ISSN 2616-5775 Vol. 8, Issue 3: 72-78, DOI: 10.25236/AJCIS.2025.080310

Time Series Forecasting for Charging Stations via Joint Learning of Missing Data and Temporal Dynamics

Mingzhao Zhu*

Nanjing University of Finance & Economics, Nanjing, China
mingzhao.zhu@qq.com
**Corresponding author*

Abstract: *Time series forecasting plays a crucial role in various real-world applications. However, the pervasive missing data caused by sensor failures, communication interruptions, and system malfunctions poses significant challenges to accurate forecasting. Existing forecasting methods that rely on imputation techniques often struggle to effectively preserve temporal dependencies and capture underlying patterns of missing data, thereby compromising forecasting accuracy and robustness. To address this issue, we propose a novel dual-stage framework that jointly learns missing data patterns and time series dynamics. It consists of (1) a pattern-aware encoder that captures missing value distributions and (2) a dual-forecasting module to enhance forecasting accuracy. Experimental results on real-world electric power data from charging stations demonstrate that our approach outperforms several baseline models, achieving superior forecasting performance under missing data conditions.*

Keywords: *Time series forecasting, Missing data pattern, Charging stations*

1. Introduction

Time series forecasting is a fundamental task in various domains, including finance, healthcare, and energy management. With the advancement of deep learning, forecasting models have significantly improved in capturing complex temporal dependencies and nonlinear patterns. However, in real-world applications, missing values frequently arise due to sensor failures, communication errors, or operational disruptions. For instance, in electric power monitoring systems at charging stations, data loss can occur due to network failures or equipment malfunctions. This results in incomplete time series data, which degrades forecasting performance.

Most existing forecasting methods are designed under the assumption of complete time series data. For example, Informer, the most classical time-series prediction model in the Transformer variant, was performed on datasets such as WHETHER, ETT (Electricity Transformer Temperature), etc., which had no missing values. When confronted with the ECL (Electricity Consuming Load) dataset with missing data, the researchers constructing Informer chose to transform the dataset to 2 years of hourly consumption as a way of circumventing the effect of missing values on the model. The rest of the algorithms are similar, mostly choosing complete datasets or avoiding missing data through some processing. When dealing with missing data, the common practice is to apply data imputation techniques as a preprocessing step. In addition to the more basic mean-filling, there are a number of statistical and machine-learning based methods such as MICE, KNNI, and TIDER; and more recently there have been a number of deep learning based filling algorithms such as GRU-D, BRITS, and so on. However, these approaches present several challenges: (1) The effect of data interpolation varies greatly in different datasets and contexts, and different scenarios require different missing value filling methods according to the reality, a situation that puts high demands on the researchers' judgement. (2) The more advanced depth filling methods can reasonably interpolate according to the temporal characteristics of the data and reduce the pressure of researchers' judgement; however, due to their complexity and high computational cost, they are not often adopted in the task of time series prediction and their popularity is limited. (3) The missing value processing method is used separately from the time-series prediction model, which causes the problem that the integration of missing values and prediction model is not close enough. The prediction model cannot fully utilise the intrinsic structure and missing patterns of the time-series data, which may lead to the accumulation of prediction errors. Especially when facing the task goal of long time series prediction, ignoring the long-term dependence and trend of the data can directly affect the

ISSN 2616-5775 Vol. 8, Issue 3: 72-78, DOI: 10.25236/AJCIS.2025.080310

accuracy of the model.

To address these issues, our research goal is to construct a model that not only achieves the time series forecasting objective, but also learns the missing patterns inherent in the time series data. Our contributions are as follows:

- To allow the model to learn the missing patterns in the sequence, we integrate missing patterns and time series data into the embedding layer of the model, followed by using a variational autoencoder to learn the time series and missing patterns simultaneously. In addition, to mitigate the influence of missing values during training, we use a loss function based on weight decay to reduce the impact of missing data on model training.Bottom: 2.54 cm;
- To improve the prediction performance, we adopt a dual prediction mechanism, specifically a decomposition prediction and fusion approach. In this method, a simple convolutional neural network (CNN) is used to capture the linear components of the time series, while a more complex network handles the nonlinear components. We offer two module options: one based on Fourier transform and another with time segmentation and fusion capabilities. The final step is to fuse both components, which improves the model's ability to fit time series data.
- We construct a real-world dataset of electricity consumption from charging stations based on actual order data. In the input phase, we manually mask parts of the data for experimentation to validate the model's effectiveness in real-world scenarios.

The remainder of this paper is structured as follows. Section 2 introduces the Related Work. In Section 3, we present an overview of our framework along with its key components. Section 4 details the experimental results. Section 5 concludes the paper.

2. Related Work

Time series forecasting techniques have shown significant application value in areas such as medical diagnosis, energy planning, weather forecasting and financial risk control. Initially, time series forecasting was largely based on statistical algorithms, leading to the development of classical methods such as ARIMA and the sliding window approach, which remain common strategies, or ensemble methods to improve model performance. With breakthroughs in machine learning and deep learning, researchers have improved neural network architectures, and these structural reorganisations and functional optimisations have greatly expanded the theoretical boundaries and application scenarios of time series modelling.

In terms of improvements to classical models, Lai et al. developed the LSTNet framework, which innovatively constructed a convolutional neural network without pooling layers to extract local time series features, followed by a GRU network with skip connections to capture long-range temporal dependencies.[1] The Transformer architecture proposed by the Vaswani team, although originally from the field of natural language processing, quickly attracted attention in the field of time series analysis due to its advantages in sequence modelling.[2] The improvements made by Zhou's team based on this architecture are particularly noteworthy, as they creatively employed a probabilistic sparse attention mechanism to effectively reduce the computational complexity of the model, while the generative decoder they designed broke the limitations of traditional step-by-step prediction methods.[3] Scholars such as Wu took a different approach, designing a time series decomposition unit to extract periodic features and innovatively replacing the traditional attention mechanism with autocorrelation operations, resulting in double improvements in computational efficiency and prediction accuracy. [4]

In recent years, time series forecasting research has diversified. By systematically decoupling the relationships between the time dimension and feature channels, the Li research team developed a forecasting framework with independent embedding modules, which significantly improved the representability of complex time series patterns. [5]The Wang team constructed a multi-frequency sampling framework that achieved collaborative modelling of multi-scale time series features through frequency domain decomposition and dynamic information fusion mechanisms. These innovative methods have broken the inherent paradigms of traditional architectures and brought new impetus to the field of time series analysis. [6]

The task of predicting time series with missing values has already been explored to some extent in the deep learning field. Zhengping Che et al. proposed GRU-D, a model based on gated recurrent units that simultaneously learns two representations of missing patterns: the mask and the time interval. These

ISSN 2616-5775 Vol. 8, Issue 3: 72-78, DOI: 10.25236/AJCIS.2025.080310

are integrated into the model architecture to capture missing value patterns and temporal dependencies in long time series. [7]Cristian Challu et al. proposed SpectraNet, which uses latent space spectral decomposition to estimate missing data while performing time series forecasting. Although this study guarantees good prediction performance even under extreme missing data conditions, the study set the prediction window size at 24 time points, which belongs to short and medium term forecasting, leaving the task of long-term time series forecasting still to be explored. [8]Chengqing Yu et al. proposed a new graph interpolation attention recursive network, replacing the fully connected layer of simple recurrent units with interpolation attention and adaptive graph convolutions to recover all missing variables and reconstruct the correct spatiotemporal dependencies. [9]However, this method was developed for spatio-temporal data with missing values and has not yet been adapted to multivariate time series data.

3. Methodology

3.1. Problem Statement

The aim of this paper is to solve the problem of Long-Term Time Series Forecasting (LSTF) under missing values. Given a multivariate time series $Xn = \{x1, x2, \dots, xn\}$ sampled at regular intervals, we aim to construct an appropriate model that, when given a sequence with missing values Xn as input, can output the next m time points of the multivariate time series $Xm = \{xn + 1, xn + 2, \dots, xn + m\}$, where $m \geq 24$.

3.2. Model

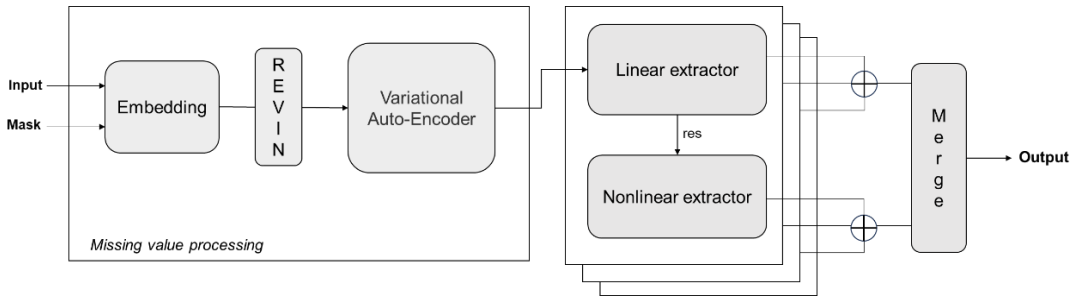

Figure 1: General framework of the model. The model consists of two parts: a missing value processing module and a time series prediction module.

As shown in Figure 1, the main part of the model consists of two parts. The first part is the missing pattern processing module, which employs a special Embedding to dimensionally expand and combine both the original input sequence and the mask representing the missing pattern, followed by the REVIN method to process the resulting hybrid input, and finally the Variational Auto-Encoder VAE is used to capture the underlying structure of the input data. The second part serves as the main network module for time series prediction, and the structure is predicted jointly using several sub-modules as a way to ensure the model's ability to capture complex time series dependencies.

3.2.1. Missing Value Processing Module

The model constructed in this paper can accept two parts of input: the initial data X and its missing value matrix $Mask_x$ where the missing value matrix has the same size as the initial data and contains only two variables, 0 and 1. If the value in the Mask matrix is 0, it indicates that the data at that position is missing; otherwise, it means the data at that position is not missing.

To enable the model to learn the missing value patterns, we designed a missing value processing module, as shown in Figure 2, to adapt to the time series missing scenario. Unlike the time series preprocessing work that fills missing values in advance, we aim for the model to deeply learn the missing patterns through this module, so the information in the mask matrix needs to be effectively utilized.

In time series forecasting tasks, the most commonly used embedding is proposed by Informer, which includes the input feature sequence, positional embeddings, and specific information at each time point. However, in this task, considering that the module is designed for data processing, the embedding design should pursue simplicity. To avoid unnecessary complexity, the feature embedding only focuses on the input sequence and mask matrix, which are mapped to the same embedding space through linear transformation and then weighted and fused. The formula is expressed as: $Z = Embedding(X, Mask_x)$.

ISSN 2616-5775 Vol. 8, Issue 3: 72-78, DOI: 10.25236/AJCIS.2025.080310

In this case, the model does not need to consider many additional time series structures, allowing it to focus more on learning the missing patterns.

The Variational Autoencoder (VAE) is a generative model that can model the joint embedding obtained through the latent space, thus learning the latent features of the sequence. As shown in the figure, the VAE is divided into three main parts: encoder, latent space, and decoder. The latent space primarily learns the parameters of the probability distribution, such as the mean μ, variance σ, and random noise ϵ sampled from the standard normal distribution, to learn the latent distribution. By incorporating the variational autoencoder into the model, we can leverage its modeling capability in the latent space to capture the latent missing patterns in the data and generate possible missing values or fill in the missing parts.

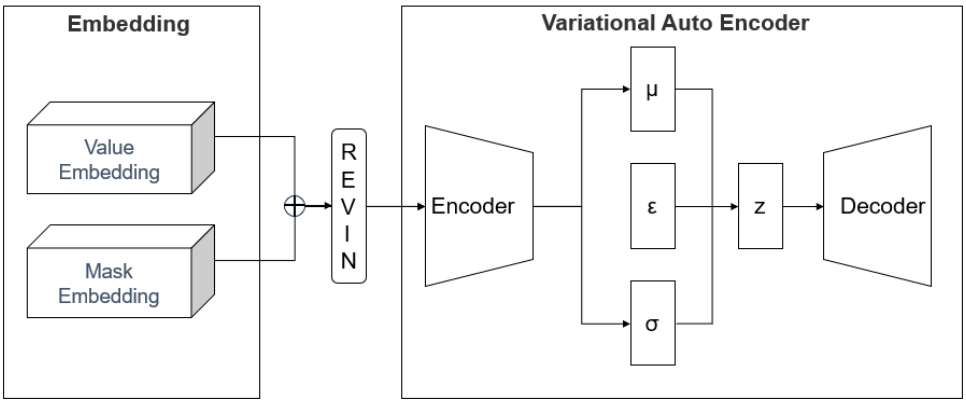

Figure 2: Missing value processing module, consisting of an Embedding layer that receives input from both parts, REVIN normalization, and a Variational Autoencoder module.

Although this module processes the data with respect to missing values and achieves the goal of embedding missing features and variational learning, from the perspective of long-term time series forecasting tasks, the module has poor ability to capture trends and seasonality in long time series and cannot capture the long-term dependencies in time series. To address this shortcoming, we added a dedicated time series processing module after this module to achieve high-precision time series forecasting.

3.2.2. Timing Prediction Module

This module receives the output of the missing value processing module and captures the hidden relationships in the time series data. This module consists of multiple layers of sub-modules stacked together, and the interior of the sub-modules is divided into two channels, the first part is a simple linear prediction channel, and the second part is a complex feature extraction channel, which are related by residual links.

In the simple linear prediction channel, we use a one-dimensional convolutional network suitable for linear feature extraction of data and providing stable output when dealing with high-dimensional time-series data. The linear component extracted by this network allows us to obtain a linear prediction at that level, and the nonlinear component obtained by subtraction is subsequently passed to the complex feature extraction channel for modeling.

ISSN 2616-5775 Vol. 8, Issue 3: 72-78, DOI: 10.25236/AJCIS.2025.080310

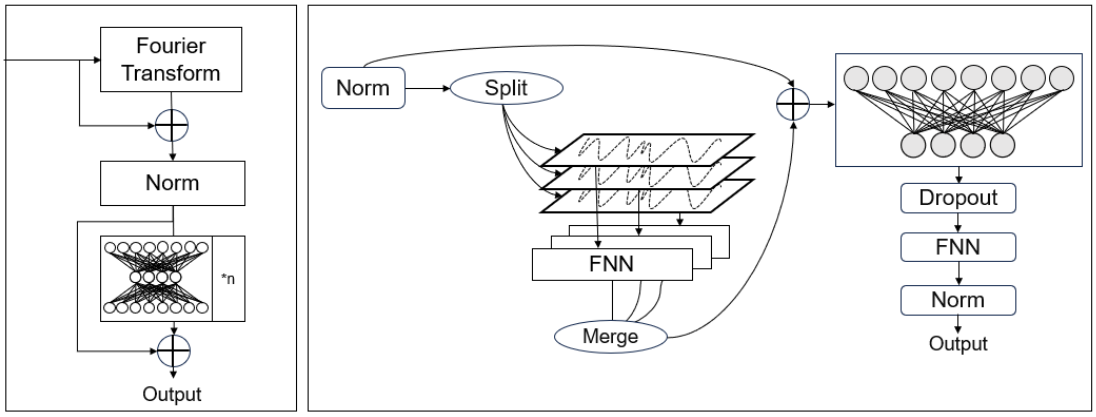

Figure 3: Two possible modeling approaches for complex feature extraction channels, nonlinear modeling based on Fourier variation (left) vs. nonlinear modeling based on temporal slicing fusion prediction (right)

In the complex feature extraction channel, the submodule selects a more complex network for prediction. In order to adapt to different types of time series, we provide two different schemes, the first one is nonlinear modeling based on temporal slice fusion prediction, and the second one is nonlinear modeling based on Fourier transform. The structure of the two complex nonlinear feature extractors is shown in Figure 3. Among them, the implementation process of nonlinear modeling based on time-sliced fusion prediction is:

(1) Split the input two types of sequence information of position and placement into multiple sub-sequences according to the sampling rate;

(2) Extract and learn the timing information of each sequence using MLP. As a combination of linear layer and nonlinear activation function, this module maps the input data to a higher dimensional space for feature extraction, and finally back to the original dimensional space;

(3) The segmented sequence information is obtained by merging in the original order, which is used as the output of forward propagation.

The implementation process of nonlinear modeling based on Fourier transform is:

(1) The input sequence is Fourier transformed to extract its frequency domain features;

(2) The normalized frequency domain features are nonlinearly transformed using a multilayer perceptron to extract and learn the timing information. This module maps the input data to a higher dimensional space for feature extraction through a combination of linear layers and GELU activation functions, and finally maps back to the original dimensional space;

(3) The MLP processed results are residually concatenated with the original inputs and layer normalized to obtain the final output.

After each sub-module returns its linear prediction and non-linear prediction results accordingly, we collect the returned prediction results separately, accumulate them one by one, and further fuse the total results of linear prediction and non-linear prediction at the end of the sub-module arithmetic to generate the final prediction results.

4. Experiments

4.1. Dataset

We process the tram charging order data provided by the Star Charging platform based on the valid information therein. The specific process is as follows: the start time of the order is processed in chunks, the order charging volume data is filtered out, and integrated with the information of each charging station, and the total hourly power consumption of the 10 charging stations in Nanjing from January 1, 2023 to July 19, 2024 at the power station is finally obtained.

Although charging is a time-sensitive scenario, considering the large scale of the stations, the content

ISSN 2616-5775 Vol. 8, Issue 3: 72-78, DOI: 10.25236/AJCIS.2025.080310

of 0 in the dataset is about 10%, which is not consistent with the intermittent characteristics. In order to obtain data containing certain missing values, we choose to randomly generate a mask matrix and obtain experimental data by controlling the proportion of 0 values in it. In this experiment we control the content of 0 in the mask matrix by setting 10%, 20% & 30%.

4.2. Setups

During the training process, we set up five window sizes for the predictions, which are 24, 48, 168, 336, and 720 time nodes afterward, corresponding to the five different day criteria of {1d, 2d, 7d, 14d, 30d, 40d}.

Our model epoch was set to 50 to ensure sufficient time for optimization and convergence; the Adam optimizer was used, with the initial learning rate set to 0.0001 and the batch size set to 32. The model was trained in an environment where Pytorch was deployed on a server equipped with NVIDIA GeForce RTX 3090 GPUs. server.

In terms of Baseline, we selected four models as references, including two types of Transformer-based models, Informer and Autoformer, a CNN-RNN-based fusion time series prediction model, LSTNet, and a model for intermittent time series data, Mixformer. in terms of evaluation metrics, we selected the models that are commonly used in time series prediction tasks. We select MSE and MAE, which are often used in time-series prediction tasks, as evaluation metrics. In terms of loss function, we design a weighted MSE loss function to give lower weight to the prediction error at the location of missing values, so that the model pays more attention to the real data while not completely ignoring the missing parts, making the model still robust in the case of missing values. The formula is as follows:

$$Loss = \frac{\sum_i \left[\left(y_{pred}^{(i)} - y_{true}^{(i)} \right)^2 \cdot \left(mask^{(i)} + \lambda \left(1 - mask^{(i)} \right) \right) \right]}{\sum_i \left(mask^{(i)} + \lambda (1 - mask^{(i)}) \right)} \quad (1)$$

Where λ is the weight assigned to the missing part and $mask^{(i)}$ denotes the missingness of the data.

4.3. Results

We selected five prediction windows with mask rates of 10%, 20% & 30% in turn on a real dataset of hour-by-hour electricity usage at charging stations to evaluate the performance of our proposed model and the four benchmark models, respectively. The experimental results are shown in Table 1, where bold represents the optimal results and underlined represents the sub-optimal results. The experimental results show that on the charging station electricity usage dataset with missing values, our model achieves better results in experiments with different prediction lengths.

Table 1: Multivariate time series prediction results with different mask rates. Lower MSE and MAE indicate better prediction results.

Pred_Len	24		48		168		336		720	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
mask=0.1										
Our	**0.349**	**0.456**	**0.332**	**0.445**	**0.328**	**0.441**	**0.333**	**0.443**	**0.338**	**0.448**
Mixformer	0.457	0.525	0.460	0.530	0.522	0.559	0.555	0.581	0.651	0.636
Autoformer	0.469	0.533	0.465	0.532	0.520	0.557	0.588	0.591	0.493	0.539
Informer	0.515	0.568	0.635	0.642	0.912	0.792	0.808	0.731	0.879	0.776
LSTNet	0.875	0.803	0.880	0.806	0.896	0.814	0.906	0.819	0.903	0.817
mask=0.2										
Our	**0.359**	**0.469**	**0.344**	**0.461**	**0.336**	**0.455**	**0.340**	**0.457**	**0.340**	**0.457**
Mixformer	0.452	0.527	0.470	0.546	0.497	0.545	0.528	0.574	0.592	0.614
Autoformer	0.466	0.533	0.441	0.520	0.542	0.572	0.451	0.524	0.459	0.527
Informer	0.486	0.551	0.578	0.610	0.708	0.668	0.721	0.676	0.735	0.683
LSTNet	0.797	0.758	0.805	0.762	0.818	0.769	0.827	0.775	0.825	0.773
mask=0.3										
Our	**0.359**	**0.471**	**0.346**	**0.468**	**0.337**	**0.464**	**0.338**	**0.462**	**0.336**	**0.459**
Mixformer	0.424	0.511	0.445	0.532	0.459	0.527	0.492	0.560	0.527	0.576
Autoformer	0.441	0.517	0.426	0.508	0.468	0.535	0.409	0.505	0.430	0.512
Informer	0.456	0.534	0.518	0.554	0.612	0.604	0.633	0.615	0.638	0.618
LSTNet	0.720	0.712	0.731	0.718	0.740	0.722	0.747	0.726	0.745	0.725

ISSN 2616-5775 Vol. 8, Issue 3: 72-78, DOI: 10.25236/AJCIS.2025.080310

As shown in Table 1, our proposed model improves on both types of evaluation metrics over five different time span windows. At the same time, our model plays more consistently on the dataset and does not show significant performance degradation with the change of prediction length as in Informer.

In addition, we also compare the training time of the baseline method and the proposed method, and the results are shown in Table 2. The results show that compared to some of the RNN-based models and Transformer-based models, our method is demonstrating better training efficiency. As an example, the efficiency of each model for long time series prediction with 720 timesteps is as follows:

Table 2: Comparison of modeling time, using the time spent on one experiment with prediction step = 720 and mask rate = 0.3 as an example.

Model	Single epoch time consumption	Number of epochs	Total training duration
Our	2.57	8	20.56
LSTNet	3.01	16	48.16
Mixformer	33.74	8	269.92
Autoformer	55.18	5	277.4
Informer	21.99	14	307.86

5. Conclusion

In our study, we designed a time-series prediction model that not only predicts time series but also learns potential missing data patterns. We learned the missing patterns of the data using Embedding with Variational Autoencoder that integrates the missing patterns, and subsequently combined the dual prediction mechanism of linear and nonlinear models for time series prediction. Experiments were conducted on real data of charging station electricity usage and the results validate the effectiveness of our model in handling missing values with practical applicability in real prediction tasks. Future work can explore further optimization of the model as well as extension to other domains with incomplete data.

References

[1] Lai, G., Chang, W.C., Yang, Y., et al. (2018) Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 95-104.

[2] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017) Attention Is All You Need. Advances in Neural Information Processing Systems, 30.

[3] Zhou, H., Zhang, S., Peng, J., et al. (2021) Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 35(12), 11106-11115.

[4] Wu, H., Xu, J., Wang, J., et al. (2021) Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. Advances in Neural Information Processing Systems, 34, 22419-22430.

[5] Li, Z., Rao, Z., Pan, L., et al. (2023) Mts-Mixers: Multivariate Time Series Forecasting via Factorized Temporal and Channel Mixing. arXiv preprint arXiv:2302.04501.

[6] Wang, S., Wu, H., Shi, X., et al. (2023) TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. The Twelfth International Conference on Learning Representations.

[7] Che, Z., Purushotham, S., Cho, K., et al. (2018) Recurrent Neural Networks for Multivariate Time Series with Missing Values. Scientific Reports, 8(1), 6085. DOI: 10.1038/s41598-018-24271-9.

[8] Challu, C., Jiang, P., Wu, Y.N., et al. (2022) SpectraNet: Multivariate Forecasting and Imputation Under Distribution Shifts and Missing Data. arXiv preprint arXiv:2210.12515.

[9] Yu, C., Wang, F., Shao, Z., et al. (2024) Ginar: An End-to-End Multivariate Time Series Forecasting Model Suitable for Variable Missing. Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 3989-4000.