Few-Shot Financial Time Series Forecasting via Meta-Learning and Temporal Attention

Qihan Wang*

School of Computer Science & Technology, Beijing Jiaotong University, Beijing, China

Abstract. In real-world applications, there are challenges to forecasting a financial time series due to its natural non-stationarity, presence of noise, and lack of labeled data. In particular, it is often found that newly listed or low-liquidity assets have significant data shortage problems, making it difficult to effectively generalize the commonly used deep learning model. This paper address this issue by casting the problem of financial forecasting as a few-shot learning setting and propose a meta-learning approach to improving the adaptability of the models under few-shot settings. Specifically, a framework based on a Transformer is adopted as the main model to learn the temporal long-range dependency in multivariate time series. A meta-learning approach is used to learn a transferable parameter initialization on different financial tasks, to improve its efficiency in low-data scenarios. This enables the model to adapt to new assets very fast, based on a very limited set of support samples. The proposed approach is tested on the Huge Stock Market Dataset where each stock is a task in different few shot setting (K=1, 3, 5, 10). Experimental results demonstrate that the proposed Meta-Baseline significantly outperforms the conventional baselines (including Transformer, LSTM and GRU) in each case. In the toughest scenario (K=1), the suggested approach decreases the prediction error by more than 70% when compared to typical Transformer models. Furthermore, more support samples in the model result in steady and consistent improvement in stability of the model. These findings highlight the effectiveness of meta-learning for few-shot financial time series prediction and suggest its potential for real-world applications in data scarce environments.

1 Introduction

Forecasting financial time series has become a key challenge in quantitative finance, with numerous uses in portfolio management, risk management, and algorithmic trading. Recently, deep learning models have significantly propelled the effectiveness of time series forecasting via complex temporal relationship. Models like Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and some other architectures based on Transformers have been shown for impressive skills in modeling sequential data [1]. Particularly, Transformer, utilizes its self-attention mechanism, performing better capability in capturing long-range dependencies to some extent. [2,3]. Although current methods have these advancements, they depend on plenty of historical data. However, in many real-world financial scenarios, such as newly listed stocks or emerging assets, historical data is extremely limited, which creates a difficult “cold-start” issue, where traditional deep learning models often overfit and struggle to generalize [4,5]. Therefore, improving prediction performance under limited data has become an important and practical research issue.

To deal with data limitations, few-shot learning has been tried in areas like computer vision and natural language processing. The core idea of this concept is to enable models to learn from only a few samples [6]. In various approaches, meta-learning, especially Model-Agnostic Meta-Learning (MAML), has gained significant attention [7]. MAML focuses on discovering an effective starting point, which can be rapidly adjusted for new tasks with a few training examples. Despite meta-learning has shown reliable results in some areas like computer vision, its application to financial cases is still relatively limited [8,9,10]. Recent studies explore spatial temporal interactive attention as well in order to improve model multivariate dependencies across features and time steps [11], but such methods still need abundant training data and their performances in few-shot scenarios are unknown. Plus, most of the existing studies focus either on model architecture or training strategies, but a systematic analysis of interactions with few-shot and meta-learning settings. It is still unclear whether more complex models, perform better than those simpler models in data scarce conditions.

This paper propose a method to address the problem of few-shot financial time series prediction by combining Transformer-based models and meta-learning techniques in this paper. First, build a baseline Transformer encoder model for sequence forecasting. Then, several architectural additions have been proposed, such as position encoding, feature-level attention mechanisms and regularization strategies like dropout, to improve model performance with a small amount of data. Next, I will introduce the MAML meta-learning training paradigm and have the model learn

*23722064@bjtu.edu.cn

generalisations over many financial tasks for quick adaptation to new data.

Different Model Structures and Training Strategies' Effects on the Accuracy and Stability of Predictions in Various K-Shot Settings Are the subjects of this paper, where K is the number of labelled training samples. All the samples have the same length for the historical window and the corresponding future prediction targets, and they simulate a real-world scenario where only a small amount of data is available for model training. Numerous experiments have been carried out on multiple sets of financial data to offer practical ideas for constructing high-efficiency and stable models of few-shot time series forecasting.

2 Method

2.1 Dataset

This paper used Huge Stock Market Dataset obtained from Kaggle in the experiment. This classical dataset provides historical daily trading data of U.S. stocks and ETFs listed on major exchanges including NYSE and NASDAQ. The original datasets are very large. In this study, only five independent financial time series are selected to simulate the data limited condition, each corresponding to a distinct asset and treated as an individual task. Every dataset consists of five standard market indicators, which are Open, High, Low, Close, and Volume (OHLCV) respectively.

All features are normalized using Z-score applied independently to each time series, which is helpful with eliminating scale discrepancies across assets:

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

where μ and σ represent the mean and standard deviation of the corresponding asset. In the evaluation phase, predictions are reverted into the original scale so that maintain clarity.

A sliding window approach is used to convert unprocessed time series into supervised learning samples. This paper also set an input length $L = 24$ and a prediction horizon $H = 12$, which makes each sample is constructed by mapping a historical window to its subsequent future values:

$$[x_{t-23}, \dots, x_t] \rightarrow [y_{t+1}, \dots, y_{t+12}] \quad (2)$$

For each task, the dataset is chronologically split into training (80%) and testing (20%) subsets. A small support set containing $K \in 1,3,5,10$ samples is drawn from the training set for adaptive training for a specific task. The unused samples in the training set are used as the query set for meta-training. The test set is reserved chronologically (20%) and will be used for the final evaluation.

2.2 Problem formulation

Each financial asset is modeled as a multivariate time series like:

$$\mathcal{X} = x_1, x_2, \dots, x_T, x_t \in \mathbb{R}^d \quad (3)$$

where $d = 5$ corresponds to the OHLCV features. $X_t = [x_{t-L+1}, \dots, x_t]$, the historical window, whose objective is to learn a parametric function f_θ so that predicts future closing prices over a horizon H:

$$\hat{Y}_t = f_\theta(X_t) \in \mathbb{R}^H \quad (4)$$

This paper train the model by minimizing the expected squared error over the data distribution:

$$\mathcal{L}(\theta) = \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\| f_\theta(X) - Y \|_2^2] \quad (5)$$

Unlike conventional settings with abundant data, because of the lack of data, each time series is treated as a task $\mathcal{T}_i$, composed of a support set $\mathcal{D}_i^{\text{support}}$ and a query set $\mathcal{D}_i^{\text{query}}$. Tasks are assumed to be drawn from an underlying distribution $p(\mathcal{T})$, which reflects the diversity of financial dynamics across assets.

A meta learning framework is adopted, which makes learning much more efficient. Instead of directly optimizing model parameters for a single task, the objective is to learn an initialization that can be rapidly adapted to a new task using only a few gradient updates, thus generalization in low-data regimes is improved.

2.3 Model architecture

An encoder model in the Transformer considers all financial indices at the same time. For an input sequence, each observation is first mapped to a latent representation space by a linear transformation $h_t = W_{in}x_t + b$, where the embedding dimension is set ($d_{model} = 64$). Positional encodings are subsequently incorporated to maintain the sequence's temporal order.

The sequence that has been encoded is handled by a series of Transformer encoder layers, and each layer consists of multi-head self-attention and feed-forward networks that are position-aware. Two Encoder Layers with four attention heads are used in this paper. Self-attention computes contextualised representations by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

which can capture the extended-time dependency that frequently appears in financial time series.

Temporally modelling is used to extend the representation learning of features in other dimensions. Specifically, an attention weight vector is computed as follows:

$$\alpha = \text{softmax}(W_f h_t) \quad (7)$$

and applied element-wise multiplication to the hidden representation to get $\tilde{h}_t = \alpha \odot h_t$. Thus, the model will give greater weight to important features and ignore less significant noise.

To reduce overfitting in a few-shot scenario, a dropout regularisation of 0.2 is added after the attention and feed-forward layers. The final hidden representation is mapped to the prediction space via a linear output layer, and multiple time steps of forecasts are obtained. The overall diagram of the proposed system is shown in Figure 1. The two main parts are a time-feature extractor that uses an improved Transformer block and a meta-learning optimisation strategy. As shown in the figure, every financial task is divided into a support set for inner-loop adjustment and a query set for outer-loop parameter refinement; thus, a generalisation-enabled initialisation θ can be obtained quickly and adapted to various market environments (e.g., T_1 to T_n).

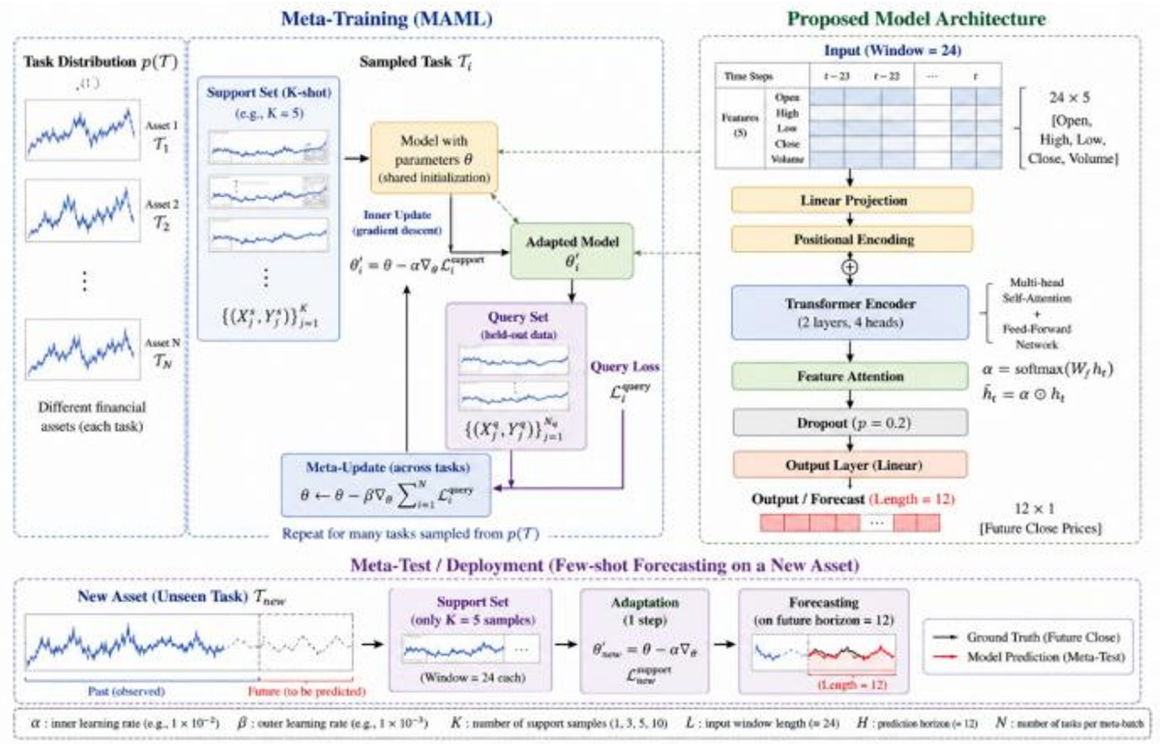

Figure 1. Overall framework of the proposed meta-learning framework for few shot financial time series forecasting (Picture credit: Original)

2.4 Evaluation metrics

The forecasting task is evaluated as a regression problem on the query set. Due to the extremely limited number of samples in few-shot scenarios, evaluation metrics must be robust to small-sample variability. In particular, the coefficient of determination (R^2) is not adopted, as it depends on the variance of the ground truth values:

$$R^2 = 1 - \frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2} \tag{8}$$

When the sample size is small, the denominator becomes unstable, leading to unreliable and highly fluctuating scores.

For this particular reason, the evaluation focuses on two widely used regression metrics, which are the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). MSE quantifies the average squared difference between predicted values and actual results:

$$\text{MSE} = \frac{1}{n}\sum (y_i - \hat{y}_i)^2 \tag{9}$$

the Mean Absolute Error (MAE) offers a clearer metric for understanding the average size of errors:

$$\text{MAE} = \frac{1}{n}\sum \mid y_i - \hat{y}_i \mid \tag{10}$$

All of the metrics are computed on the query set and averaged over multiple independent runs, which aims to assure both the mean and standard deviation can reflect predictive accuracy and stability.

2.5 Training strategy

The two training systems are as follows. In the standard fine-tuning method, randomly initialise the model parameters and then optimise them separately for each task by minimising the regression loss of the given samples. Few-shot scenarios often result in overfitting in this way due to a lack of data.

A model-agnostic meta-learning (MAML) method is used to address the above problem. MAML has been used to enhance the generalization ability of transfer learning in regression problems [12], and this work will be applied to few-shot financial time series regression.

The aim is to learn a shared initialization that can be efficiently adapted to new tasks. For each task $\mathcal{T}_i$, model parameters are first updated on the support set via a gradient descent step:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{\text{support}}(\theta) \tag{11}$$

where the inner-loop learning rate is set to $\alpha = 10^{-2}$ and a single adaptation step is performed.

The revised parameters are subsequently assessed on the query set, and the universal parameters are refined across tasks based on:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}^{\text{query}}(\theta_i') \tag{12}$$

where the outer-loop learning rate is $\beta = 10^{-3}$. Training is performed using the Adam optimizer, and all results are averaged over multiple runs with different random seeds to ensure robustness.

3 Experiments

3.1 Experimental setup

Five representative assets with different price ranges and time trends were selected to set up individual forecasting problems, and the evaluation should cover all the states of finance.

Following the preprocessing pipeline in Section 2, at each time, a supervised sample of length $L = 24$ and a

prediction horizon of $H = 12$ was generated for the time series. Individually Z-score normalise each feature for a specific asset, and then revert the prediction back to its original scale for evaluation.

To simulate a few-shot learning scenario, each dataset was divided chronologically into an 80% training set and a 20% test set. During adaptation, a support set of K samples was taken from the training set, where $K \in 1,3,5$. The unused samples from the test set served as the query set for model generalisation evaluation. This setting shows a real-life situation where little historical data is available for the new asset.

The four models in this paper were a Transformer-based baseline, LSTM, GRU and the proposed Meta-Baseline model. The first model was trained separately on all task datasets using traditional fine-tuning, and then, in a gradient-based meta-learning method, a meta-baseline was trained across multiple tasks. A first-order MAML approximation was used for meta-training, and a single inner-loop adaptation step and a fixed learning rate were set for both inner and outer updates. All models were optimising via the Adam optimizer, and multiple runs had been carried out using different random seeds for stability.

3.2 Quantitative results and analysis

It is clear as Table 1 shows that the quantitative outcomes of all models across various few-shot configurations. The proposed Meta Baseline consistently exceeds the performance of almost all of the baseline models for every K value, which significantly presents its enhanced ability in managing time series forecasting tasks with limited data.

Table 1. Performance comparison under different few-shot settings

Model	K	MSE (↓)	MAE (↓)
Transformer	1	3.8408	1.6305
	3	2.0001	1.1283
	5	1.0972	0.8088
	10	0.6043	0.5769
LSTM	1	3.7713	1.8286
	3	2.5790	1.3078
	5	1.0633	0.7489
	10	1.7245	0.7906
GRU	1	1.6339	1.0562
	3	0.8511	0.7460
	5	0.8193	0.8238
	10	1.0324	0.7061
Meta-Baseline	1	0.8875	0.6564
	3	0.4595	0.3790
	5	0.2068	0.2934
	10	0.1409	0.2313

Most models will perform better under increased support sample sizes, generally speaking. A typical case is the Transformer baseline, which, as K increases from 1 to 10, reduces the Mean Squared Error (MSE) from 3.8048 to 0.6043. This behaviour meets the expectations and, as a result, more training data generally provides richer temporal information for model fitting. However, even with this improvement, the Transformer still has a relatively large variance and high standard deviation (e.g., 3.9551 at K=1), indicating that it is unstable under very limited data.

LSTM and GRU are recurrent models that perform better than the Transformer baseline in low-shot settings, as shown below. For example, GRU achieves an MSE of 1.6339 at K=1 and is better than both Transformer and LSTM. Therefore, recurrent architectures are relatively more inductive to modelling the sequence when data is limited. However, both LSTM and GRU show a decline in performance as K increases. Their performance is not monotonic, and for some values of K, it has even decreased at K=10. The above problems are all typical issues that arise in the context of few-shot learning under stochastic sampling and limited training data.

The new Meta-Baseline model exhibits an abnormal behaviour. It has the lowest prediction errors in all few-shot settings, and the MSE values for K=1,3,5, and 10 are 0.8875, 0.4595, 0.2068, and 0.1409, respectively. At the same time, the quality of the performance will also be higher with a larger K. It shows that the meta-learned initialisation has effectively extracted transferable knowledge across tasks and can quickly adapt with a very small amount of support data.

A closer look shows the amount of improvement achieved by the new way. In the most difficult setting (K=1), Meta-Baseline reduces the MSE by about 76.7% compared with Transformer baseline (from 3.8048 to 0.8875) and by around 45.7% compared with GRU (from 1.6339 to 0.8875). The reduction is relatively small; thus, meta-learning has not solved the data scarcity problem. Even in a relatively less constrained setting of K=10, the Meta-Baseline still has a noticeable lead and reaches an MSE of 0.1409; all the other baseline models perform worse.

Another required quality of a good few-shot learner is also its stability. The Transformer baseline has a large standard deviation and is thus highly sensitive to data sampling and initialisation. Although the standard deviations of all the models were not reported, the consistently low error values of the Meta-Baseline across all K settings indicate that it is relatively stable and generalisable.

Generally speaking, according to the above experimental results, the traditional model cannot perform well in situations of very sparse data and often shows high variance and instability. The above meta-learning framework can also use cross-task knowledge to learn an adaptable parameter initialisation. Thus, the model will be more accurate and robust in a few-shot financial time series prediction scenario.

3.3 Visual analysis

This paper made Figure 2 owing to present forecasting trajectories for five representative assets under the $K = 5$ setting, which can enhance the demonstration of the proposed method's effectiveness.

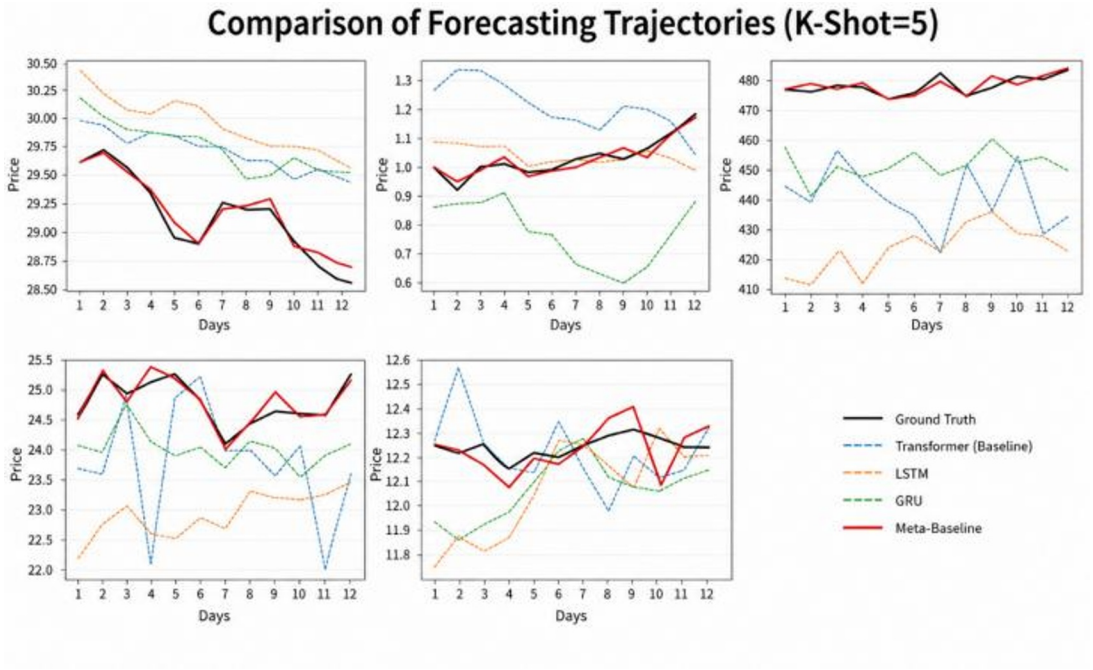

Figure 2. Comparison of forecasting trajectories across five representative stocks under the K=5 setting (Picture credit: Original)

As seen from the figure2, the Meta-Baseline model can always make the prediction that is close to the ground truth for all the assets. Specifically, it is possible to successfully model both the global trend and local fluctuations even in the presence of noise and non-stationary behaviour. This suggests that the learned initialization captures meaningful temporal patterns with good generality over tasks.

The baseline models show significant variations on the other hand. Forecasts of future values made by the Transformer baseline are frequently too low or too high, particularly in volatile areas. The trajectories of LSTM and GRU are smoother, but they are not able to capture the quick changes, indicating minimal sensitivity to short-term dynamics. This is more pronounced in more volatile assets, revealing some of the limitations of traditional sequence models in low data regimes.

The visual results are consistent with the quantitative ones, providing strong support that the proposed method achieves improved forecasting accuracy and stability for few-shot scenarios.

4 Conclusion

This paper puts forward a meta-learning framework for limited-sample prediction of financial time series and addresses the problems of data scarcity and cross-asset diversity. Each financial time series is regarded as an independent problem, a gradient-based meta-learning method is used, and an easily adjustable transferable initial point can be obtained with little data.

Based on experiments, the Meta-Baseline has shown significant performance improvements over previous models such as Transformer, LSTM and GRU in various few-shot tasks. In short, the above method has reduced the prediction error significantly under the condition of very little data, and can also capture generalisable temporal features. Visual analysis also shows that the model's predictions are relatively close to the actual values, and both the overall changes and regional variations have been reproduced.

Although the above studies have made progress, some problems remain. Only the univariate prediction of closing prices is carried out in the present framework; models for inter-asset relationships and external factors such as macroeconomic indicators have not been built. The meta-learning process is also relatively computationally expensive, and thus it may not be suitable for large-scale applications.

Future work will study the inclusion of cross-asset dependencies, expand the framework to multivariate prediction targets, and improve the efficiency of meta-training. Add a probabilistic forecasting method to determine the degree of uncertainty in the results of financial decision-making.

In short, this paper shows that meta-learning can effectively address the problem of few-shot financial forecasting by providing theoretical support and practical applications for data-scarce situations.

References

1. Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379(2194).
2. Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in Neural Information Processing Systems, 34, 22419-22430.
3. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021, May). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI

Conference on Artificial Intelligence (Vol. 35, No. 12, pp. 11106-11115).

4. Wood, K., Kessler, S., Roberts, S. J., & Zohren, S. (2023). Few-shot learning patterns in financial time-series for trend-following strategies. arXiv preprint arXiv:2310.10500.
5. Blasco, T., Sánchez, J. S., & García, V. (2024). A survey on uncertainty quantification in deep learning for financial time series prediction. Neurocomputing, 576, 127339.
6. Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys, 53(3), 1-34.
7. Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International Conference on Machine Learning (pp. 1126-1135). PMLR.
8. Gregnanin, M., Smedt, J. D., Gnecco, G., & Parton, M. (2023, September). Stock price time series forecasting using dynamic graph neural networks and attention mechanism in recurrent neural networks. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 357-373). Cham: Springer Nature Switzerland.
9. Vaiciukynas, E., Danenas, P., Kontrimas, V., & Butleris, R. (2021). Two-step meta-learning for time-series forecasting ensemble. IEEE Access, 9, 62687-62696.
10. Zhang, H., Guo, H., Bai, H., Zhang, C., & Li, L. (2025). MAML-based temporal supervised information maximizing GAN for few-shot time series data generation. Expert Systems with Applications, 129342.
11. Wei, B., Hei, Y., & Wan, Y. (2025). Attention-based spatial-temporal interactive couple neural networks for multivariate time series forecasting. Information Sciences, 122647.
12. Satrya, W. F., & Yun, J. H. (2023). Combining model-agnostic meta-learning and transfer learning for regression. Sensors, 23(2), 583.