MAD-STF: A Dual-Branch Framework for Multivariate Time Series Forecasting under Complex Missingness

Tianhao Zhang*
Department of Automation
University of Science and Technology of China
Hefei, Anhui, China
Zthsa21010026@mail.ustc.edu.cn

Xiaoran Jin
Department of Automation
University of Science and Technology of China
Hefei, Anhui, China
jinxr@mail.ustc.edu.cn

Qijiu Xia
Department of Automation
University of Science and Technology of China
Hefei, Anhui, China
1059350098@qq.com

Yunbo Zhao
Department of Automation
University of Science and Technology of China
Hefei, Anhui, China
ybzhao@ustc.edu.cn

Abstract

In recent years, multivariate time series forecasting has found widespread applications in many fields. However, during data collection, sensor aging or malfunctions often lead to complex missingness patterns—including both partial and complete (entire-column) missing data. Existing methods for handling missing values either introduce imputation errors that degrade prediction performance or significantly increase model complexity without sufficient interpretability. To address these challenges, we propose the Missingness-Aware Dual-branch Structure for Time Series Forecasting (MAD-STF) model, which employs two dedicated components: the Target Autoregressive Structure (TAS) and the Exogenous Joint Modelling Structure (EJS). TAS processes the target variable sequence using interpolation-based imputation, while EJS handles covariates through an uncertainty-aware implicit imputation mechanism based on Monte Carlo Dropout, tailored to their distinct missing characteristics. Extensive experiments on public datasets with realistic missing patterns demonstrate that our approach achieves superior forecasting performance under complex missing-data scenarios.

CCS Concepts

• **Computing methodologies** → **Batch learning**; **Neural networks**.

Keywords

time series forecasting, complex missingness, dual-branch architecture, Monte Carlo dropout, singular spectrum analysis

ACM Reference Format:
Tianhao Zhang, Xiaoran Jin, Qijiu Xia, and Yunbo Zhao. 2026. MAD-STF: A Dual-Branch Framework for Multivariate Time Series Forecasting under Complex Missingness. In *2026 2nd International Conference on Artificial Intelligence and Computational Intelligence (AICI 2026), February 06–08, 2026, Zhengzhou, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3810280.3810310

*Corresponding author

ACM ISBN 979-8-4007-2279-0/26/02
https://doi.org/10.1145/3810280.3810310

1 Introduction

Multivariate Time Series Forecasting (MSTF) currently serves as a core task in time series prediction, primarily leveraging data from all collected variables to forecast one or multiple target variables at future time steps. This task has broad applications across various domains, such as weather forecasting [16, 18], stock market prediction [15], and traffic flow prediction [9]. In recent years, deep learning-based neural network models have significantly improved both the forecasting horizon and accuracy, evolving from RNN-based architectures (e.g., GRU [29], LSTM [1]) to self-attention-based models (e.g., Transformer [25], iTransformer [12]). However, in real-world scenarios, data collection devices like sensors may suffer from ageing or malfunction, leading to various types of missing data [3, 14, 22]. Such missingness compromises the integrity of the original data, making it considerably more difficult for models to capture temporal trends [5]. Consequently, the analysis of historical information and the extraction of meaningful features are severely disrupted under missing-data conditions.

Currently, there are primarily two methods for addressing missing data issues: (1) Imputation through preprocessing, which fills in missing variables [17]. The filled-in data is then fed into the model for processing. This method performs well when the missing rate is low, but for situations with high missing rates or continuous missing data, it often introduces artificial imputation errors, leading to performance biases in prediction results [21]. (2) Another method, such as the Ginar model [27], employs an end-to-end structure, where the model internally provides analysis results for missing values, which are then used for subsequent feature extraction and other processes. However, this method lacks interpretability, making it difficult to determine whether the model has effectively handled missing values. For situations with fewer missing values, the end-to-end approach can significantly increase model complexity, which is counterproductive for target prediction.

/Therefore, in response to the complex missing data scenarios present in real datasets, we have designed a MAD-STF model to predict target variables based on input multivariable sequences.

This model primarily consists of two structures: TAS and EJS. The TAS structure is solely responsible for analyzing the historical data of the target variable. After imputing the partially missing data using interpolation methods, it employs singular spectrum decomposition to decompose the target sequence into trend and periodic components, and designs different modules for feature extraction and subsequent integration. The EJS structure first screens covariates using Pearson correlation coefficients. For sequences with strong correlations, it designs an end-to-end structure to analyze the missing parts and utilizes the Monte Carlo Dropout method to provide uncertainty estimates for the imputations. Subsequently, it combines the imputed information and corresponding uncertainties and predicts the target using historical data of the target variable and covariates. Finally, the prediction results from TAS and EJS are integrated to obtain the future predictions of the target variable.

2 Related work

For the task of multivariate time series prediction with missing data, the main research approach in the early stage was to adopt a step-by-step strategy. Specifically, a missing value imputation strategy (such as the k-nearest neighbor method) was first designed to fill in the missing distribution. Then, using the imputed data, the problem was simplified into a common multivariate time series prediction problem for model design. For example, the Joint-IF model [10] adopts a repair-before-prediction strategy, first inferring missing values using low-rank matrix factorization and then designing a model for subsequent analysis. And the BRITS model [4] directly learns missing values within a bidirectional recurrent dynamical system and employs task-specific decoding layers for different downstream tasks.

Afterwards, to avoid artificially introducing filling errors in the two-stage method and further enhance the model's prediction performance in the missing state, end-to-end implicit filling strategies emerged. For example, STGNN-based models such as ASTGCN [30] and AGCRN [2] all model dynamic graph structures among variables to capture spatiotemporal features, and GRU-D further builds upon them by leveraging dynamic graph structures to implicitly model long-sequence missing data. However, due to design issues, these models often lack observation methods for implicit filling results, thus lacking interpretability and controllability.

Meanwhile, uncertainty modeling for models has gradually become a widely used high-quality strategy to enhance model interpretability in recent years [11, 24]. However, most existing uncertainty modeling is primarily used as a visualization tool to display model results or module effects, rather than incorporating uncertainty as additional information to feed back to the model, thereby improving its performance.

In response to the aforementioned problems and challenges, the proposed model MAD-STF in this paper makes two improvements: (1) For complex missing scenarios involving target variables and covariates, we adopt two structures to handle the target variables and covariates separately, employing two different strategies based on their missing characteristics: distribution and end-to-end. (2) In the EJS structure for the end-to-end model structure, we use the Monte Carlo Dropout method to provide uncertainty, and simultaneously, treat uncertainty as a weight that affects the imputed values, thereby improving the model's prediction performance. Experimental verification shows that this design significantly outperforms existing methods in complex missing scenarios.

3 Preliminary

In the multivariate time series forecasting task with missing data, consider an input sequence $X \in \mathbb{R}^{H \times N}$, where H represents the length of the historical time window and N represents the total number of variables. Each row $X_t = (x_t^1, x_t^2, \ldots, x_t^N) \in \mathbb{R}^N$ represents the multivariate observations at time t. We designate one of the variables x^g ($g \in \{1, 2, \ldots, N\}$) as the target variable, denoted as $Y = (x_1^g, x_2^g, \ldots, x_H^g)^\top$, with the remaining $N-1$ variables serving as auxiliary covariates.

Next, consider two scenarios of missing data:

- **Partial missingness of the target variable**: At certain time steps t, there is a loss of the target variable x_t^g.
- **Structural missingness of covariates**: Variables other than the target variable may not only exhibit random missingness (i.e., partial missingness), but may also be completely missing throughout the entire time window (i.e., complete column missingness)

Therefore, the objective of this paper is to design a predictive model $f(\cdot)$ that can effectively utilize the historical observation sequence X containing the aforementioned complex missing patterns, while predicting the target variables for the next L time steps, as expressed in Equation (1)

$$\hat{Y}_{\text{future}} = f\big(X_{1:H}, \text{horizon} = L\big), \tag{1}$$

where $\hat{Y}_{\text{future}} = (\hat{y}_{H+1}, \hat{y}_{H+2}, \ldots, \hat{y}_{H+L})^\top \in \mathbb{R}^L$ epresents the predicted values of the target variables for the next L steps.

4 Method

The MAD-STF model draws inspiration from the DA-SPS model [28], and designs two parallel substructures: the Target Autoregressive Structure (TAS) and the Exogenous Joint Modelling Structure (EJS). Model Figure as shown in Figure 1

In TAS, to address the issue of partial missing target variables, we first employ linear interpolation to fill in the missing values. Subsequently, we utilize the Singular Spectrum Analysis (SSA) strategy [8] to decompose them into trend and periodic components. We then employ the P-ConvLSTM architecture and standard LSTM to extract temporal dynamic features, respectively. Finally, we integrate the predicted future trend and periodic components to obtain the future prediction results based on the univariate historical sequence.

In EJS, inspired by the IBN model [13], we designed an implicit missingness-aware mechanism: dynamically modeling the dependencies between variables through a graph attention structure, and incorporating Monte Carlo Dropout [6] for implicit imputation of uncertainty awareness for missing covariates. On this basis, an improved L-Attn decoder is introduced to integrate the spatiotemporal contextual information of the target variable and covariates, generating predictions based on the spatiotemporal correlation between the target variable and covariates.

Finally, MAD-STF performs weighted fusion on the outputs of TAS and EJS to obtain the final prediction result. Next, we will introduce the details of this model.

4.1 Target Autoregressive Structure

The TAS structure is primarily responsible for extracting temporal dynamic features from the historical sequence of the target variable x^g itself. Given the partially missing target variable sequence $X^g = (x_1^g, ..., x_H^g)^T$, considering the characteristics of strong local continuity and smooth variation often present in continuous data, although it exhibits trend and periodic changes under long-term observation, it shows approximate linear variation in shorter time windows. Therefore, we choose to use linear interpolation to fill in the missing points of the target variable.

Define the missing value of the input variable at time t as $x_t^g = NaN$. Let $O = \{t \in \{1, .., H\} | x_t^g \ is\ observed\}$ and $M = \{t \in \{1, .., H\} | x_t^g \ is\ NaN\}$ represent the observed value set and the missing value set, respectively. For missing values at non-initial or non-terminal positions $t \in M$, their imputed values $\hat{x}_t^g$ can be calculated using equation (2):

$$\hat{x}_t^g = x_{t_{left}}^g + \frac{t - t_{left}}{t_{right} - t_{left}} \left(x_{t_{right}}^g - x_{t_{left}}^g \right) \tag{2}$$

where $t_{left} = max\{s \in O | s < t\}$ is the nearest observation time to the left of t, and $t_{right} = min\{s \in O | s > t\}$ is the nearest observation time to the right of t. If the missing X^g sequence occurs at the beginning or end, forward filling or backward filling strategies are used for boundary processing. Specifically, the first $x_{t_{min}}^g$ and the last $x_{t_{max}}^g$ are used for filling, where $t_{min} = min(O)$ is the first valid observation value, and $t_{max} = max(O)$ is the last valid observation value.

After TAS fills in the input target variable sequence X^g using the interpolation filling method to obtain $\hat{X}^g$, we will extract the sequence features of the target variable from the filled sequence. The core idea is to decompose the target variable into periodic and trend terms, and adopt different model structures according to the different characteristics of different sequences.

We first apply Singular Spectrum Analysis (SSA) to decompose the interpolated target sequence $\hat{X}^g = (x_1^g, ..., x_H^g)$ (where, for the value x_t^g at time t, when $t \in M$, the value is an interpolated estimate; otherwise, it is the original observed value x_t^g). This method first constructs a trajectory matrix with an embedding window length of m, where m satisfies $1 < m < H$. The construction method is shown in Equation (3):

$$\hat{X}^g = \begin{bmatrix} \hat{x}_1^g & \hat{x}_2^g & \cdots & \hat{x}_{H-m+1}^g \\ \hat{x}_2^g & \hat{x}_3^g & \cdots & \hat{x}_{H-m+2}^g \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_m^g & \hat{x}_{m+1}^g & \cdots & \hat{x}_H^g \end{bmatrix} \in \mathbb{R}^{m \times (H-m+1)}, \tag{3}$$

Next, perform singular spectrum decomposition on the trajectory matrix, with the specific method as follows:

$$X = \sum_{i=1}^{d} \lambda_i \, \mathbf{u}_i \mathbf{v}_i^\top, \tag{4}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d > 0$, and $\mathbf{u}_i \in \mathbb{R}^m$ and $\mathbf{v}_i \in \mathbb{R}^{H-m+1}$ are the corresponding left and right singular vectors, respectively.

Next, based on the eigenvalue decay characteristics and the contribution rate of principal components in the TAS structure, we divide $1, \ldots, d$ into three groups: I_{tr} represents the trend eigenvalues, which signify the slowly changing trend item sequence in the target sequence; I_{cy} denotes the cycle sequence, indicating the periodic fluctuation item sequence present in the target sequence; and I_{no} stands for the noise eigenvalues, representing the disturbance item sequence in the target sequence. Subsequently, using the diagonal reconstruction method, we can extract the trend item sequence $\hat{X_{tr}}^g$ and the periodic item sequence $\hat{X_{cy}}^g$ while removing noise interference. Leveraging the distinct temporal characteristics of these sequences, we employ different modules for sequence analysis and information extraction. And this process is shown in Figure 2

For the trend term sequence $\hat{X_{tr}}^g$, its variation is gradual but spans the entire time window. Directly using FC-LSTM may lead to vanishing gradients due to the excessively long sequence. To address this, we adopt the P-CONV-LSTM structure. First, we perform patch division on the target trend term sequence, splitting it into $n = \lfloor H/l \rfloor$ non-overlapping subsequences of length l:

$$\mathbf{p}_e = \left[y_{(e-1)l+1}^{\text{tr}}, \, y_{(e-1)l+2}^{\text{tr}}, \, \ldots, \, y_{el}^{\text{tr}} \right]^\top \in \mathbb{R}^l, \quad e = 1, \ldots, n. \tag{5}$$

Next, we will input P_e as the input at a certain moment into each CONV-LSTM unit, and the internal state of the unit is updated as follows:

$$\begin{aligned} \mathbf{i}_e &= \sigma\big(\mathbf{W}_i * \mathbf{p}_e + \mathbf{U}_i * \mathbf{h}_{e-1} + \mathbf{V}_i \odot \mathbf{c}_{e-1} + \mathbf{b}_i\big), \\ \mathbf{f}_e &= \sigma\big(\mathbf{W}_f * \mathbf{p}_e + \mathbf{U}_f * \mathbf{h}_{e-1} + \mathbf{V}_f \odot \mathbf{c}_{e-1} + \mathbf{b}_f\big), \\ \tilde{\mathbf{c}}_e &= \tanh\big(\mathbf{W}_c * \mathbf{p}_e + \mathbf{U}_c * \mathbf{h}_{e-1} + \mathbf{b}_c\big), \\ \mathbf{c}_e &= \mathbf{f}_e \odot \mathbf{c}_{e-1} + \mathbf{i}_e \odot \tilde{\mathbf{c}}_e, \\ \mathbf{o}_e &= \sigma\big(\mathbf{W}_o * \mathbf{p}_e + \mathbf{U}_o * \mathbf{h}_{e-1} + \mathbf{V}_o \odot \mathbf{c}_e + \mathbf{b}_o\big), \\ \mathbf{h}_e &= \mathbf{o}_e \odot \tanh\big(\mathbf{c}_e\big). \end{aligned} \tag{6}$$

where $*$ represents one-dimensional convolution, $\odot$ denotes element-wise multiplication, and $\sigma(.)$ is the activation function. In the unit structure, the P_e input into each unit of P-CONV-LSTM captures the local patterns within each patch through the internal convolution kernel, and the gating mechanism between units models the long-term dependencies between each patch. Finally, we take the hidden state h_n output from the nth unit as the trend representation, and map it to the predicted value of the target variable trend for the next L steps through a fully connected layer, $\hat{Y}_{tr} \in \mathbb{R}^L$.

For the cycle sequence part $\hat{X}_{cy}^g$, due to its stable periodic variation, a standard LSTM [7] is adopted to model this sequence while considering both prediction accuracy and computational complexity reduction:

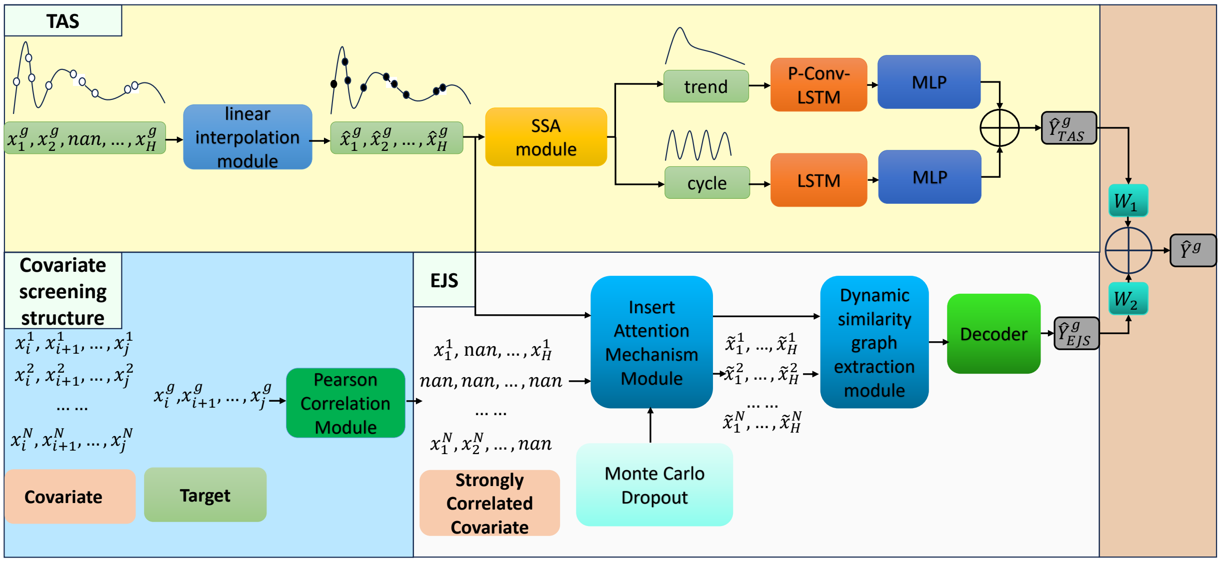

Figure 1: MAD-STF model structure figure

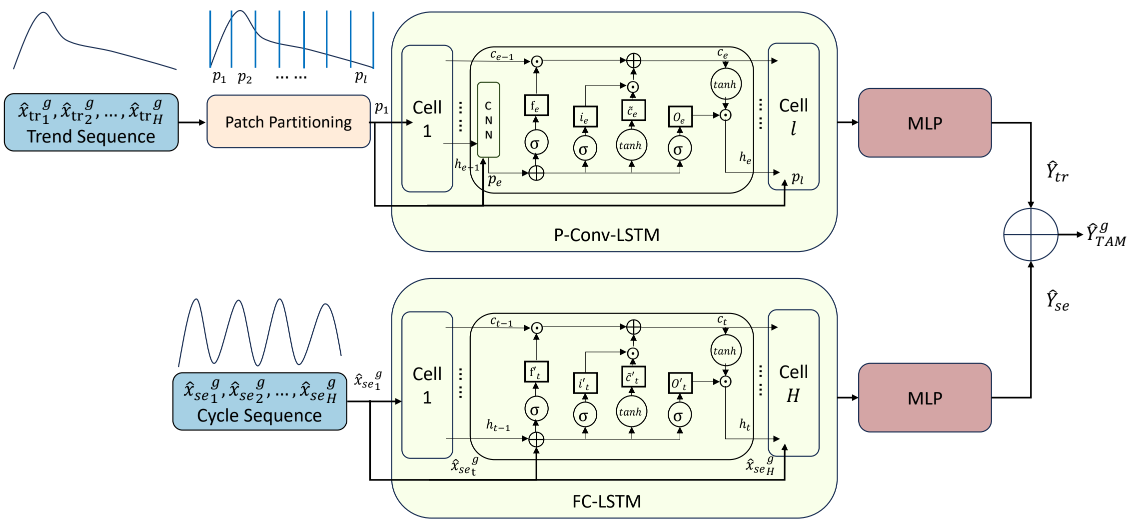

Figure 2: TAS module structure and its processing flow

$$
\begin{aligned}
\mathbf{f}'_t &= \sigma\left(\mathbf{W}'_f\left[\mathbf{h}'_{t-1};\ \hat{\mathbf{x}}^g_{\mathrm{cy},t}\right]+\mathbf{b}'_f\right),\\
\mathbf{i}'_t &= \sigma\left(\mathbf{W}'_i\left[\mathbf{h}'_{t-1};\ \hat{\mathbf{x}}^g_{\mathrm{cy},t}\right]+\mathbf{b}'_i\right),\\
\tilde{\mathbf{c}}'_t &= \tanh\left(\mathbf{W}'_c\left[\mathbf{h}'_{t-1};\ \hat{\mathbf{x}}^g_{\mathrm{cy},t}\right]+\mathbf{b}'_c\right),\\
\mathbf{c}'_t &= \mathbf{f}'_t \odot \mathbf{c}'_{t-1} + \mathbf{i}'_t \odot \tilde{\mathbf{c}}'_t,\\
\mathbf{o}'_t &= \sigma\left(\mathbf{W}'_o\left[\mathbf{h}'_{t-1};\ \hat{\mathbf{x}}^g_{\mathrm{cy},t}\right]+\mathbf{b}'_o\right),\\
\mathbf{h}'_t &= \mathbf{o}'_t \odot \tanh\left(\mathbf{c}'_t\right).
\end{aligned}
\tag{7}
$$

Afterwards, the hidden states of the last L units of the LSTM are projected through a fully connected layer to obtain the cycle term predictions $\hat{Y}_{cy} \in \mathbb{R}^L$ for the next L steps. The final prediction output of the TAS for the target variable is the linear superposition of the two predictions:

$$
\hat{Y}^g_{TAS} = \hat{Y}_{tr} + \hat{Y}_{cy}. \tag{8}
$$

4.2 Exogenous Joint Modelling Structure

In real-world scenarios, covariates in multivariate time series often present two types of issues: Firstly, some covariates may only have weak or even no correlation with the target variable, and introducing these variables during analysis can actually introduce noise. Secondly, during the collection of sequence information, covariate data may be partially missing, or even continuously missing for extended periods. Therefore, directly using the original input will not only reduce model efficiency but may also lead to prediction bias due to the propagation of erroneous information. Hence, we first perform a correlation-driven pre-screening of covariates.

In EJS, we employ the Pearson correlation coefficient [20] to compute the correlation between the other $N-1$ variables and the target variable X^g. The calculation method is illustrated in (9):

$$\rho_i = \frac{\mathrm{Cov}(X^i, X^g)}{\sigma_{X^i}\,\sigma_{X^g}}, \quad i \in \{1, \ldots, N\} \setminus \{g\}. \tag{9}$$

where $Cov(.)$ represents the covariance formula, and $\sigma(.)$ denotes the standard deviation formula. Using the aforementioned formulas, we can calculate the degree of correlation between each covariate and X^g. Simultaneously, we define τ as the correlation threshold. If $|\rho_i| < \tau$, it is defined as a weakly correlated variable and is eliminated, retaining only the highly correlated subset to form a reduced input $\tilde{X} \in \mathbb{R}^{H\times\tilde{N}} \quad (\tilde{N} \leq N)$. Through this preprocessing method, we can significantly reduce the dimensionality of the input data while ensuring that subsequent modeling focuses on covariates with strong correlation to the target variable.

Next, considering the partial and complete missing values in the remaining highly correlated covariates, traditional methods (such as mean imputation and forward filling) can introduce significant artificial biases due to the complexity of the missing data and the abundance of missing values in complete column missingness. Moreover, they fail to reflect the credibility of the imputation. Therefore, inspired by the IBN model, we propose a new end-to-end implicit imputation method, DUIN(Dual-graph Uncertainty-aware Implicit Network): instead of explicitly generating imputed values, it synchronizes information recovery through a learnable spatiotemporal dependency structure during the feature encoding process, and simultaneously employs Monte Carlo Dropout to quantify the uncertainty of the imputation.

Specifically, first, the missing values NaN in the input sequence $\tilde{X} \in \mathbb{R}^{H\times\tilde{N}}$ are temporarily replaced with 0, and then the input at each time step is fed into each unit of DUIN. Each unit comprises two key modules:

(1) **Insert Attention Mechanism Module**, The Insert Attention Mechanism Module can be shown in Figure 3. Used to learn the global dependencies between variables. At the t-th time step, we first map the input $\tilde{X}_t$ to an embedding representation $h \in \mathbb{R}^{\tilde{N}\times d}$ through a learnable projection matrix W, that is:

$$\mathbf{h} = \tilde{\mathbf{X}}_t\mathbf{W} \in \mathbb{R}^{\tilde{N}\times d}. \tag{10}$$

Subsequently, we obtain the attention score by concatenating the features of any two variables $[\mathbf{h}^i||\mathbf{h}^j]$ and scoring them with a learnable weight vector $\mathbf{a} \in \mathbb{R}^{2d}$:

$$e_{ij} = \mathbf{a}^\top[\mathbf{h}_i\|\mathbf{h}_j]. \tag{11}$$

The attention scores are normalized using LeakyReLU and Softmax, and after adding self-loops, a learnable adaptive adjacency matrix $A_{ia} \in \mathbb{R}^{\tilde{N}\times\tilde{N}}$ is finally obtained. Ultimately, neighbor information is aggregated to generate context-enhanced representations:

$$\mathbf{h}^{\mathrm{att}} = \mathbf{A}_{\mathrm{ia}}\mathbf{h} \in \mathbb{R}^{\tilde{N}\times d}. \tag{12}$$

Subsequently, we introduce the Monte Carlo Dropout method for uncertainty calibration, which consists of a multi-layer perceptron with Dropout structure. By filling the same missing position multiple times, we obtain a set of fillings $\{m^{(s)} \in \mathbb{R}^{\tilde{N}\times d}\}_{s=1}^{S}$. Here, S represents the number of fillings, that is:

$$m^{(s)} = \mathbf{MC}^{\mathrm{s}}(\mathbf{h}^{\mathrm{att}}), \quad s = 1, \ldots, S, \tag{13}$$

Here, $\mathbf{MC}^{\mathrm{s}}(.)$ represents different sub-networks activated by Dropout during the forward pass. Then, the mean and standard deviation of the set $\{m^{(s)}\}$ are calculated:

$$\begin{aligned} \mu &= \frac{1}{S}\sum_{s=1}^{S} m^{(s)}, \\ \sigma &= \sqrt{\frac{1}{S}\sum_{s=1}^{S}\left(m^{(s)} - \mu\right)^2}. \end{aligned} \tag{14}$$

Finally, output the uncertainty-weighted representation:

$$\tilde{x}_t' = \frac{\mu}{1+\sigma} \in \mathbb{R}^{\tilde{N}\times d}. \tag{15}$$

Through this module, EJS not only acquires the ability to fill missing covariates but also utilizes the Monte Carlo Dropout method to provide confidence for each imputed missing value. When there are significant differences between multiple imputation results (as indicated by a high σ value), the corresponding imputation results will be suppressed; conversely, signals with high confidence will be retained. At the same time, the uncertainty estimate σ generated by inserting an attention structure can also serve as an external indicator of data reliability. If σ for a certain covariate continues to increase, it may indicate sensor performance degradation, thereby triggering a maintenance alert. Similarly, in safety-critical applications such as traffic control or power grid management, when the overall uncertainty of the prediction results is high (i.e., confidence is low), the system can activate a backup model or request manual intervention for verification, thereby enhancing the robustness of the system.

(2) **Dynamic similarity graph extraction module**, Considering the potential short-term correlations between variables, such as synchronous drastic fluctuations and rapid rises, we employ a dynamic similarity graph extraction module to construct the adjacency structure actually used for information transmission. Based on $\tilde{x}_t'$, we first calculate the Euclidean distance between nodes and map it to similarity using a Gaussian kernel. The method is illustrated in Equation (16):

$$\tilde{A}_{ij} = \exp\left(-\frac{\left\|\tilde{\mathbf{x}}_{t,i}' - \hat{\tilde{x}}_{t,j}'\right\|_2^2}{2d}\right), \tag{16}$$

Afterwards, normalization is performed and a unit matrix is added to preserve self-loops, resulting in the dynamic adjacency matrix:

$$\mathbf{A}_{\text{dyn}} = \text{Softmax}(\tilde{\mathbf{A}} + \mathbf{I}) \in \mathbb{R}^{\tilde{N} \times \tilde{N}}. \tag{17}$$

Finally, multiply $\mathbf{A}_{\text{dyn}}$ by the original input at time t $\tilde{\mathbf{X}}_t$ to successfully integrate the temporal and spatial information of the input covariates and target variables.

By integrating the attention mechanism module and the dynamic similarity graph extraction module, the EJS structure establishes a complementary dual attention mechanism: the former models long-term stable dependencies, while the latter responds to short-term changes. Additionally, by introducing the concept of uncertainty, a fully learnable, end-to-end, and highly interpretable dual-graph fusion strategy is established.

In the decoder of the EJS, we refer to the L-Attn decoding module in the DA-SPS model, focusing only on the part of the encoder output that is related to the target variable. Assuming the hidden state at time t is $h_t \in \mathbb{R}^{d \times \tilde{H}}$, we extract the column corresponding to the target variable to form a new sequence $\mathbf{e}_t = \mathbf{h}_t^{(g)} \in \mathbb{R}^d$. These extracted target variables are concatenated into a sequence $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_H]^\top \in \mathbb{R}^{H \times d}$ for subsequent prediction. This strategy avoids introducing interference from irrelevant covariates during the decoding stage.

Subsequently, the improved L-Attn decoder is employed to generate predictions for the next L steps:

$$\begin{aligned}
\mathbf{q}_k &= \tanh(\mathbf{W}_q \mathbf{h}_{k-1}^{\text{dec}}), \\
\alpha_{k,t} &= \frac{\exp(\mathbf{q}_k^\top \mathbf{e}_t)}{\sum_{s=1}^{H} \exp(\mathbf{q}_k^\top \mathbf{e}_s)}, \\
\text{context}_k &= \sum_{t=1}^{H} \alpha_{k,t}\, \mathbf{e}_t, \\
\mathbf{h}_k^{\text{dec}}, \mathbf{c}_k^{\text{dec}} &= \text{LSTMCell}(\text{context}_k,\, \mathbf{h}_{k-1}^{\text{dec}},\, \mathbf{c}_{k-1}^{\text{dec}}), \\
\hat{y}_{H+k} &= \mathbf{w}^\top [\text{context}_k \,\|\, \mathbf{h}_k^{\text{dec}}] + b.
\end{aligned} \tag{18}$$

Finally, the EJS output is:

$$\hat{\mathbf{Y}}_{EJS}^{g} = (\hat{y}_{H+1}, \ldots, \hat{y}_{H+L})^\top \in \mathbb{R}^L \tag{19}$$

Following this, the prediction results obtained from EJS and TAS are integrated using two weight parameters, w_1 and w_2, to get the final predicted output $\hat{Y}_{\text{future}}$.

5 Experiments

In the experimental section, to further verify the effectiveness of our model, we utilize three public datasets (SML2010, PEMS04, Electricity) for validation. The relevant descriptions of the three public datasets are as follows:

(1) **SML2010 [19]** is a publicly available multivariate dataset collected from environmental and energy consumption sensors at the Spanish Intelligent Buildings Laboratory from March to April 2010. The dataset records a total of 24 variables, including temperature, humidity, sensor concentration, and light intensity, at 1-minute intervals.

(2) **Electricity [23]** is a dataset extracted from the hourly electricity consumption records of 321 large power companies in Australia from 2012 to 2014

(3) **PEMS04 [26]** is collected by the California Department of Transportation and updated every 5 minutes. It records the real-time traffic status of 307 sensor nodes in the San Francisco Bay Area freeway network for 59 days in 2018.

For specific information about each public dataset, please refer to the table 2:

Afterwards, the aforementioned dataset will be divided into training set, validation set, and test set according to the proportions of 60%, 20%, and 20%, respectively.

Next, to better evaluate the performance of the model, we choose MAE and RMSE as evaluation metrics, with the specific formulas as follows:

$$\begin{aligned}
\text{MAE} &= \frac{1}{H}\sum_{i=1}^{H} |y_i - \hat{y}_i| \\
\text{RMSE} &= \sqrt{\frac{1}{H}\sum_{i=1}^{H} (y_i - \hat{y}_i)^2}
\end{aligned} \tag{20}$$

Based on the aforementioned dataset and evaluation metrics, we tested the MAD-STF model under conditions where 10% of the covariates were missing entirely, and the target variable and other covariates were missing at 10%, 20%, and 50% sequentially. We utilized a historical input length of $history\_len = 48$ to predict the future target variable with a horizon of $horizon = 12$. Additionally, to verify the model's generalization ability, we designed unified hyperparameters for the MAD-STF model, with a correlation threshold of $\tau = 0.5$, a Monte Carlo Dropout filling count of $S = 10$, and a unified hidden layer dimension for both TAS and EJS set at $hidden\_size = 64$. We set the training epochs to $epoch = 100$, the batch size to $batch = 32$, the initial learning rate to $lr = 0.001$, and the learning rate decay to $lr\_decay = 0.9$. Afterwards, we selected the following models as the comparative models for the MAD-STF model:

(1) **DA-SPS [28]**: It utilizes two modules, TVPS and EVPS, to extract information from target variables and covariates, ultimately predicting the target variables.

(2) **IBN [13]**: Utilizes UAI, GGCN, and bidirectional LSTM to enhance the prediction of input sequences in the presence of entire column missingness

(3) **LSTM**: A classic model that utilizes long short-term memory modules to predict target variables

The final predicted estimation results are presented in the table below:

According to Table 1, MAD-STF maintains excellent predictive performance in large-scale scenarios involving tens or even hundreds of exogenous variables, such as the Electricity and PEMS04 datasets. Even in the presence of complete column missingness in these datasets (30 exogenous variables in Electricity and 80 in PEMS04 are completely missing), the model's MAE and RMSE metrics remain stable under different missing ratios, without significant attenuation or degradation. This indicates that, despite the theoretical increase in prediction variance that may arise from applying

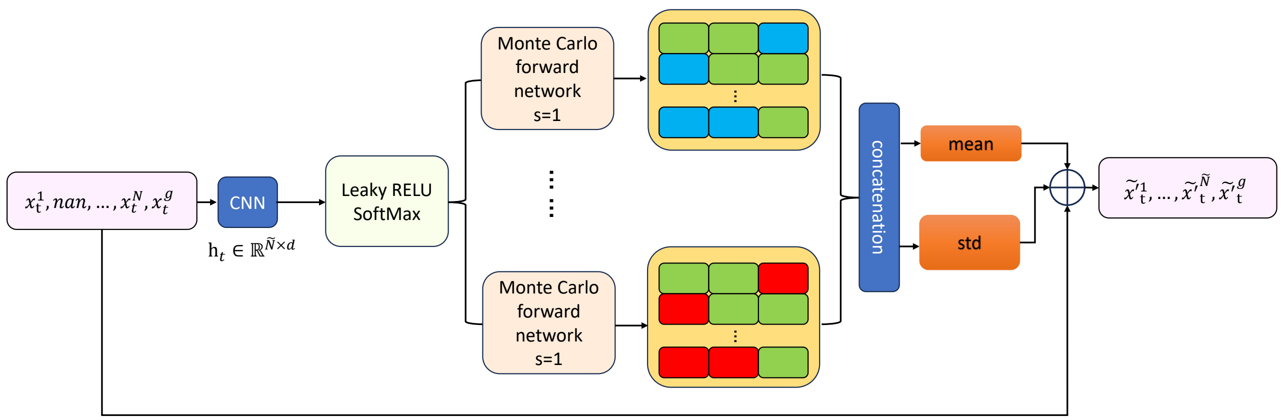

Figure 3: Architecture of the Exogenous Joint Modelling Structure (EJS)

Table 1: Predictions for $horizon = 12$ **using historical data with** $history\_len = 48$ **under different missing rates of variables, with results normalized.**

		Electricity			PEMS04			SML2010		
Model	Indicator	10%	20%	50%	10%	20%	50%	10%	20%	50%
LSTM	MAE	0.321	0.358	0.472	0.551	0.601	0.767	0.184	0.215	0.302
	RMSE	0.412	0.459	0.587	0.690	0.756	0.949	0.231	0.272	0.385
DA-SPS	MAE	0.287	0.312	0.398	0.489	0.536	0.687	0.162	0.189	0.267
	RMSE	0.368	0.401	0.512	0.613	0.673	0.860	0.205	0.240	0.342
IBN	MAE	0.254	0.278	0.341	0.440	0.480	0.607	0.138	0.162	0.221
	RMSE	0.325	0.356	0.442	0.553	0.603	0.759	0.174	0.205	0.283
MAD-STF (Ours)	MAE	**0.221**	**0.242**	**0.289**	**0.377**	**0.413**	**0.503**	**0.112**	**0.131**	**0.178**
	RMSE	**0.283**	**0.310**	**0.372**	**0.473**	**0.519**	**0.629**	**0.141**	**0.166**	**0.227**

Table 2: Dataset description, where n_{th} **denotes the** n**th variable in the dataset arranged in original order**

Dataset	Target Variable	Total Variable Count	Total Time Steps
Electricity	321_{th}	321	26,304
PEMS04	1_{th}	921	16,992
SML2010	16_{th}	20	57,600

Monte Carlo Dropout to a large number of missing channels, MAD-STF effectively mitigates this impact through its uncertainty-aware imputation mechanism, achieving scalable and robust performance under high-dimensional exogenous inputs.

Next, to further demonstrate the effectiveness of the model module, we designed ablation experiments for tasks with different missing rates in the SML2010 dataset. We selected the following two sets of experiments:

(1) **W/O TAS**: In the MAD-STF model, the TAS module is removed, and only the EJS module, which includes the target variable and covariates, is used for model prediction.
(2) **W/O EJS** Remove the EJS module from the MAD-STF model and use only the TAS module, which includes the target variable, for model prediction.

The estimated analysis results of the ablation experiment are presented in Table 3:

Based on the ablation experiment results, we can analyze and conclude that: (1) introducing target variables and covariates through the EJS module can indeed effectively enhance the predictive ability of the model; (2) removing TAS or the EJS module results in poorer

Table 3: MAD-STF ablation experiment results (results have been normalized).

Model Variant	Metric	Missing Rate		
		10%	20%	50%
W/O TAS	MAE	0.148	0.172	0.241
	RMSE	0.186	0.217	0.305
W/O EJS	MAE	0.135	0.156	0.212
	RMSE	0.170	0.197	0.268
MAD-STF (Full)	MAE	**0.112**	**0.131**	**0.178**
	RMSE	**0.141**	**0.166**	**0.227**

performance, indicating that historical data of the target variable is the main factor affecting the task's effectiveness.

6 Conclusion

This paper proposes a MAD-STF model by analyzing the task of predicting target variables with multiple variables in complex scenarios. The model utilizes two structures, TAS and EJS, to separately handle target variable sequences with partial missing data and covariate sequences with both partial and complete missing data, and ultimately integrates the two prediction results. Public datasets and ablation experiments demonstrate that the collaborative efforts of each module of the model ultimately achieve good prediction performance on public datasets.

However, one limitation of MAD-STF is that it does not explicitly learn the missing mechanism. In cases where the missing pattern is informative, such as non-random missingness arising from sensor overheating (MNAR). A promising research direction in the future is to introduce a missing mechanism prediction module for MAD-STF, such as by incorporating an auxiliary classifier and training with missing masks, to infer the missing pattern and adaptively adjust the imputation strategy accordingly.

References

[1] Ibtissam Amalou, Naoual Mouhni, and Abdelmounaim Abdali. 2022. Multivariate time series prediction by RNN architectures for energy consumption forecasting. *Energy Reports* 8 (2022), 1084–1091.

[2] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems* 33 (2020), 17804–17815.

[3] Mohamed Boujoudar, Massaab El Ydrissi, Mounir Abraim, Ibtissam Bouarfa, Omaima El Alani, Hicham Ghennioui, and El Ghali Bennouna. 2025. Comparing machine learning algorithms for imputation of missing time series in meteorological data. *Neural Computing and Applications* 37, 29 (2025), 23773–23787.

[4] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems* 31 (2018).

[5] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1 (2018), 6085.

[6] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[7] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2016. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems* 28, 10 (2016), 2222–2232.

[8] Hossein Hassani. 2007. Singular spectrum analysis: methodology and comparison. (2007).

[9] WANG Jiu-sheng, DAI Xu-hai, MIU Zhong-yan, TIAN Xiao-xia, and ZHOU Cheng. 2023. A model for time series prediction on highway traffic flow based on multivariate spatio-temporal relationship. *Journal of Highway and Transportation Research and Development* 40, 10 (2023), 175–182.

[10] Jinlong Li, Pan Wu, Hengcong Guo, Ruonan Li, Guilin Li, and Lunhui Xu. 2023. Multivariate transfer passenger flow forecasting with data imputation by joint deep learning and matrix factorization. *Applied Sciences* 13, 9 (2023), 5625.

[11] Tianhong Liu, Haikun Wei, Sixing Liu, and Kanjian Zhang. 2020. Industrial time series forecasting based on improved Gaussian process regression. *Soft Computing* 24, 20 (2020), 15853–15869.

[12] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023).

[13] Shusen Ma, Tianhao Zhang, Qijiu Xia, and Yun-Bo Zhao. 2025. IBN: An Interpretable Bidirectional-Modeling Network for Multivariate Time Series Forecasting with Variable Missing. *arXiv preprint arXiv:2509.07725* (2025).

[14] Ivan Marisca, Cesare Alippi, and Filippo Maria Bianchi. 2024. Graph-based forecasting with missing data through spatiotemporal downsampling. *arXiv preprint arXiv:2402.10634* (2024).

[15] Kasun Mendis, Manjusri Wickramasinghe, and Pasindu Marasinghe. 2024. Multivariate time series forecasting: A review. In *Proceedings of the 2024 2nd Asia Conference on Computer Vision, Image Processing and Pattern Recognition*. 1–9.

[16] Edward Appau Nketiah, Li Chenlong, Jing Yingchuan, and Simon Appah Aram. 2023. Recurrent neural network modeling of multivariate time series and its application in temperature forecasting. *Plos one* 18, 5 (2023), e0285713.

[17] Stefan Oehmcke, Oliver Zielinski, and Oliver Kramer. 2016. kNN ensembles with penalized DTW for multivariate time series imputation. In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2774–2781.

[18] Luís Sanhudo, Joao Rodrigues, and Enio Vasconcelos Filho. 2021. Multivariate time series clustering and forecasting for building energy analysis: Application to weather data quality control. *Journal of Building Engineering* 35 (2021), 101996.

[19] Victor Garcia Satorras, Syama Sundar Rangapuram, and Tim Januschowski. 2022. Multivariate time series forecasting with latent graph inference. *arXiv preprint arXiv:2203.03423* (2022).

[20] Philip Sedgwick. 2012. Pearson's correlation coefficient. *Bmj* 345 (2012).

[21] Qiuling Suo, Liuyi Yao, Guangxu Xun, Jianhui Sun, and Aidong Zhang. 2019. Recurrent imputation for multivariate time series with missing values. In *2019 IEEE international conference on healthcare informatics (ICHI)*. IEEE, 1–3.

[22] Agung Bella Putra Utama, Aji Prasetya Wibawa, Anik Nur Handayani, Wahyu Sakti Gunawan Irianto, Amat Nyoto, et al. 2024. Improving time-series forecasting performance using imputation techniques in deep learning. In *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*. IEEE, 232–238.

[23] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* 34 (2021), 22419–22430.

[24] Weizhong Yan, Hai Qiu, and Ya Xue. 2009. Gaussian process for long-term time-series forecasting. In *2009 international joint conference on neural networks*. IEEE, 3420–3427.

[25] Ye Yang and Jiangang Lu. 2023. Foreformer: an enhanced transformer-based framework for multivariate time series forecasting. *Applied Intelligence* 53, 10 (2023), 12521–12540.

[26] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[27] Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, and Yongjun Xu. 2024. Ginar: An end-to-end multivariate time series forecasting model suitable for variable missing. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 3989–4000.

[28] Tianhao Zhang, Shusen Ma, Yu Kang, and Yun-Bo Zhao. 2026. DA-SPS: A Dual-stage Network based on Singular Spectrum Analysis, Patching-strategy and Spearman-correlation for Multivariate Time-series Prediction. *arXiv preprint arXiv:2601.21381* (2026). arXiv:2601.21381 [cs.LG] doi:10.48550/arXiv.2601.21381

[29] Yifan Zhang, Rui Wu, Sergiu M Dascalu, and Frederick C Harris Jr. 2024. A novel extreme adaptive GRU for multivariate time series forecasting. *Scientific Reports* 14, 1 (2024), 2991.

[30] Jiawei Zhu, Qiongjie Wang, Chao Tao, Hanhan Deng, Ling Zhao, and Haifeng Li. 2021. AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. *Ieee Access* 9 (2021), 35973–35983.