ZTFed-MAS2S: A Zero-Trust Federated Learning Framework with Verifiable Privacy and Trust-Aware Aggregation for Wind Power Data Imputation

Yang Li, *Senior Member, IEEE*, Hanjie Wang, Yuanzheng Li, *Senior Member, IEEE*, Jiazheng Li, Zhaoyang Dong, *Fellow, IEEE*

***Abstract*—Wind power data often suffers from missing values due to sensor faults and unstable transmission at edge sites. While federated learning enables privacy-preserving collaboration without sharing raw data, it remains vulnerable to anomalous updates and privacy leakage during parameter exchange. These challenges are amplified in open industrial environments, necessitating zero-trust mechanisms where no participant is inherently trusted. To address these challenges, this work proposes ZTFed-MAS2S, a zero-trust federated learning framework that integrates a multi-head attention-based sequence-to-sequence imputation model. ZTFed integrates verifiable differential privacy with non-interactive zero-knowledge proofs and a confidentiality and integrity verification mechanism to ensure verifiable privacy preservation and secure model parameters transmission. A dynamic trust-aware aggregation mechanism is employed, where trust is propagated over similarity graphs to enhance robustness, and communication overhead is reduced via sparsity- and quantization-based compression. MAS2S captures long-term dependencies in wind power data for accurate imputation. Extensive experiments on real-world wind farm datasets validate the superiority of ZTFed-MAS2S in both federated learning performance and missing data imputation, demonstrating its effectiveness as a secure and efficient solution for practical applications in the energy sector.**

***Index Terms*—wind power data imputation, federated learning, zero-trust architecture, verifiable differential privacy, trust-aware aggregation, multi-head attention mechanism.**

NOMENCLATURE

ZTFed -MAS2S	zero-trust federated learning framework with a multi-head attention-based sequence-to-sequence imputation model
FL	federated learning
DP	differential privacy
MA	multi-head attention
NIZK	non-interactive zero-knowledge proofs
CIV	Confidentiality and Integrity Verification
DTAA	Dynamic Trust-Aware Aggregation
Bi-LSTM	bidirectional long short-term memory
IIoT	industrial internet of things scenarios
GAIN	generative adversarial imputation nets
FHE	fully homomorphic encryption
TSS	threshold secret sharing
T-Mean	Trimmed Mean
MAAPE	arctangent absolute percentage error
NREL	National Renewable Energy Laboratory
AES-CBC	Advanced Encryption Standard in Cipher Block Chaining mode
HMAC	Hash-based Message Authentication Code
MAD	median absolute deviation
MIA	membership inference attacks

I. Introduction

TODAY, as global energy demand grows and the emphasis on sustainable environmental development intensifies, the advancement and integration of renewable energy, particularly wind power, have become key trends. Wind power generation, the most mature and broadly commercialized form of renewable energy, significantly impacts the power system as it advances and integrates into the grid, due to its randomness, volatility, and intermittency [1]. Historical operational data from wind farms is crucial for studying wind power dynamics, improving forecasting accuracy, assessing its impact on the grid, and developing effective control strategies [2]–[4]. However, during the data collection, transmission, and conversion processes, especially at network edge wind farms, frequent data missing occurs due to sensor failures, transmission problems, and noise interference. This degrades measurement quality, complicates data analysis, and impacts power grid operations. Furthermore, how to achieve secure and efficient data sharing while ensuring privacy for wind power missing data imputation is increasingly becoming a crucial and urgent topic [5].

A. Literature Review

Based on previous research on wind data imputation, existing methods can be categorized into three types: statistical, traditional machine learning, and deep learning. Statistical methods include value interpolation and modeling. Mean imputation offers a simple implementation. Modeling approaches such as autoregressive (AR) [6] and expectation-maximization (EM) [7] assume specific data distributions for imputation. Traditional machine learning methods, including k-nearest neighbors (kNN) [8] and Missforest [9], build predictive models based on existing features and are effective in handling nonlinear relationships and large-scale data.

While traditional methods have made progress, deep learning techniques have demonstrated even greater potential. For

Y. Li and H. Wang are with the School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China (e-mail: liyang@neepu.edu.cn; wanghanjie1@163.com).

Y. Z. Li is with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (email: Yuanzheng_Li@hust.edu.cn).

J. Z. Li is with the State Grid Xiamen Electric Power Supply Company, Xiamen 361004, China (email: tytrytytgr@gmail.com).

Z. Y. Dong is with the Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong, (email: zydong@cityu.edu.hk).

example, reference [10] proposes a super-resolution perception convolutional neural network that integrates super-resolution techniques to reconstruct missing data from low-resolution inputs; reference [11] introduces a multiscale spatiotemporal Transformer model for industrial process imputation; reference [12] develops an attention-based network leveraging polynomial regression for solar photovoltaic power data; reference [13] proposes an adaptive multi-head self-attention-based variational auto-encoder model for imputation; reference [14] designs an unsupervised conditional generative adversarial network model using data-mining techniques for energy data imputation; and reference [15] puts forward an integrated long short-term memory (LSTM) model for missing data tolerance.

Despite progress in missing data imputation, challenges remain in capturing complex dependencies, long-sequence features, and dynamic variation patterns. Traditional CNN- and RNN-based models struggle with long-range dependencies and generalization, particularly on large-scale, high-dimensional data. In parallel, centralized learning methods, though benefiting from aggregated data, face practical issues such as privacy risks in cross-farm data integration and high communication overhead between edge nodes and central servers.

Federated learning (FL) enables collaborative model training by sharing parameters instead of raw data, mitigating privacy and communication challenges compared to centralized approaches [16]. Consequently, it has been widely adopted in industrial Internet of Things (IIoT) scenarios [17]–[19]. However, existing FL methods still face notable limitations. First, model parameters may exhibit anomalies during aggregation, such as integrity violations from network issues or adversarial manipulations like backdoor attacks, compromising system performance and security. To address this, robust aggregation mechanisms such as MultiKrum [20], Trimmed Mean (T-mean) [21], and Median [22] have been employed. Yet, these rely on fixed thresholds (e.g., adversary ratios, trimming rates), often excluding benign clients or failing to detect sophisticated anomalies. Second, despite avoiding raw data sharing, FL still faces privacy risks during parameter transmission. Techniques such as differential privacy (DP), fully homomorphic encryption (FHE) [23], and threshold secret sharing (TSS) [24] attempt to address these risks via noise injection or encryption, but rely on the assumption of honest client behavior—often unrealistic in practice. In open and decentralized settings, these challenges highlight the need for zero-trust (ZT) mechanisms, where no entity is inherently trusted [25].

In summary, there are three key gaps in wind power missing data imputation:

1) A distributed learning approach ensuring privacy preservation and verifiability is needed under a ZT architecture to address the privacy risks and high communication overhead associated with edge-based wind farms.

2) The lack of advanced models capable of capturing long-time dependencies and dynamic changes in time series data for missing data imputation, particularly for wind power data.

3) Existing methods for constructing missing data scenarios often consider discrete or continuous missing patterns individually, but seldom address the hybrid of both simultaneously. This leads to difficulties in capturing the complexity of the coexistence of patterns in real-world scenarios.

B. Contribution of This Paper

This paper proposes ZTFed-MAS2S, a zero-trust federated learning framework with a multi-head attention-based sequence-to-sequence imputation model for wind power data. The key contributions are as follows:

1) The ZTFed framework integrates verifiable Differential Privacy with Non-Interactive Zero-Knowledge Proofs (DP-NIZK) and a Confidentiality and Integrity Verification (CIV) mechanism to enable verifiable privacy preservation and secure, integrity-assured model transmission. In addition, it employs a Dynamic Trust-Aware Aggregation (DTAA) mechanism to enhance resilience against anomalous clients and incorporates sparsity- and quantization-based compression to reduce communication overhead.

2) The MAS2S imputation model leverages wind features as inputs to provide richer contextual information. It incorporates a bidirectional long short-term memory (Bi-LSTM) encoder and an LSTM decoder, enhanced by a multi-head attention (MA) mechanism. The MA mechanism in the decoder effectively captures the correlations between wind power and other features, thereby enhancing the accuracy of the final output.

3) To simulate hybrid missing patterns in missing data scenarios, we propose a hybrid masking strategy. By designing a missing matrix, this strategy precisely coordinates discrete missing points and continuous missing intervals, enabling more realistic simulation of complex real-world scenarios.

4) Extensive experiments on real-world wind farm datasets were conducted, including comparisons with state-of-the-art privacy-preserving federated learning frameworks—covering both privacy and aggregation mechanisms—as well as with traditional imputation methods. To the best of our knowledge, this is the first FL-based imputation framework to integrate verifiable differential privacy and dynamic trust-aware aggregation under a zero-trust architecture.

II. Methods

A. MAS2S Imputation Model

The key feature of the sequence-to-sequence model is its end-to-end learning [26], which directly maps input sequences to output sequences without relying on observation windows or prediction windows. This makes it ideal for multivariate time series analysis [27]. Fig. 1 illustrates its structure.

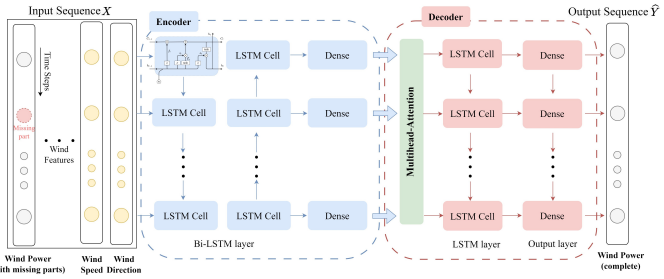

Fig. 1. The basic structure of the MAS2S.

As shown in the figure, the input sequence X and the output sequence $\widehat{Y}$ are represented as follows:

$$X = \{x_1, \ldots, x_t, \ldots, x_T\}, \quad \widehat{Y} = \{\widehat{y_1}, \ldots, \widehat{y_t}, \ldots, \widehat{y_T}\} \quad (1)$$

where T represents the total number of time steps, x_t denotes the input feature vector at time step t, which includes the wind power data with missing parts and other wind features (e.g., wind speed and wind direction), $\widehat{y}$ represents the output wind power data, with missing parts imputed.

The MAS2S adopts an encoder-decoder architecture, where a Bi-LSTM encoder maps the input time series to a semantic vector capturing global context, and a decoder with MA and LSTM generates the target sequence based on feature correlations.

1) BiLSTM Encoder: In each t, the LSTM maintains its internal hidden state h, which results in a hidden sequence of $\{h_1, h_2, \ldots, h_T\}$. The hidden vector h_t is updated as follows:

$$f_t = sigmoid\left(W_f\left[h_{t-1} \parallel x_t\right] + b_f\right), \quad (2)$$

$$i_t = sigmoid\left(W_i\left[h_{t-1} \parallel x_t\right] + b_i\right), \quad (3)$$

$$o_t = sigmoid\left(W_o\left[h_{t-1} \parallel x_t\right] + b_o\right), \quad (4)$$

$$\widetilde{C_t} = \tanh\left(W_C\left[h_{t-1} \parallel x_t\right] + b_C\right), \quad (5)$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t}, \quad h_t = o_t * \tanh\left(C_t\right). \quad (6)$$

Here, f_t, i_t, and o_t represent the forget gate, input gate, and output gate, respectively; W_f, W_i, and W_o denote the weight matrices; b_f, b_i, and b_o are the bias vectors; $\widetilde{C_t}$ and C_t represent the cell vector and the candidate cell vector, respectively; h_t refers to the encoder hidden state; the symbol $\parallel$ denotes concatenation.

In a BiLSTM, rather than generating a single sequence of hidden states, it computes both forward and backward hidden states, denoted by $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, respectively. These are subsequently concatenated to form the hidden state:

$$h_t = [\overrightarrow{h_t} \parallel \overleftarrow{h_t}]. \quad (7)$$

2) LSTM Decoder with Multi-Head Attention: To enhance the decoder's ability to capture temporal dependencies, we employ a MA mechanism. The decoder's previous hidden state s_{t-1} and the encoder hidden states $\{h_1, \ldots, h_T\}$ are projected into N_h parallel subspaces.

For each attention head $m \in \{1, \ldots, N_h\}$, we compute:

$$Q^m = W_Q^m s_{t-1}, \quad K^m = W_K^m h_i, \quad V^m = W_V^m h_i, \quad (8)$$

where Q^m, K^m, and V^m are the query, key, and value vectors, and W_Q^m, W_K^m, and W_V^m are the corresponding projection matrices.

The attention weights $\alpha_{t,i}^m$ are computed via scaled dot-product attention:

$$\alpha_{t,i}^m = \text{softmax}\left(\frac{Q^m (K^m)^\top}{\sqrt{d_k}}\right), \quad (9)$$

where d_k is the key dimension. The head-specific context vector $c_t^{a,m}$ is obtained as a weighted sum of the value vectors:

$$c_t^{a,m} = \sum_{i=1}^{T} \alpha_{t,i}^m V^m. \quad (10)$$

Finally, the outputs of all heads are concatenated and linearly projected to form the final context vector c_t^a:

$$c_t^a = W_O \cdot [c_t^{a,1} \parallel \cdots \parallel c_t^{a,N_h}]. \quad (11)$$

where W_O is a projection matrix. The decoder updates its hidden state s_t and output $\hat{y}_t$ as:

$$s_t = \text{LSTM}(c_t^a \parallel s_{t-1}), \quad \hat{y}_t = \text{Dense}(W_d \cdot s_t + b_d). \quad (12)$$

To maintain the consistency and integrity of the data, we considered both the missing and non-missing parts of the output sequence for correction, defining the loss function for the entire sequence:

$$\mathcal{L}(\theta) = \frac{1}{T} \sum_{t=1}^{T} |y_t - \widehat{y_t}|, \quad (13)$$

where $\mathcal{L}(\cdot)$ is the loss function, y_t represents the ground truth at time step t, and θ represents the model parameters.

The model update process can be expressed as:

$$\theta \leftarrow \theta - \eta \cdot Adam(\theta, \nabla\mathcal{L}(\theta), \beta_1, \beta_2), \quad (14)$$

where η is the learning rate; β_1 and β_2 are the momentum parameters; θ represents the MAS2S imputation model parameters; and $\nabla\mathcal{L}(\theta)$ denotes the gradient of the loss function.

The overall algorithmic flow of the MAS2S imputation model is shown in Algorithm 1.

Algorithm 1 MAS2S imputation model

Require: Training epoch I_l and its index i_l; input sequence X; ground truth sequence Y.
Ensure: MAS2S imputation model with initialized parameters θ.

```
for epoch i_l = 1, 2, ..., I_l do
    for batch of input sequences X do
        Run encoding by feeding sequence X into Bi-LSTM according to (2)-(7).
        Obtain the entire hidden state sequence {h_1, h_2, ..., h_T}.
        Run decoding by computing attention vector c_t^a according to (8)-(11).
        Obtain the output sequence Ŷ according to (12).
        Calculate the loss according to (13).
        Update parameters according to (14).
    end for
end for
```

B. Zero-Trust Federated Learning Framework

1) Differential Privacy with Non-Interactive Zero-Knowledge Proofs: To achieve both privacy preservation and verifiability in FL, we propose DP-NIZK. DP protects client data by adding noise to model parameters. NIZK enables the server to verify the correct application of DP noise without disclosing sensitive information [28].

DP guarantees that for any adjacent datasets $\mathcal{D}_i$ and $\mathcal{D}_i'$, and for any measurable set $\mathcal{S}$ in the output space, the randomized mechanism $\mathcal{M}$ satisfies:

$$\Pr[\mathcal{M}(\mathcal{D}_i) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}_i') \in \mathcal{S}] + \delta, \quad (15)$$

where δ represents the probability of privacy leakage, ϵ is the privacy budget, and $\Pr[\cdot]$ denotes the probability over the randomness of $\mathcal{M}$. To achieve global (ϵ, δ)-DP, we employ

the Gaussian mechanism [29]. The local model parameters θ_i are first clipped to bound their ℓ_2-norm:

$$\bar{\theta}_i = \theta_i / \max(1, \|\theta_i\|_2 / \tau_c), \tag{16}$$

where $\bar{\theta}_i$ denotes the clipped parameters of client i, and τ_c is the clipping threshold. Gaussian noise $\mathbf{n}_i$ is then added:

$$\widetilde{\theta}_i = \bar{\theta}_i + \mathbf{n}_i \sim \mathcal{N}(0, \sigma^2). \tag{17}$$

Here, $\widetilde{\theta}_i$ denotes the perturbed model parameters, and the standard deviation σ of the Gaussian noise is determined by:

$$\sigma = \frac{\sqrt{2\ln(1.25/\delta)} \cdot \Delta s}{\epsilon} \times \frac{T_g}{K}. \tag{18}$$

Here, T_g denotes the global epoch, and K is the synchronization interval. The global sensitivity Δs is defined as the maximum local sensitivity across all clients:

$$\Delta s = \max\{\Delta s_i\}, \quad \forall i, \tag{19}$$

where Δs_i denotes the local sensitivity of client i, computed as the maximum change in the local model output when the dataset changes from $\mathcal{D}_i$ to $\mathcal{D}_i'$, with the local model output defined as $s^{\mathcal{D}_i} = \arg\min_{\bar{\theta}_i} \mathcal{L}(\bar{\theta}_i, \mathcal{D}_i)$. The local sensitivity is:

$$\Delta s_i = \max_{\mathcal{D}_i, \mathcal{D}_i'} \left\| s^{\mathcal{D}_i} - s^{\mathcal{D}_i'} \right\|_2 = 2\tau_c / |\mathcal{D}_i|, \tag{20}$$

After determining the σ, the client deterministically generates Gaussian noise from a private seed $s \in \mathbb{Z}_q$, which serves as the secret witness. To construct the NIZK, the client performs three steps—commitment, challenge, and response—starting with computing commitments using s and a random nonce $k \in \mathbb{Z}_q$:

$$h_s = g^s \bmod p, \quad t_k = g^k \bmod p. \tag{21}$$

Here, h_s and t_k denote the public and temporary commitments, respectively. The parameters p and q are large primes, and $g \in \mathbb{Z}_p^*$ is a generator of a subgroup of order q.

Next, the challenge c_s is autonomously derived by hashing the commitments and parameters using a collision-resistant hash function $H(\cdot)$:

$$c_s = H(t_k \parallel H(\widetilde{\theta}_i) \parallel H(\theta_i)) \bmod q. \tag{22}$$

With the challenge determined, the client computes the corresponding response r_s:

$$r_s = (k + c_s \cdot s) \bmod q. \tag{23}$$

Finally, the client uploads the NIZK proof $p_f = \{t_k, r_s, c_s, h_s, H(\widetilde{\theta}_i), H(\theta_i)\}$ along with $\widetilde{\theta}_i$. Upon reception, the server verifies whether:

$$g^{r_s} \bmod p = t_k \cdot h^{c_s} \bmod p. \tag{24}$$

If verification succeeds, it confirms correct noise application, and the server accepts $\widetilde{\theta}_i$ as valid model parameters for aggregation. Otherwise, the parameters are deemed untrustworthy and excluded.

2) Communication-Efficient and Secure Transmission: To reduce communication overhead while ensuring privacy-preserving and verifiable model parameter transmission, we propose a framework that integrates sparsity- and quantization-based compression with a Confidentiality and Integrity Verification (CIV) mechanism.

The transmitted parameters $\theta^c \in \{\widetilde{\theta}, \theta_g\}$, where θ_g denotes the aggregated global model parameters, are compressed as:

$$\theta^c = Q_b\left(S_p(\theta^c) \odot M^{\text{com}}\right), \tag{25}$$

where θ^c denotes the compressed parameters; $S_p(\cdot)$ is a sparsification function that selects the top-$p\%$ entries with the largest magnitudes; $Q_b(\cdot)$ quantizes the selected values into b-bit integers; M^{com} is a binary mask indicating the retained positions; $\odot$ denotes the Hadamard product.

Then, θ^c is encrypted using a symmetric key k_{sym}, which is assumed to be securely established under a zero-trust architecture. To ensure integrity, a Hash-based Message Authentication Code (HMAC) is appended:

$$\mathcal{M}_{\text{enc}} = \text{AES-CBC}(k_{\text{sym}}, \theta^c) \parallel \text{HMAC}(\theta^c), \tag{26}$$

where AES-CBC stands for the Advanced Encryption Standard in Cipher Block Chaining mode, and $\mathcal{M}_{enc} \in \{\mathcal{M}_{enc}^U, \mathcal{M}_{enc}^D\}$ denotes the encrypted transmission message, where $\mathcal{M}_{enc}^U$ and $\mathcal{M}_{enc}^D$ refer to the client's encrypted upload and download messages, respectively.

Upon receiving $\mathcal{M}_{\text{enc}}$, decryption with k_{sym} yields θ^c, and integrity is verified by recomputing the HMAC. If successful, the original parameter is reconstructed as:

$$\theta = Q_b^{-1}(\theta^c). \tag{27}$$

3) Dynamic Trust-Aware Aggregation: We propose DTAA to enable robust aggregation under a ZT architecture. DTAA estimates inter-client trust through similarity analysis and graph-based propagation in each aggregation round, and then selects clients for aggregation based on trust scores.

First, the cosine similarity between client model parameters is computed as:

$$S_{i,j} = \cos(\tilde{\theta}_i, \tilde{\theta}_j) = \frac{\tilde{\theta}_i \cdot \tilde{\theta}_j}{\|\tilde{\theta}_i\|_2 \cdot \|\tilde{\theta}_j\|_2}, \quad i \neq j, \tag{28}$$

where $S_{i,j}$ indicates the similarity between clients i and j. To enhance contrast, we apply a power transformation $T_{i,j}^s = (S_{i,j})^r$ where $r > 0$ is a sharpening coefficient, and $T_{i,j}^s \in [0, 1]$ is the normalized trust score.

Then, a sparse trust graph $G' = (V, E', T^s)$ is constructed, where V is the client set, and $E' \subseteq V \times k_g$ contains directed edges from each client $i \in V$ to its top-k_g most trusted neighbors based on $T_{i,j}^s$. This graph defines a weighted adjacency matrix $A \in \mathbb{R}^{N \times N}$ as:

$$A_{i,j} = \begin{cases} T_{i,j}, & \text{if } (i \to j) \in E', \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

The initial trust score of client i is then computed as:

$$\mathbf{t}_i^{(0)} = \frac{\sum_j A_{i,j}}{\sum_{i,j} A_{i,j}}. \tag{30}$$

Trust scores are iteratively updated via propagation with damping:

$$\mathbf{t}_i^{(t+1)} = (1-d)A\mathbf{t}_i^{(t)} + d\mathbf{t}_i^{(0)}, \tag{31}$$

where $d \in (0,1)$ is the damping factor. The process converges when:

$$\left|\mathbf{t}^{(t+1)} - \mathbf{t}^{(t)}\right| < \tau_t, \tag{32}$$

where $\tau_t > 0$ is a convergence threshold, and $\mathbf{t}^{(t)} = [\mathbf{t}_1^{(t)}, \ldots, \mathbf{t}_N^{(t)}]^\top \in \mathbb{R}^N$ denotes the trust scores of all clients at iteration t.

To enhance robustness, an anomaly detection step based on the median absolute deviation (MAD) is introduced:

$$\text{MAD} = \text{Median}(|\mathbf{t}_i - \text{Median}(\mathbf{t})|), \tag{33}$$

Based on this, clients with $\mathbf{t}_i < \text{Median}(\mathbf{t}) - k_m \cdot \text{MAD}$ are excluded, where k_m is a tuning coefficient.

Finally, the aggregation is performed in a layer-wise manner, applying a median operation over the selected client set V_s:

$$\theta_g = \text{Median}(\{\widetilde{\theta}_i \mid i \in V_\text{s}\}). \tag{34}$$

III. ZTFed-MAS2S For Wind Power Imputation

A. Problem Formulation

For the input sequence X, it can be partitioned into an observed part X_obs containing all the observed data components, and a missing part X_miss that includes all the missing data, i.e., $X = \{X_\text{obs}, X_\text{miss}\}$. The objective is to derive a complete sequence $\widehat{Y}$ where the missing entries are imputed using the observed data. This task can be formalized as follows:

$$\widehat{Y} = impute(X_\text{miss}|X_\text{obs}), \tag{35}$$

where, $impute(\cdot)$ represents an imputation method.

B. Simulation of Missing Scenarios

The construction of missing scenarios is driven by two factors: missing rate (proportion of absent values) and missing patterns (continuous or discrete), where discrete missing refers to missing completely at random. To emulate real-world challenges in wind farm data, our method combines these factors using a hybrid masking strategy to obtain the deteriorated data. As illustrated in Fig. 2, this process includes three key steps:

First, a total missing rate mr_t is defined for the complete wind power sequence $X^w = \{x_1^w, \ldots, x_t^w, \ldots, x_T^w\}$. A tunable ratio parameter $r_m \in [0,1]$ is then introduced to partition mr_t into continuous and discrete components. Specifically:

$$\begin{cases} mr_d = r_m \cdot mr_t, \\ mr_c = (1-r_m) \cdot mr_t, \end{cases} \tag{36}$$

where mr_d and mr_c represent the discrete and continuous missing rates, respectively.

Next, the continuous missing mask matrix M_c and the discrete missing mask matrix M_d are generated using random sampling based on the missing rates mr_c and mr_d, and then they are concatenated as a hybrid mask matrix M_h. In these matrices, 0 indicates complete data; 1 indicates missing data, with green and yellow denoting continuous and discrete patterns.

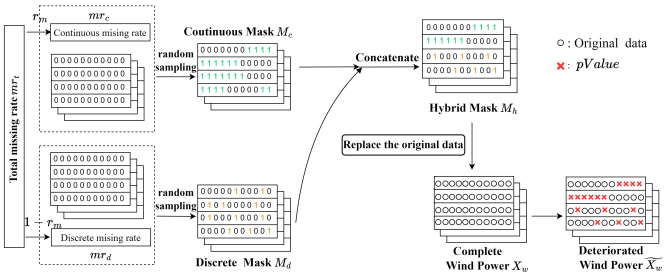

Fig. 2. Construction of missing scenarios using hybrid masking strategy.

Finally, missing positions are identified by M_h, missing values are injected into X_w to generate the deteriorated incomplete wind power sequence $\widetilde{X_w}$:

$$\widetilde{X^w} = X^w \odot (1 - M_h) + pValue \cdot M_h, \tag{37}$$

where $pValue$ is a placeholder value, ensuring true zeros (e.g., no wind power generation) are not confused with missing data.

C. Data Preprocessing

To eliminate scale differences between variables and fairly account for the influence of each wind feature on wind power, all features are normalized using min-max normalization as:

$$x_t^{j'} = \frac{x_t^j - \min(x^j)}{\max(x^j) - \min(x^j)}, \tag{38}$$

where x_t^j is the original data of the j-th feature at time step t, and $x_t^{j'}$ is the normalized value.

D. ZTFed-MAS2S Architecture

Based on the theoretical analysis above, the ZTFed-MAS2S method, illustrated in Fig. 3, is developed to impute wind power missing data in wind farms. Further details are provided in Algorithm 2.

IV. Case Studies

A. Experimental Setup

The wind energy datasets used for simulations are real-world datasets from the National Renewable Energy Laboratory (NREL) [30], comprising 15-minute interval measurements from 16 onshore wind farms in Washington State (2007–2013). It includes wind power, air density, temperature, wind direction, wind speed, and surface air pressure. Each sample has a sequence length of 96, and following [31], the data are split into 80% training, 10% validation, and 10% testing. All methods are implemented and evaluated on the same hardware platform: an Intel Core i9-10900K CPU, NVIDIA GeForce RTX 3090 GPU, and 32 GB RAM. Experiments are conducted using Python 3.8.18 and TensorFlow 2.6.0.

For fair comparison, baseline methods are evaluated across privacy protection and communication efficiency, aggregation, and imputation. For privacy protection and communication efficiency, we consider standard DP (with τ_c set to the 95th percentile of parameter norms [32] and δ set to 1×10^{-4}), FHE (using the Cheon-Kim-Kim-Song scheme), and TSS (with

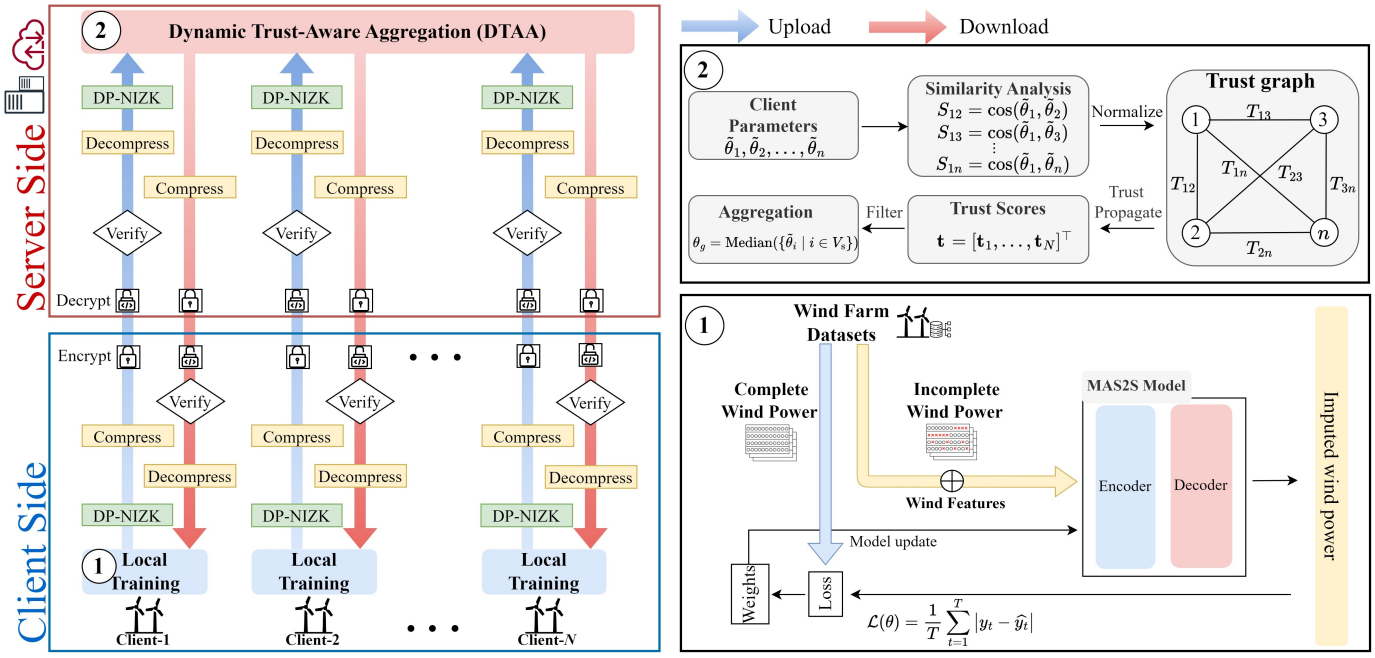

Fig. 3. The ZTFed-MAS2S method.

Algorithm 2 ZTFed-MAS2S

Require: Global epoch T_g and its index t_g; I_l and i_l; Total client N and n; Selected client N_e and n_e; $\mathbf{n}$; K; Client participation rate E; local models with parameters $\{\theta_i\}_{i=1}^{N}$; Global parameters θ_g on the server.

for epoch $t_g = 1, 2, \ldots, T_g$ **do**
  Randomly choose N_e clients from all clients N using E.
  Clients execute:
  for each selected client n_e **in parallel do**
    for local training epoch $i_l = 1, 2, \ldots, I_l$ **do**
      Update parameters θ_{n_e} using the MAS2S.
    end for
  end for
  if $t_g \bmod K = 0$ **then**
    Clip parameters $\{\theta_{n_e}\}_{n_e=1}^{N_e}$, add noises $\mathbf{n}$ to the parameters using (16), (17), and generate NIZK proofs p_f according to (20)-(22).
    Compress and then encrypt the perturbed parameters $\{\widetilde{\theta}_{n_e}\}_{n_e=1}^{N_e}$ according to (24)-(26).
    Upload the $\mathcal{M}_{\text{enc}}^{U}$ with the p_f to the server.
    Server executes:
    Decrypt and decompress the $\mathcal{M}_{\text{enc}}^{U}$ according to (26).
    if HMAC and NIZK proofs are valid **then**
      Aggregate $\{\widetilde{\theta}_{n_e}\}_{n_e=1}^{N_e}$ using DTAA and obtain the θ_g according to (27)-(33).
      Compress and encrypt the θ_g according to (24)-(25).
    end if
    Clients execute:
    for each client n **in parallel do**
      Download the $\mathcal{M}_{\text{enc}}^{D}$ from the server and decrypt it.
      if HMAC is valid **then**
        Decompress and update the local models using θ_g:

$$\{\theta_i\}_{i=1}^{N} \leftarrow \theta_g$$

      end if
    end for
  end if
end for

3-share splitting). Aggregation mechanisms include FedAvg, MultiKrum (with an adversary ratio of 0.3), and T-Mean (with a trimming rate of 0.1). For imputation, we benchmark statistical (Mean, EM), traditional machine learning (Missforest, kNN), and deep learning (GAIN, Bi-LSTM, Transformer) methods. Detailed imputation configurations are provided in Table I.

Unless otherwise specified, all experiments adopt a unified FL configuration: synchronization interval $K = 10$, global epochs $T_g = 100$, local epochs $I_l = 20$, total clients $N = 16$, and client participation rate $E = 50\%$. Model parameters are compressed with a sparsity ratio of 0.7 and quantized to 4 bits. For DTAA, the number of trusted neighbors is set to $k_g = 3$, the tuning coefficient is set to $k_m = 3$, the sharpening coefficient is set to $r = 2$, and the convergence threshold is set to $\tau_t = 10^{-4}$.

B. Evaluation Metrics

To comprehensively evaluate the proposed framework, we adopt the following metrics:

1) Imputation Accuracy: We evaluate imputation performance using three standard metrics: mean absolute error (MAE), root mean square error (RMSE), and mean arctangent absolute percentage error (MAAPE). MAAPE mitigates the impact of extreme relative errors, especially when ground truth values are near zero [33]. It is defined as:

$$\text{MAAPE} = \frac{1}{n}\sum_{i=1}^{n} \arctan\left(\left|\frac{y_i - \widehat{y}_i}{y_i}\right|\right). \tag{39}$$

Here, n is the number of missing sample points in the test set, $\widehat{y}_i$ is the imputed value (only evaluated at missing positions), and y_i is the corresponding ground truth.

TABLE I
COMPARATIVE ANALYSIS SETTINGS

Methods	Parameters	Value/Configuration
MAS2S	MA_heads	2
	Key_dim	32
	Optimizer	Adam
	Neurons	128
	Learning rate	0.001
Bi-LSTM	Neurons	128
	Optimizer	Adam
	Learning rate	0.001
Transformer	MA_heads	4
	Key_dim	64
	Ff_dim	256
	Optimizer	Adam
	Learning rate	0.001
GAIN	Optimizer	Adam
	Learning rate	0.0002
	Hint rate	0.99
	Noise	0.0-0.1
kNN	N_neighbors	5
	Weights_initializer	Uniform
Missforest	Max_iter	10
	N_estimators	100
EM	Loops	50
	Tolerance	10%

Additionally, RMSE-based sensitivities are introduced to evaluate the methods' robustness [34], detailed as follows:

$$S_{mr,rmse} = \frac{\sum_{i=1}^{n_m} \left|(mr_i - \overline{mr})(rmse_i - \overline{rmse})\right|}{\sum_{i=1}^{n} (mr_i - \overline{mr})^2}, \tag{40}$$

$$S_{prC,rmse} = \frac{\sum_{i=1}^{n_c} \left|(prC_i - \overline{prC})(rmse_i - \overline{rmse})\right|}{\sum_{i=1}^{n_c} (prC_i - \overline{prC})^2}, \tag{41}$$

where prC_i denotes the proportion of continuous missing in the hybrid missing pattern; $S_{prC,rmse}$ is the RMSE-based sensitivity under varying missing rates; $S_{mr,rmse}$ is the RMSE-based sensitivity under varying hybrid missing patterns; n_m and n_c indicate the number of mr and prC, respectively.

2) Privacy Protection and Model Utility: DP limits the influence of individual samples on model outputs, reducing the risk of membership inference attacks (MIA) [35]. To evaluate the privacy protection offered by DP, we adopt a black-box MIA threat model [36], assuming that the attacker has access to the global model $f_{\theta_g}(\cdot)$. The attacker aims to infer the membership status of target samples $\{(X_a, Y_a)\}_{a=1}^{A}$ by exploiting model predictions.

For each target sample (X_a, Y_a), the attacker computes the prediction error function $\mathcal{L}_a$ as:

$$\mathcal{L}_a(X_a, Y_a; \theta_g) = \frac{1}{T} \sum_{t=1}^{T} \left| f_{\theta_g}(X_{a,t}) - Y_{a,t} \right|, \tag{42}$$

where $(X_{a,t}, Y_{a,t})$ indicates the input-output pair at time step t within sample a. The attacker infers the membership status $\hat{m}_a$ of sample a by comparing the error to a threshold τ_m:

$$\hat{m}_a = \begin{cases} 1, & \text{if } \mathcal{L}_a(X_a, Y_a; \theta_g) < \tau_m, \\ 0, & \text{if } \mathcal{L}_a(X_a, Y_a; \theta_g) \geq \tau_m, \end{cases} \tag{43}$$

where $\hat{m}_a = 1$ indicates that the attacker classifies the sample as a member, and $\hat{m}_a = 0$ as a non-member. The overall attack performance is evaluated by the MIA success rate (MIA-SR), defined as:

$$\text{MIA} - \text{SR} = \frac{1}{A} \sum_{a=1}^{A} 1[\hat{\text{m}}_a = \text{m}_a] \times 100\%, \tag{44}$$

where $m_a \in \{0, 1\}$ denotes the true membership label of the a-th sample, and A is the total number of target samples considered in the evaluation.

To assess the impact of DP on model performance, a utility metric is defined as:

$$\text{Utility} = \frac{1 - \text{MAAPE}}{1 - \text{MAAPE(no DP)}} \times 100\%. \tag{45}$$

3) Communication Overhead: In this work, the communication overhead is defined as the total amount of data uploaded and downloaded by clients across all communication rounds, measured in megabytes (MB). Specifically, the communication overhead (CO) in our method is computed as:

$$\text{CO} = \sum_{r=1}^{R} \left(\sum_{n_{e=1}}^{N_e} \left(\mathcal{M}_{enc,n_e}^{U,r} + p_{f,n_e}^{r} \right) + N \times \mathcal{M}_{enc}^{D,r} \right) \tag{46}$$

where R is the total number of communication rounds, computed as $R = T_g/K$, with r denoting the round index. N_e and n_e denote the number of clients selected in each communication round and the index of a selected client, respectively. $\mathcal{M}_{\text{enc},n_e}^{U,r}$ is the encrypted upload message from client n_e in round r, while $\mathcal{M}_{\text{enc}}^{D,r}$ is the encrypted download message from the server.

C. Performance Evaluation on Privacy Protection and Communication Efficiency

To evaluate the trade-off between DP protection and model performance, we used 500 each of member and non-member samples as MIA attack targets. As shown in Table II, with the privacy leakage probability $\delta = 1 \times 10^{-4}$ fixed, a smaller ε leads to a lower MIA success rate, indicating stronger privacy protection. For instance, reducing ε from 60 to 20 decreases the MIA success rate by 32.58% (from 87.80 to 59.19). However, this comes at the cost of reduced model utility, which declines by 16.63% (from 98.79 to 82.36). Without DP (i.e., no noise added), the model achieves the highest utility but faces the highest MIA risk. These results highlight the inherent trade-off in DP: higher ε improves utility but weakens privacy protection, while lower ε strengthens privacy but reduces utility. In this task, we set $\varepsilon = 40$ as a balanced compromise between privacy protection and model performance [37].

Table III compares DP-NIZK+CIV and standard DP in terms of RMSE, MIA-Resist (resistance to membership inference attacks), Verifiable (support for verifying correct DP application), and Secure-Trans (protection of model parameters transmission). While both methods achieve comparable RMSE, only DP-NIZK+CIV provides verifiable differential privacy and transmission integrity via non-interactive zero-knowledge proofs and authenticated encryption. Additionally, comparative experiments across client scales ($N \in$

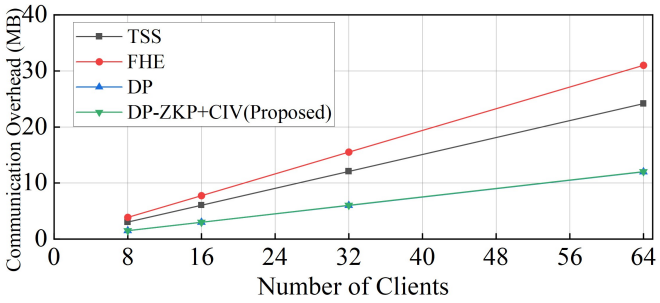

Fig. 4. Communication overhead comparison among TSS, FHE, DP, and the proposed DP-NIZK+CIV under different client scales.

$\{8, 16, 32, 64\}$) were conducted using FHE, TSS, and standard DP as baselines. Fig. 4 shows that the communication overhead of all methods increases as the number of clients grows. Nevertheless, the proposed DP-NIZK+CIV consistently achieves lower communication overhead, reducing costs by at least 61.17% and 50.13% compared to FHE and TSS, respectively, across varying client scales. These results demonstrate that DP-NIZK+CIV effectively combines enhanced privacy protection with improved communication efficiency.

TABLE II
IMPACT OF DIFFERENTIAL PRIVACY'S PRIVACY BUDGET ON MODEL UTILITY AND MIA SUCCESS RATE (MEAN (STANDARD DEVIATION))

(ϵ, δ)-DP	Utility	MIA-SR
No DP	100.00(0)	91.93(0.0181)
(60, 1×10^{-4})-DP	98.79(0.0317)	87.80(0.0258)
(40, 1×10^{-4})-DP	97.45(0.0471)	80.65(0.0333)
(30, 1×10^{-4})-DP	89.31(0.0324)	69.80(0.0313)
(20, 1×10^{-4})-DP	82.36(0.0299)	59.19(0.0548)

TABLE III
COMPARATIVE EVALUATION OF DP AND DP-NIZK+CIV ON ACCURACY AND PRIVACY PROTECTION (MEAN (STANDARD DEVIATION))

Method	RMSE	MIA-Resist	Verifiable	Secure-Trans
DP	0.0379(0.0049)	✓	✗	✗
DP-NIZK+CIV	0.0384(0.0066)	✓	✓	✓

D. Assessment of Aggregation Mechanism

To evaluate the robustness of the proposed DTAA method against various client-side anomalies, we use the sign-flipping attack [38] as a representative scenario, where 20% of each client's model parameters are reversed. DTAA is compared with FedAvg, MultiKrum, and T-Mean under different client scales ($N = \{8, 16, 32, 64\}$) and anomaly rates ($P = \{0, 0.1, 0.3\}$), where P denotes the proportion of anomalous clients among all clients.

As shown in Fig. 5, DTAA achieves competitive RMSE across most client scales N and anomaly rates P, demonstrating superior robustness. In contrast, FedAvg suffers significant degradation as P increases, due to its equal-weight averaging that indiscriminately aggregates abnormal clients. While T-Mean and MultiKrum show some resilience, they have clear limitations: T-Mean applies fixed-ratio trimming, leading to unnecessary discarding of benign model parameters, and MultiKrum requires predefined adversary bounds, both lacking the ability to dynamically adapt to varying anomaly proportions in a ZT environment. As a result, DTAA outperforms T-Mean and MultiKrum, achieving approximately 28.21% and 17.02% lower RMSE, respectively, under $N = 16$, $P = 0.5$.

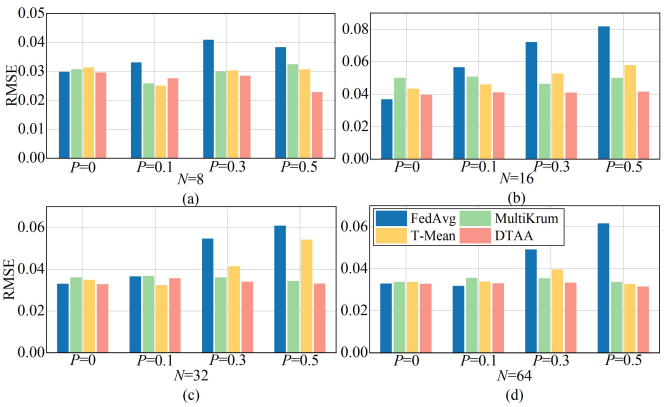

Fig. 5. Comparison of aggregation mechanisms under sign-flipping attacks: RMSE across varying anomaly rates (P) and client scales (N).

TABLE IV
ABLATION STUDY OF KEY COMPONENTS IN THE PROPOSED METHOD (MEAN (STANDARD DEVIATION))

Method	MAE	RMSE	MAAPE	ΔRMSE
Proposed(Baseline)	0.0098(0.0015)	0.0384(0.0049)	0.1380(0.0238)	-
No MA	0.0117(0.0010)	0.0424(0.0073)	0.1530(0.0124)	↑ 0.040
No Fed (Centralization)	0.0071(0.0015)	0.0283(0.0066)	0.0995(0.0162)	↓ 0.101
No DTAA (FedAvg)	0.0108(0.0014)	0.0403(0.0024)	0.1469(0.0095)	↑ 0.019
No DP-NIZK+CIV	0.0091(0.0009)	0.0326(0.0041)	0.1329(0.0144)	↓ 0.058

E. Ablation Experiment

To investigate the contribution of key components in the proposed method, we conduct ablation experiments by individually removing the multi-head attention mechanism (No MA), the federated learning setting (No Fed), the Dynamic Trust-Aware Aggregation (No DTAA), and the DP-NIZK+CIV privacy-preserving mechanism (No DP-NIZK+CIV). The results are summarized in Table IV.

Compared to the proposed method, removing the MA leads to the largest RMSE increase (0.040), highlighting its importance in modeling temporal dependencies. The No Fed setting achieves lower errors due to centralized training but sacrifices privacy. Omitting DP-NIZK+CIV slightly improves accuracy, but compromises privacy protection, communication integrity, and verifiability—critical requirements in ZT federated environments.

F. Comparative Experiments with Missing Rates

Keeping the hybrid missing pattern proportion constant, the missing rate of wind power data in the input samples is set to 20%, 50%, 70%, and 90%. By repeatedly testing the methods, the evaluation metrics (MAE, RMSE, MAAPE) on the test sets of all clients are summed and averaged, as shown in Table V.

The evaluation results show that the ZTFed-MAS2S achieves superior overall performance across all missing rates, generally yielding lower MAE, RMSE, and MAAPE

TABLE V
PERFORMANCE COMPARISON ACROSS DIFFERENT MISSING RATES UNDER 25% DISCRETE MISSING PROPORTION (MEAN (STANDARD DEVIATION))

Metrics	Missing rate (%)	Statistics		Machine Learning		Deep Learning			**ZTFed-MAS2S**
		Mean	EM	Missforest	kNN	GAIN	Bi-LSTM	Transformer	
MAE	20	0.2049(0.0011)	0.2538(0.0023)	0.0966(0.0021)	0.0879(0.0003)	0.0217(0.0005)	0.0088(0.0004)	0.0065(0.0005)	0.0083(0.0010)
	50	0.2251(0.0001)	0.2660(0.0001)	0.1492(0.0006)	0.1575(0.0013)	0.0360(0.0039)	0.0295(0.0122)	0.0077(0.0004)	0.0091(0.0014)
	70	0.2285(0.0005)	0.2663(0.0004)	0.1692(0.0011)	0.1860(0.0002)	0.0530(0.0047)	0.0462(0.0127)	0.0112(0.0033)	0.0107(0.0016)
	90	0.2309(0.0006)	0.2632(0.0003)	0.1905(0.0015)	0.2075(0.0006)	0.0751(0.0098)	0.0620(0.0133)	0.0167(0.0028)	0.0116(0.0023)
	Average	0.2223	0.2623	0.1514	0.1597	0.0464	0.0366	0.0105	0.0099
RMSE	20	0.2861(0.0022)	0.3771(0.0042)	0.1719(0.0030)	0.1765(0.0004)	0.0343(0.0004)	0.0411(0.0031)	0.0460(0.0014)	0.0360(0.0065)
	50	0.3176(0.0010)	0.3932(0.0007)	0.2435(0.0003)	0.2699(0.0008)	0.0496(0.0055)	0.0787(0.0303)	0.0486(0.0036)	0.0348(0.0065)
	70	0.3263(0.0018)	0.3976(0.0004)	0.2711(0.0024)	0.2980(0.0011)	0.0689(0.0055)	0.1127(0.0255)	0.0617(0.0118)	0.0408(0.0043)
	90	0.3367(0.0009)	0.4005(0.0012)	0.3046(0.0037)	0.3212(0.0010)	0.0955(0.0080)	0.1375(0.0282)	0.0712(0.0230)	0.0411(0.0045)
	Average	0.3167	0.3921	0.2478	0.2664	0.0620	0.0925	0.0569	0.0382
MAAPE	20	0.6677(0.0073)	0.7397(0.0050)	0.4746(0.0022)	0.4355(0.0034)	0.2821(0.0040)	0.1628(0.0163)	0.1308(0.0380)	0.1366(0.0199)
	50	0.6767(0.0002)	0.7393(0.0030)	0.5499(0.0029)	0.5427(0.0041)	0.4051(0.0055)	0.2463(0.0408)	0.1332(0.0358)	0.1347(0.0224)
	70	0.6776(0.0017)	0.7370(0.0003)	0.5817(0.0003)	0.5934(0.0016)	0.4813(0.0070)	0.3040(0.0431)	0.1501(0.0343)	0.1398(0.0253)
	90	0.6788(0.0040)	0.7303(0.0028)	0.6104(0.0004)	0.6339(0.0030)	0.5573(0.0344)	0.3568(0.0360)	0.1803(0.0382)	0.1463(0.0285)
	Average	0.6752	0.7366	0.5542	0.5514	0.4314	0.2675	0.1486	0.1394

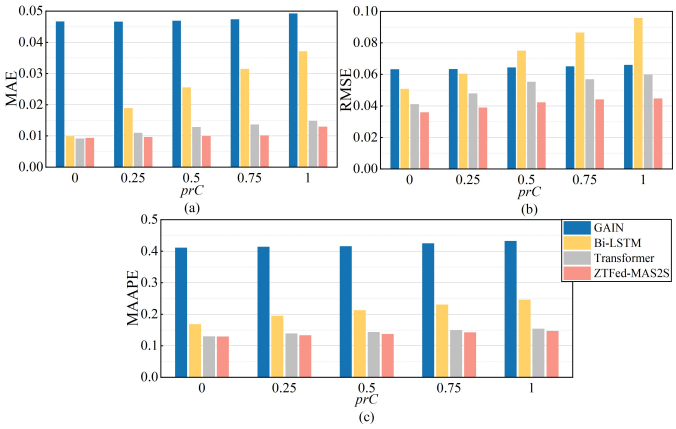

Fig. 6. Comparison results with different proportions in the hybrid missing patterns. prC represents the proportion of continuous missing in the hybrid missing pattern.

compared to other methods. Under extreme missing rates (90%), the ZTFed-MAS2S maintains superior performance with only marginal increases in the metrics, showcasing its exceptional resilience to extreme data sparsity. By leveraging its attention mechanism to effectively capture and integrate relevant information from available data points near missing values, our method surpasses the second-best method, with average improvements of 5.71%, 32.86%, and 6.19% in MAE, RMSE, and MAAPE, respectively. At a 90% missing rate, it outperforms the second-best method by 30.54%, 42.28%, and 18.86% in the same metrics, further demonstrating its superiority.

G. Comparative Experiments with Hybrid Missing Patterns

With the environment unchanged, the continuous missing proportion in the hybrid missing patterns decreases from 100% to 0%, while the discrete missing proportion rises from 0% to 100%, ensuring the missing rate remains constant (by taking the average value), as shown in Fig. 6. The results show that, among the compared deep learning methods:

1) As the continuous missing proportion increases, the performance of all methods declines. The reason for this phenomenon is that a higher proportion of continuous missing data reduces the availability of neighboring data points, making imputation more challenging.

2) Notably, the ZTFed-MAS2S outperforms the other alternatives by leveraging an encoder-decoder architecture and an attention mechanism to effectively capture long-term dependencies. This makes it particularly suitable for handling complex hybrid missing patterns.

TABLE VI
SENSITIVITY ANALYSIS OF DIFFERENT IMPUTATION METHODS

Sensitivities	GAIN	Bi-LSTM	Transformer	ZTFed-MAS2S
$S_{mr,rmse}$	0.0908	0.1401	0.0373	0.0088
$S_{prC,rmse}$	0.0107	0.0464	0.0186	0.0091

H. Sensitivity Analysis

According to (40) and (41), the results of the sensitivity analysis under various missing scenarios are presented in Table VI, where the proposed method is compared against other deep learning-based approaches. Among all the methods, the ZTFed-MAS2S demonstrates the lowest sensitivity values for both metrics, with $S_{mr,rmse} = 0.0088$ and $S_{prC,rmse} = 0.0091$. Specifically, regarding $S_{mr,rmse}$, our method demonstrates a sensitivity reduction of 76.41% compared to the second-best performing method, the Transformer. Similarly, for $S_{prC,rmse}$, our approach achieves a 14.95% lower sensitivity compared to the second-best method, the GAIN. These results fully demonstrate the superior robustness of the ZTFed-MAS2S in addressing complex missing scenarios.

V. CONCLUSION

ZTFed-MAS2S pioneers the integration of verifiable differential privacy and dynamic trust-aware aggregation under a zero-trust architecture for wind data imputation. Comprehensive experiments on the NREL datasets demonstrate the effectiveness and superiority of the proposed method. Specifically:

1) The DP-NIZK+CIV mechanism ensures verifiable privacy preservation and secure model parameters transmission, reducing communication overhead by at least 61.17%

and 50.13% compared to the FHE and TSS, respectively. The DTAA enhances robustness against anomalous clients, achieving up to 28.21% and 17.02% lower RMSE than the MultiKrum and T-Mean, respectively, under $N = 16$, $P = 0.5$.

2) The ZTFed-MAS2S consistently outperforms existing methods in both accuracy and robustness across various missing scenarios. Compared to the Transformer, it achieves 5.71%, 32.86%, and 6.19% higher accuracy in MAE, RMSE, and MAAPE, respectively, with improvements reaching 30.54%, 42.28%, and 18.86% under a 90% missing rate. Additionally, the method reduces sensitivity to missing rates by 76.41% compared to the Transformer and to hybrid missing patterns by 14.95% compared to the GAIN.

While ZTFed provides strong privacy guarantees, its privacy-utility trade-offs require co-optimized designs. Future work will develop adaptive strategies for enhanced robustness against non-independent and identically distributed (non-IID) data heterogeneity in IIoT systems.

REFERENCES

[1] P. Aaslid, M. Korpås, M. M. Belsnes, and O. B. Fosso, "Stochastic optimization of microgrid operation with renewable generation and energy storages," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 3, pp. 1481–1491, 2022.

[2] Z. Wang and S. Bu, "Probabilistic frequency stability analysis considering dynamics of wind power generation with different control strategies," *IEEE Transactions on Power Systems*, vol. 39, no. 5, pp. 6412–6425, 2024.

[3] W. Song, J. Yan, S. Han, N. Zang, S. Liu, C. Ge, and Y. Liu, "A self-supervised pre-learning method for low wind power forecasting," *IEEE Transactions on Sustainable Energy*, pp. 1–14, 2025.

[4] Q. Meng, S. Hussain, F. Luo, Z. Wang, and X. Jin, "An online reinforcement learning-based energy management strategy for microgrids with centralized control," *IEEE Transactions on Industry Applications*, vol. 61, no. 1, pp. 1501–1510, 2025.

[5] T. Wang, H. Ke, A. Jolfaei, S. Wen, M. S. Haghighi, and S. Huang, "Missing value filling based on the collaboration of cloud and edge in artificial intelligence of things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5394–5402, 2022.

[6] X. Chen, M. Lei, N. Saunier, and L. Sun, "Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 301–12 310, 2022.

[7] F. Mouret, A. Hippert-Ferrer, F. Pascal, and J.-Y. Tourneret, "A robust and flexible EM algorithm for mixtures of elliptical distributions with missing data," *IEEE Transactions on Signal Processing*, vol. 71, pp. 1669–1682, 2023.

[8] Y. Li, L. Yu, L. Xing, and F. Liu, "Analysis of influencing factors of lane change prediction with data missing," *IEEE Transactions on Intelligent Vehicles*, pp. 1–13, 2024.

[9] N. U. Okafor and D. T. Delaney, "Missing data imputation on IoT sensor networks: Implications for on-site sensor calibration," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 22 833–22 845, 2021.

[10] W. Liu, C. Ren, and Y. Xu, "Missing-data tolerant hybrid learning method for solar power forecasting," *IEEE Transactions on Sustainable Energy*, vol. 13, no. 3, pp. 1843–1852, 2022.

[11] X.-Y. Li, Y. Xu, Q.-X. Zhu, and Y.-L. He, "Industrial data imputation based on multiscale spatiotemporal information embedding with asymmetrical transformer," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2025.

[12] J. V. P R, N. Sam K, V. T, and A. C. Kathiresan, "Development and performance analysis of aquila algorithm optimized SPV power imputation and forecasting models," *IEEE Transactions on Sustainable Energy*, vol. 15, no. 3, pp. 2103–2114, 2024.

[13] L. Chen, Y. Xu, Q.-X. Zhu, and Y.-L. He, "Adaptive multi-head self-attention based supervised VAE for industrial soft sensing with missing data," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 3, pp. 3564–3575, 2024.

[14] H. J. Kim and M. K. Kim, "An unsupervised data-mining and generative-based multiple missing data imputation network for energy dataset," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 11, pp. 1–12, 2024.

[15] Q. Li, Y. Xu, B. S. H. Chew, H. Ding, and G. Zhao, "An integrated missing-data tolerant model for probabilistic PV power generation forecasting," *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4447–4459, 2022.

[16] Y. Li, R. Wang, Y. Li, M. Zhang, and C. Long, "Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach," *Applied Energy*, vol. 329, p. 120291, 2023.

[17] R. Fotohi, F. Shams Aliee, and B. Farahani, "A lightweight and secure deep learning model for privacy-preserving federated learning in intelligent enterprises," *IEEE Internet of Things Journal*, vol. 11, no. 19, pp. 31 988–31 998, 2024.

[18] Y. Li, X. Wei, Y. Li, Z. Dong, and M. Shahidehpour, "Detection of false data injection attacks in smart grid: A secure federated deep learning approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 6, pp. 4862–4872, 2022.

[19] R. Fotohi, F. S. Aliee, and B. Farahani, "Decentralized and robust privacy-preserving model using blockchain-enabled federated deep learning in intelligent enterprises," *Applied Soft Computing*, vol. 161, p. 111764, 2024.

[20] K. Wei, J. Li, M. Ding, C. Ma, Y.-S. Jeon, and H. V. Poor, "Covert model poisoning against federated learning: Algorithm design and optimization," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 3, pp. 1196–1209, 2024.

[21] T. Wang, Z. Zheng, and F. Lin, "Federated learning framework based on trimmed mean aggregation rules," *Expert Systems with Applications*, p. 126354, 2025.

[22] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[23] N. M. Hijazi, M. Aloqaily, M. Guizani, B. Ouni, and F. Karray, "Secure federated learning with fully homomorphic encryption for IoT communications," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4289–4300, 2023.

[24] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2020.

[25] V. Stafford, "Zero trust architecture," *NIST special publication*, vol. 800, no. 207, pp. 800–207, 2020.

[26] D. Bahdanau, "Neural machine translation by jointly learning to align and translate," *in Proc. 3rd Int. Conf. Learn. Representa*, 2015.

[27] O. Rubasinghe, X. Zhang, T. K. Chau, Y. H. Chow, T. Fernando, and H. H.-C. Iu, "A novel sequence to sequence data modelling based CNN-LSTM algorithm for three years ahead monthly peak load forecasting," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1932–1947, 2024.

[28] Z. Xing, Z. Zhang, Z. Zhang, Z. Li, M. Li, J. Liu, Z. Zhang, Y. Zhao, Q. Sun, L. Zhu, and G. Russello, "Zero-knowledge proof-based verifiable decentralized machine learning in communication network: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025.

[29] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[30] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa, "The wind integration national dataset (wind) toolkit," *Applied Energy*, vol. 151, pp. 355–366, 2015.

[31] S. Yang, M. Dong, Y. Wang, and C. Xu, "Adversarial recurrent time series imputation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 1639–1650, 2023.

[32] C. Wei, W. Li, C. Gong, and W. Chen, "DC-SGD: Differentially private SGD with dynamic clipping through gradient norm distribution estimation," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 4498–4511, 2025.

[33] P. Malhan and M. Mittal, "A novel ensemble model for long-term forecasting of wind and hydro power generation," *Energy Conversion and Management*, vol. 251, p. 114983, 2022.

[34] J. Ma, J. C. Cheng, F. Jiang, W. Chen, M. Wang, and C. Zhai, "A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data," *Energy and Buildings*, vol. 216, p. 109941, 2020.

[35] B. Jiang, J. Li, G. Yue, and H. Song, "Differential privacy for industrial internet of things: Opportunities, applications, and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 430–10 451, 2021.

[36] L. Song, R. Shokri, and P. Mittal, "Membership inference attacks against adversarially robust deep learning models," in *2019 IEEE Security and Privacy Workshops (SPW)*, 2019, pp. 50–56.

[37] J. P. Near, D. Darais, N. Lefkovitz, G. Howarth *et al.*, "Guidelines for evaluating differential privacy guarantees," *National Institute of Standards and Technology, Tech. Rep*, pp. 800–226, 2023.

[38] L. Yang, Y. Miao, Z. Liu, Z. Liu, X. Li, D. Kuang, H. Li, and R. H. Deng, "Enhanced model poisoning attack and multi-strategy defense in federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 3877–3892, 2025.