from privacy-preserving co-inference," *Chin. J. Electron.*, vol. 32, pp. 1–12, 2023.
[19] H. Cao et al., "HandKey: Knocking-triggered robust vibration signature for keyless unlocking," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 520–534, Jan. 2024.
[20] F. Lyu et al., "LEAD: Large-scale edge cache deployment based on spatio-temporal WiFi traffic statistics," *IEEE Trans. Mobile Comput.*, vol. 20, no. 8, pp. 2607–2623, Aug. 2021.
[21] F. Montagna et al., "A low-power transprecision floating-point cluster for efficient near-sensor data analytics," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 5, pp. 1038–1053, May 2022.
[22] S. Suryavansh, A. Benna, C. Guest, and S. Chaterji, "A data-driven approach to increasing the lifetime of IoT sensor nodes," *Sci. Rep.*, vol. 11, no. 1, 2021, Art. no. 22459.
[23] K. Xie et al., "Deep adversarial tensor completion for accurate network traffic measurement," *IEEE/ACM Trans. Netw.*, vol. 31, no. 5, pp. 2101–2116, Oct. 2023.
[24] I. Markovsky, *Low Rank Approximation - Algorithms, Implementation, Applications*. Berlin, Germany: Springer, 2012.
[25] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218, 1936.
[26] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," 2008, *arXiv:0805.4471*.
[27] I. M. Al Jawarneh, P. Bellavista, L. Foschini, and R. Montanari, "Spatial-aware approximate Big Data stream processing," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
[28] A. Jawarneh, I. Mashhour, P. Bellavista, A. Corradi, L. Foschini, and R. Montanari, "Spatially representative online Big Data sampling for smart cities," in *Proc. 25th IEEE Int. Workshop Comput. Aided Model. Des. Commun. Links Netw.*, 2020, pp. 1–6.
[29] I. M. Al Jawarneh, P. Bellavista, A. Corradi, L. Foschini, and R. Montanari, "SpatialSSJP: QoS-aware adaptive approximate stream-static spatial join processor," *IEEE Trans. Parallel Distrib. Syst.*, vol. 35, no. 1, pp. 73–88, Jan. 2024.
[30] K. Xie et al., "Neural tensor completion for accurate network monitoring," in *Proc. 39th IEEE Conf. Comput. Commun.*, 2020, pp. 1688–1697.
[31] L. Ren, Y. Liu, D. Huang, K. Huang, and C. Yang, "MCTAN: A novel multichannel temporal attention-based network for industrial health indicator prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6456–6467, Sep. 2023.
[32] J. Feng et al., "User identity linkage via co-attentional neural network from heterogeneous mobility data," *IEEE Trans. Knowl Data Eng.*, vol. 34, no. 2, pp. 954–968, Feb. 2022.
[33] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[34] K. Xie, L. Wang, X. Wang, G. Xie, and J. Wen, "Low cost and high accuracy data gathering in WSNs with matrix completion," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1595–1608, Jul. 2018.
[35] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.
[36] C. L. Yu and X. Xi, "Tensor completion with nearly linear samples given weak side information," *ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 2, pp. 1–35, 2022.
[37] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust network compressive sensing," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 545–556.
[38] L. Wang et al., "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 683–694.
[39] L. Wang et al., "SPACE-TA: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 1–28, 2017.
[40] L. Deng, H. Zheng, X. Liu, X. Feng, and Z. Chen, "Network latency estimation with leverage sampling for personal devices: An adaptive tensor completion approach," *IEEE/ACM Trans. Netw.*, vol. 28, no. 6, pp. 2797–2808, Dec. 2020.
[41] Z. He, B. Zhao, and Z. Zhang, "Active sampling for accelerated MRI with low-rank tensors," in *Proc. 44th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2022, pp. 3024–3028.
[42] E. J. Candès et al., "Compressive sampling," in *Proc. Int. Congr. Math.*, 2006, pp. 1–20.
[43] R. H. Keshavan, S. Oh, and A. Montanari, "Matrix completion from a few entries," in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 324–328.
[44] Z. Jia, Q. Jin, M. K. Ng, and X.-L. Zhao, "Non-local robust quaternion matrix completion for large-scale color image and video inpainting," *IEEE Trans. Image Process.*, vol. 31, pp. 3868–3883, 2022.
[45] H. Li et al., "SGD-Tucker: A novel stochastic optimization strategy for parallel sparse tucker decomposition," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1828–1841, Jul. 2021.
[46] J. Xue, Y. Zhao, Y. Bu, J. C.-W. Chan, and S. G. Kong, "When laplacian scale mixture meets three-layer transform: A. parametric tensor sparsity for tensor completion," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13887–13901, Dec. 2022.
[47] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. Int. Conf. World Wide Web*, 2017, pp. 173–182.
[48] L. Deng, X. Liu, H. Zheng, X. Feng, and Z. Chen, "Graph-tensor neural networks for network traffic data imputation," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 3010–3024, Dec. 2023.
[49] L. Han, K. Zheng, L. Zhao, X. Wang, and H. Wen, "Content-aware traffic data completion in ITS based on generative adversarial nets," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11950–11962, Oct. 2020.
[50] L. Tan, X. Lin, D. Niu, D. Wang, M. Yin, and X. Zhao, "Projected generative adversarial network for point cloud completion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 2, pp. 771–781, Feb. 2023.

Huali Lu (Graduate Student Member, IEEE) received the BSc and MSc degrees from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2017 and 2020, respectively. She is currently working toward the PhD degree with the School of Computer Science and Engineering, Central South University, Changsha, China. Her researches mainly focus on spatial-temporal data mining, compact data collection, and trajectory similarity computing.

Feng Lyu (Senior Member, IEEE) received the BS degree in software engineering from Central South University, Changsha, China, in 2013, and the PhD degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2018. During respective 2018–2019 and 2016–2017, he worked as a postdoctoral fellow and was a visiting PhD student with BBCR Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently a professor with the School of Computer Science and Engineering, Central South University, Changsha, China. His research interests include vehicular networks, beyond 5G networks, Big Data measurement and application design, and edge computing. He is the recipient of the best paper award of IEEE ICC 2019. He currently serves as associate editor for *IEEE Systems Journal* and leading guest editor for *Peer-to- Peer Networking and Applications*, and served as TPC members for many international conferences. He is a member of the IEEE Computer Society, Communication Society, and Vehicular Technology Society.

Ju Ren (Senior Member, IEEE) received the BSc, MSc, and PhD degrees in computer science from Central South University, China, in 2009, 2012 and 2016, respectively. He is currently an associate professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include Internet-of-Things, edge computing, distributed and embedded AI, and operating system. He received the IEEE ComSoc Asia-Pacific Best Young Researcher Award in 2021. He is a member of the ACM. He was recognized as a highly cited researcher by Clarivate in 2020-2022.

Huaqing Wu (Member, IEEE) received the BE and ME degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014 and 2017, respectively, and the PhD degree from the University of Waterloo, Ontario, Canada, in 2021. She received the prestigious Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship Award, in 2021 and worked as a postdoctoral fellow with the Department of Electrical and Computer Engineering, MacMaster University, from 2021 to 2022. She is currently an assistant professor with the Department of Electrical and Software Engineering, University of Calgary, Alberta, Canada. Her current research interests include B5G/6 G, space-air-ground integrated networks, Internet of vehicles, edge computing/caching, and artificial intelligence (AI) for future networking. She received the Best Paper Award at IEEE GLOBECOM 2018, Chinese Journal on Internet of Things 2020, and IEEE GLOBECOM 2022.

Conghao Zhou (Member, IEEE) received the BEng degree from Northeastern University, Shenyang, China, in 2017, the MSc degree from the University of Illinois Chicago, Chicago, IL, USA, in 2018, and the PhD degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2022. He is currently a postdoctoral fellow with the University of Waterloo, Waterloo, ON, Canada. His research interests include space-air-ground integrated networks, network slicing, and machine learning for wireless networks.

Zhongyuan Liu is a student in the third year with the High School Affiliated to Renmin University of China. He is interested in spatial-temporal data mining and edge computing.

Yaoxue Zhang (Senior Member, IEEE) received the BSc degree from the Northwest Institute of Telecommunication Engineering, China, in 1982, and the PhD degree in computer networking from Tohoku University, Japan, in 1989. Currently, he is a professor with the Department of Computer Science and Technology, Tsinghua University, China and also a professor with the School of Computer Science and Engineering, Central South University, China. His research interests include computer networking, operating systems, ubiquitous/pervasive computing, transparent computing, and Big Data. He has published more than 200 technical papers in international journals and conferences, as well as 9 monographs and text-books. Currently, he is serving as the editor-in-chief of *Chinese Journal of Electronics*. He is a fellow of the Chinese Academy of Engineering.

Xuemin (Sherman) Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is a university professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks. He is a registered professional engineer of Ontario, Canada, an Engineering Institute of Canada fellow, a Canadian Academy of Engineering fellow, a Royal Society of Canada fellow, a Chinese Academy of Engineering foreign member, and a distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society. He received "West Lake Friendship Award" from Zhejiang Province, in 2023, President's Excellence in Research from the University of Waterloo, in 2022, the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT), in 2021, the R.A. Fessenden Award, in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario), in 2019, James Evans Avant Garde Award, in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award, in 2015 and Education Award, in 2017 from the IEEE Communications Society (ComSoc), and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award, in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA), in 2003 from the Province of Ontario, Canada. He serves/served as the general chair for the 6G Global Conference'23, and ACM Mobihoc'15, Technical Program Committee chair/co-chair for IEEE Globecom'24, 16 and 07, IEEE Infocom'14, IEEE VTC'10 Fall, and the chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the president of the IEEE ComSoc. He was the vice president for Technical & Educational Activities, vice president for Publications, member-at-large on the Board of Governors, chair of the Distinguished Lecturer Selection Committee, and member of IEEE Fellow Selection Committee of the ComSoc. He served as the editor-in-chief of *IEEE Internet of Things Journal*, *IEEE Network*, and *IET Communications*.