

Article

A Multi-Head Attention-Enhanced Fusion Model for Cross-Domain Short-Term Time Series Forecasting

Zhenyu Song ^{1,2,*}, Yunuo Zhang ^{1,3,*}, Zenan Lu ^{1,2}, Lixing Tan ^{1,2}, Chengfei Cai ^{1,2} and Cheng Tang ⁴¹ School of Artificial Intelligence, Taizhou University, Taizhou 225300, China² Jiangsu Key Laboratory of Intelligent Drug Screening and Repositioning, Taizhou 225300, China³ School of Artificial Intelligence and Computer Science, Jiangsu Normal University, Xuzhou 221116, China⁴ Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan

* Correspondence: songzhenyu@tzu.edu.cn (Z.S.); yunuo7799@gmail.com (Y.Z.)

† These authors contributed equally to this work.

Abstract

With the rapid advancement of artificial intelligence technologies in the era of big data, time series forecasting has become indispensable in critical fields such as environmental monitoring and financial market analysis. However, the existing forecasting models often encounter performance limitations when extracting high-dimensional features and generally cannot dynamically focus on critical information within long-term sequences. To address these challenges, this study proposes a multi-head attention fusion model (MAFM) designed to enhance the predictive accuracy and modelling capability for high-dimensional and nonlinear data across diverse application scenarios. Experiments were conducted on two heterogeneous datasets from the environmental and financial domains. After the key hyperparameters of the MAFM were optimized through an orthogonal experimental design, the model achieved coefficients of determination exceeding 0.90 on both datasets. Furthermore, the results of four comparative experiments demonstrate that the MAFM consistently outperforms traditional machine learning models, including support vector regression and extreme gradient boosting, as well as state-of-the-art deep learning models such as long short-term memory, temporal convolutional networks, and transformers. Compared with the best-performing baseline model on each sub-dataset, the MAFM reduced the mean squared error by 44.4%, 8.3%, 29.4%, and 65.5%, respectively, highlighting its superior predictive performance and strong generalization capability. In summary, the proposed MAFM provides an efficient, robust, and interpretable solution for time series forecasting tasks across multiple domains. Its outstanding performance demonstrates significant potential for practical applications in environmental monitoring, financial forecasting, and other real-world scenarios.



Academic Editor: Francesc Pozo

Received: 23 June 2026

Revised: 20 July 2026

Accepted: 22 July 2026

Published: 24 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: time series; multi-head attention; fusion model; prediction**MSC:** 68T07

1. Introduction

Short-term time series forecasting methods aim to uncover underlying patterns within historical data arranged in chronological order and subsequently predict future trends. With the rapid development of Internet of Things sensor technologies and financial information systems, large volumes of temporal data have been accumulated across various industries, providing an essential basis for data-driven decision-making [1]. Time series forecasting

methods have been widely applied in fields such as environmental monitoring [2,3] and financial market analysis [4,5] and have become critical technological tools for supporting intelligent decision-making [6]. For example, the accurate prediction of pollutant concentrations can provide early warnings for urban environmental governance [7], whereas reliable stock market forecasting serves as a foundation for risk management and investment decision-making [8]. Consequently, achieving high-precision time series forecasting has become a fundamental challenge in data science [9].

In the environmental domain, air pollutants such as fine particulate matter ($PM_{2.5}$) have become serious environmental and public health concerns. Owing to the extremely small particle size of $PM_{2.5}$, long-term exposure to $PM_{2.5}$ can lead to respiratory diseases such as asthma and bronchitis [10]. Therefore, accurate prediction of $PM_{2.5}$ concentrations is highly important for protecting human health. In the financial domain, stock price forecasting remains a highly challenging task with profound implications for investment management, risk assessment, and market efficiency theory [11]. Consequently, developing a forecasting model capable of effectively integrating spatiotemporal information and capturing complex multidimensional feature relationships has considerable academic and practical importance.

In recent years, the rapid advancement of deep learning has led to the emergence of numerous hybrid forecasting models. Although these models theoretically possess strong representation and learning capabilities, several challenges remain in their practical application [12]. First, when convolutional neural network (CNN)-based architectures are applied to time series forecasting, pooling operations, which were originally designed for image processing, are frequently adopted. Such downsampling operations may disrupt the temporal continuity of sequential data, thereby degrading forecasting performance [13,14]. Second, many existing models are designed and evaluated within specific application domains, limiting their generalizability across diverse datasets [15]. Furthermore, hybrid models often involve many hyperparameters, and their configuration still relies heavily on the empirical tuning of these hyperparameters, resulting in limited interpretability and a lack of systematic optimization strategies [16,17]. As a result, hybrid forecasting models generally suffer from high model complexity, substantial training costs, difficulties in temporal feature fusion, and stringent data quality and quantity requirements.

In addition, graph neural networks (GNNs) have attracted increasing attention in multivariate time series forecasting because they can explicitly model dependency relationships among variables through graph structures [18,19]. Unlike the conventional sequence models, which treat input variables independently, GNN-based methods can capture complex interactions among heterogeneous features, thereby improving the feature representation and forecasting accuracy. Representative methods include adaptive sparse graph attention networks [20] and multigraph attention networks [21,22], which have demonstrated promising performance in renewable energy and multivariate forecasting tasks. In addition, transformers capture dependencies between arbitrary positions in a sequence through multi-head self-attention, incorporate temporal order information by using positional encoding, and employ an encoder–decoder architecture to model the mapping between historical sequences and future short-term outputs for forecasting [23]. However, the computational complexity of self-attention increases quadratically with the sequence length, making the model prone to overfitting in short-term forecasting tasks with limited training samples. In addition, the transformer lacks an inherent prior for local temporal smoothness and is sensitive to high-frequency noise [24]. In summary, the graph learning mechanism in GNNs and the quadratic time and memory complexity of self-attention in transformers both increase the computational complexity and design difficulty of the model.

Motivated by these observations, in this study, an improved CNN architecture, a bidirectional long short-term memory (BiLSTM) network, and an attention mechanism are integrated to develop a multi-head attention fusion model (MAFM). The proposed model dynamically reconstructs the contributions of hidden representations across multiple feature subspaces, thereby enhancing the extraction of critical temporal information. The main contributions of this work are summarized as follows:

- A novel hybrid forecasting model, termed the MAFM, is proposed. By removing conventional pooling layers to preserve temporal continuity and incorporating a multi-head scaled dot-product self-attention mechanism, the ability of the model to capture critical features is significantly enhanced.
- An orthogonal experimental design is employed to optimize the model hyperparameters systematically. This strategy involves identifying the optimal parameter configurations for different datasets and ensures superior forecasting performance across various application scenarios.
- The proposed model is evaluated on representative environmental (PM_{2.5} concentration) and financial (Google stock price) forecasting datasets to verify its effectiveness across different application domains. The results of extensive experiments demonstrate that the MAFM consistently outperforms both traditional machine learning models and mainstream deep learning models.

The remainder of this paper is organized as follows: Section 2 reviews related work on time series forecasting methods. Section 3 presents the architecture of the proposed MAFM, including the design principles and computational processes of each component. Section 4 describes the datasets, preprocessing procedures, experimental settings, hyperparameter optimization process, and evaluation metrics, followed by a comprehensive analysis and discussion of the experimental results. Finally, Section 5 concludes the paper, discusses its limitations, and outlines future research directions.

2. Related Work

Time series forecasting has attracted considerable attention because of its extensive application value in environmental monitoring, financial analysis, energy management, and other domains. With the continuous advancement of research, forecasting approaches have evolved from traditional statistical methods to machine learning techniques and subsequently to deep learning and hybrid modelling frameworks. This section provides a concise review of these representative methodologies.

Early research on time series forecasting was dominated by statistical models, among which the autoregressive integrated moving average (ARIMA) model is among the most influential approaches [25–27]. The ARIMA model provided the theoretical foundation for modelling stationary time series data [28]. Subsequently, to address seasonal variations and conditional heteroscedasticity, researchers developed the seasonal ARIMA and generalized autoregressive conditional heteroscedasticity models [29,30], which have been widely applied in meteorological forecasting and financial volatility analysis. However, these statistical approaches are fundamentally based on linear assumptions and often struggle to achieve satisfactory performance in multivariate forecasting tasks characterized by complex nonlinear relationships and high-dimensional noisy data [31].

To overcome the limitations of traditional statistical models in nonlinear modelling, machine learning approaches have been introduced and have significantly improved forecasting performance through their nonlinear mapping capabilities [32,33]. In particular, support vector regression (SVR) [34,35] and ensemble learning methods such as extreme gradient boosting (XGBoost) have been widely adopted in time series forecasting [36]. For example, Jiang et al. improved gradient boosting trees using optimization algorithms

and demonstrated the strong generalizability of their model in multivariate time series modelling [37]. Similarly, Fan et al. proposed an SVR model and a simple tree-based XGBoost model for accurate prediction of daily global solar radiation using limited meteorological data [38]. Despite these advantages, machine learning methods still have several inherent limitations. They generally require extensive manual feature engineering, which depends substantially on domain expertise [39,40]. Furthermore, when dealing with long input sequences or high-dimensional feature spaces, machine learning models often struggle to effectively capture long-range temporal dependencies [41].

With the rapid development of deep learning technologies, deep learning-based forecasting methods have gradually become a central research focus because of their ability to automatically learn hierarchical feature representations through multiple nonlinear transformations [1,12,42]. Convolutional neural networks (CNNs) can efficiently extract local patterns and capture short-term dependencies in time series data [43,44]. Moreover, long short-term memory (LSTM) networks introduce gating mechanisms that effectively alleviate the vanishing gradient problem encountered in traditional recurrent neural networks, thereby exhibiting superior capability in modelling long-term temporal dependencies [45]. Although deep learning models have achieved remarkable success in nonlinear fitting and long-range dependency modelling, several limitations remain. Specifically, CNNs are constrained by the receptive field of the convolution kernels and focus primarily on local feature extraction, making it difficult to capture global dependencies. On the other hand, LSTM networks may still suffer from memory degradation when processing extremely long sequences [46]. Consequently, a single deep learning architecture often struggles to simultaneously model both local features and global temporal dependencies.

To address the limitations of individual deep learning models, researchers have increasingly explored hybrid architectures that combine the strengths of multiple techniques [47]. A common strategy is to integrate a CNN and an LSTM network, where the CNN is employed to extract local spatial features from input sequences and the LSTM network is applied for temporal dependency modelling [16]. Lu et al. applied a CNN–LSTM framework to stock market forecasting and verified the effectiveness of this hybrid architecture [48]. Nsengiyumva et al. employed a CNN–LSTM model for foreign exchange rate prediction and demonstrated that it outperformed standalone CNN and bidirectional LSTM (BiLSTM) models [49]. Although these hybrid architectures have achieved promising forecasting performance, attention mechanisms have been introduced to further enhance the identification of critical features from large-scale data. For instance, Romia and Huang applied a CNN–BiLSTM–Attention model to electric vehicle charging load forecasting [50]. Jiang et al. proposed a hybrid framework that integrates variational mode decomposition with the CNN–BiLSTM–Attention model for natural gas futures price prediction [51]. Furthermore, Zhang et al. developed a CNN–BiLSTM–Attention-based forecasting model for day-ahead wind speed prediction [52]. These studies demonstrate that incorporating attention mechanisms can effectively enhance feature representations and forecasting performance by allowing the model to dynamically focus on the most informative temporal patterns.

In addition to attention-based hybrid models, GNN-based prediction methods have recently emerged as important research directions for multivariate time series forecasting [18,19,53]. These methods model variables as graph nodes and explicitly learn their dependencies through graph messages passing or graph attention mechanisms. Compared with the CNN and RNN architectures, GNNs are particularly effective at capturing complex relationships among heterogeneous variables. However, graph construction typically relies on prior knowledge or adaptive graph learning, which increases model complexity and the computational cost. Therefore, the proposed MAFM focuses on enhanc-

ing temporal feature fusion without introducing graph structures, providing a lightweight and efficient solution for cross-domain time series forecasting.

3. Methodology

In this section, the fundamental methods and theories involved in this study are introduced, including the classic CNN, the BiLSTM network, and the detailed architecture of the proposed MAFM.

3.1. Convolutional Neural Network

A CNN typically consists of an input layer, convolutional layers, pooling layers, and fully connected layers. Among these components, convolutional layers are responsible for extracting local features through convolution kernels, pooling layers reduce feature dimensionality while preserving important information, and fully connected layers integrate the extracted features to generate prediction outputs. The architecture of a CNN is illustrated in Figure 1.

In this study, a one-dimensional convolution (Conv1D) is employed to process sequential data, enabling the model to capture the relationships between PM_{2.5} concentrations or stock prices and their associated multidimensional features. The one-dimensional convolution operation can be expressed as follows:

$$y_i^{(l)} = \sum_{k=1}^K w_k^{(l)} \cdot x_{(i+k-1)}^{(l-1)} + b^{(l)}, \quad (1)$$

where $y_i^{(l)}$ denotes the i -th element of the output sequence produced by the l -th convolutional layer, K represents the kernel size of the one-dimensional convolution filter, $w_k^{(l)}$ is the k -th weight of the convolution kernel in the l -th layer, $x_{(i+k-1)}^{(l-1)}$ denotes the corresponding input value within the receptive field from the $(l-1)$ -th layer, and $b^{(l)}$ is the bias term of the l -th convolutional layer.

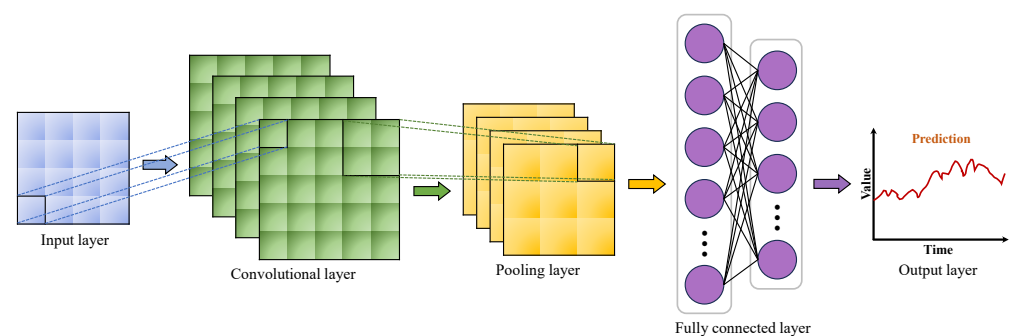


Figure 1. Architecture of the CNN model.

3.2. Bidirectional LSTM

The LSTM network consists of three primary gating mechanisms: the forget gate, input gate, and output gate. The architecture of the LSTM network is illustrated in Figure 2. The forget gate controls the extent to which historical information is retained or discarded by applying a sigmoid activation function to the input. The input gate is responsible for updating the memory cell by regulating the incorporation of new information through the combined action of the *Sigmoid* and *tanh* activation functions. The output gate determines the final output of the hidden state using both the sigmoid and tanh functions.

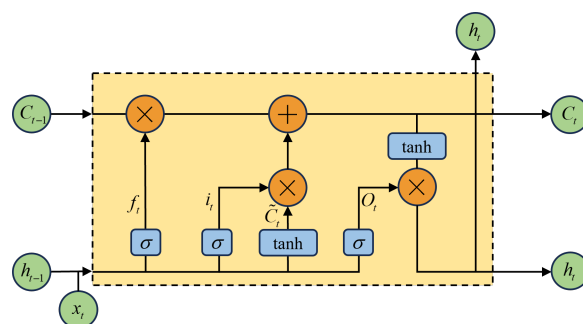


Figure 2. Architecture of the LSTM network.

The BiLSTM network is an enhanced version of the conventional LSTM. As shown in Figure 3, the BiLSTM network consists of a forwards LSTM layer and a backwards LSTM layer. By simultaneously exploiting past and future temporal information, the BiLSTM network can capture long-term dependencies in time series data more comprehensively. In the forwards LSTM layer, hidden states are computed sequentially from time step 1 to time step t . The update process at time step t is defined as follows:

$$f_t = \sigma(W_f[\vec{h}_{t-1}, x_t] + b_f), \quad (2)$$

$$i_t = \sigma(W_i[\vec{h}_{t-1}, x_t] + b_i), \quad (3)$$

$$\tilde{C}_t = \tanh(W_c[\vec{h}_{t-1}, x_t] + b_c), \quad (4)$$

$$C_t = f_t \otimes C_{t-1} \oplus i_t \otimes \tilde{C}_t, \quad (5)$$

$$O_t = \sigma(W_o[\vec{h}_{t-1}, x_t] + b_o), \quad (6)$$

$$\vec{h}_t = O_t \otimes \tanh(C_t), \quad (7)$$

where f_t denotes the activation value of the forget gate, x_t is the input vector at time step t , and $\vec{h}_{t-1}$ represents the hidden state at time step $t - 1$. The function $\sigma(\cdot)$ denotes the *sigmoid* activation function, while i_t represents the activation value of the input gate. $\tilde{C}_t$ denotes the candidate memory state, and $\tanh(\cdot)$ represents the hyperbolic tangent activation function. C_t is the updated cell state at time step t , and O_t denotes the activation value of the output gate. $[\cdot, \cdot]$ denotes the vector concatenation operation. W_f , W_i , W_c , and W_o are the weight matrices, while b_f , b_i , b_c , and b_o are the corresponding bias terms.

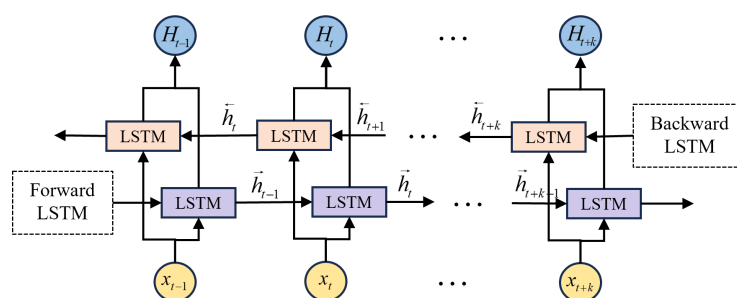


Figure 3. Architecture of the BiLSTM model.

In the backward LSTM layer, hidden states are computed in the reverse direction, from time step t to time step 1. Its internal gating mechanism follows the same formulation as that of Equations (2) and (7); however, an independent set of parameters is employed, and the sequence is processed in the opposite temporal direction. The hidden state of the backward LSTM layer can be expressed as follows:

$$\overleftarrow{h}_t = \text{LSTM}_{\text{backward}}(x_t, \overleftarrow{h}_{t+1}). \quad (8)$$

Finally, the hidden states generated by the forwards and backwards LSTM layers are concatenated to obtain the bidirectional temporal representation:

$$H_t = [\overrightarrow{h}_t, \overleftarrow{h}_t], \quad (9)$$

where H_t represents the fused hidden state containing information from both temporal directions.

3.3. Proposed MAFM

3.3.1. Attention Mechanism

The self-attention mechanism enables each hidden state in a sequence to attend not only to its own input but also to the hidden states at other positions within the sequence. By evaluating the relationships among different positions, the mechanism can capture rich contextual information and model long-range dependencies effectively. The architecture of the self-attention mechanism is illustrated in Figure 4.

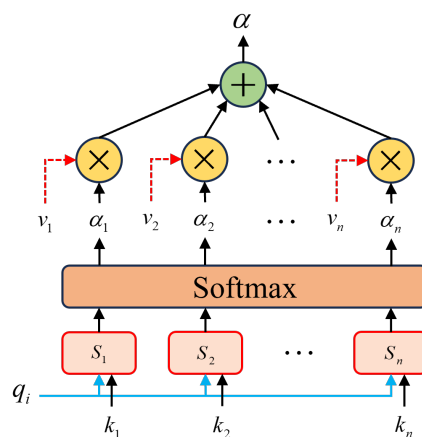


Figure 4. Architecture of the self-attention mechanism.

Given an input sequence H , the hidden representations are first projected to three vector spaces to obtain the query vector Q , key vector K , and value vector V , respectively:

$$Q = H \cdot W^Q, \quad (10)$$

$$K = H \cdot W^K, \quad (11)$$

$$V = H \cdot W^V, \quad (12)$$

where W^Q , W^K , and W^V denote the learnable projection matrices. To measure the correlations among different positions in the sequence, the Q vector at each position is multiplied by the $K^\top$ vectors of all positions. This operation produces an attention score matrix S , which reflects the degree of dependency between any two time steps:

$$S(Q, K) = Q \cdot K^\top. \quad (13)$$

As the dimensionality of K increases, the magnitude of the dot-product values may become excessively large, leading to unstable gradients during the softmax operation. To alleviate this issue, a scaling factor $1/\sqrt{d_k}$ is introduced, yielding the scaled attention score matrix:

$$S(Q, K) = \frac{QK^\top}{\sqrt{d_k}}, \quad (14)$$

where d_k denotes the dimensionality of K . The scaled attention scores are subsequently normalized using the *softmax* function to generate a weight distribution over all positions. The resulting attention weights are subsequently used to compute a weighted sum of the value vectors, thereby aggregating contextual information and producing the final attention output α . The corresponding formulations are given as follows:

$$\alpha_i = \text{Softmax}(s_i) = \frac{\exp(s_i)}{\sum_{j=1}^n \exp(s_j)}, \quad (15)$$

$$\text{Attention}(Q, K, V) = \sum_{i=1}^n \alpha_i \cdot v_i = \text{Softmax}(S) \cdot V, \quad (16)$$

where s_i denotes the scaled attention score at the i -th time step, n represents the sequence length, and α_i is the normalized attention weight satisfying $\sum_{i=1}^n \alpha_i = 1$ and $0 \leq \alpha_i \leq 1$.

To further enhance the model's representation capability, a multihead attention mechanism is adopted. Specifically, Q , K , and V are projected to multiple independent feature subspaces, where the attention operations are performed in parallel. This design enables the model to learn diverse temporal dependency patterns from different representation spaces. The output of the i -th attention head is defined as follows:

$$\text{head}_i = \text{Attention}(Q \cdot W_i^Q, K \cdot W_i^K, V \cdot W_i^V), \quad (17)$$

where W_i^Q , W_i^K , and W_i^V represent the weight matrices associated with the i -th attention head. Finally, the outputs of all the attention heads are concatenated and projected through a linear transformation to obtain the final multihead attention representation:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) \cdot W^O, \quad (18)$$

where W^O represents the output weight matrix and h denotes the number of attention heads.

3.3.2. MAFM Framework

The proposed MAFM achieves multilevel feature extraction for complex time series data by stacking different network structures. The model consists of a convolutional layer, a BiLSTM layer, an attention mechanism layer, and a fully connected layer. The overall architecture of the model is illustrated in Figure 5.

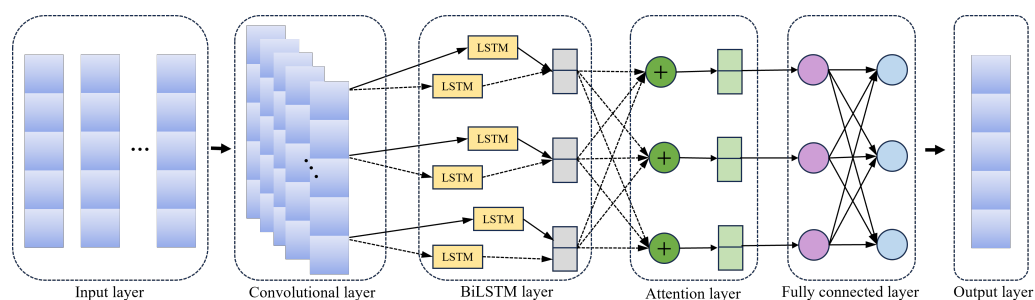


Figure 5. Framework of the MAFM for time series prediction.

Considering the characteristics of time series data in the environmental and financial domains, several modifications were made to the conventional CNN architecture. Specifically, the model retains only the one-dimensional convolutional (1D-CNN) layer, whereas the traditional pooling layer is removed. Since pooling is primarily used for dimensionality reduction and feature compression, this operation may disrupt temporal continuity and consequently affect the model's ability to learn detailed patterns and peak information. Therefore, in the MAFM, the convolutional layer is not intended for spatial sampling; in-

stead, it performs cross-channel linear combinations and mappings of the multidimensional input features at the same time step through pointwise convolution operations.

After local feature extraction is performed by the convolutional layer, the resulting sequence is fed into the BiLSTM layer for deep temporal modelling. The BiLSTM layer consists of two independent LSTM networks operating in the forward and backward directions to capture historical and future dependencies, respectively. Each LSTM unit uses the cell state C_t to store and selectively forget information under the control of the forget gate f_t , input gate i_t , and output gate O_t . Finally, the hidden states generated by the forward and backward LSTM networks are concatenated to form a feature vector H containing bidirectional temporal information.

However, when processing long time series, recurrent neural networks such as BiLSTM networks may suffer from memory degradation as the sequence length increases, making it difficult to capture critical historical information. To address this issue, a multi-head self-attention mechanism is introduced after the BiLSTM layer. The attention layer takes the output sequence of the BiLSTM layer H as the input and sequentially projects it to Q , K , and V , which are processed independently. The outputs of multiple attention heads are then concatenated and fused to obtain the attention-weighted sequence representation. This mechanism enables the model to capture long-range dependencies while focusing on important local features, thereby compensating for the limitations of both the BiLSTM model and the single attention strategy.

After local feature extraction is performed by the convolutional layer, temporal modelling is performed by the BiLSTM layer, and weighted aggregation is performed by the attention layer; the resulting features are fed into the fully connected layer to perform the final mapping and prediction task. The fully connected layer maps the abstract representations learned by the preceding layers to the output space of the corresponding forecasting task and generates the final prediction results.

4. Experimental Studies

4.1. Data Description and Processing

In this study, datasets from both environmental and financial domains were employed. The environmental dataset consists of PM_{2.5} pollution-related data collected in Beijing, China, and was obtained from the website (<https://www.aqistudy.cn/historydata> (accessed on 20 July 2026)). This dataset contains hourly air quality records from 00:00 on 1 January 2010, to 23:00 on 31 December 2014, comprising a total of 43,824 samples. It includes seven meteorological variables, such as the PM_{2.5} concentration, temperature, humidity, atmospheric pressure, and wind direction, with some missing values present in the original data. In this study, two continuous subsets without missing values were selected. In addition, the textual records of the wind direction attribute (cbwd) were encoded, as shown in Table 1.

Table 1. The encoding processing for the cbwd attribute.

Original Value		Encoded Value		
NE	1	0	0	0
NW	0	1	0	0
SE	0	0	1	0
cv	0	0	0	1

The financial dataset consists of Google stock data obtained from the Kaggle website (<https://www.kaggle.com/datasets> (accessed on 20 July 2026)). The dataset contains daily stock trading records from 14 June 2016, to 8 January 2020, comprising a total of 915 samples.

Each record includes multiple trading indicators, such as the date, opening price, highest price, lowest price, closing price, and trading volume. The dataset is complete and contains no missing values.

For the two datasets described above, four subsets, denoted as PM_{2.5} (D_1 and D_2) and Google stock (D_3 and D_4), were selected for subsequent experiments. To preserve the temporal ordering of the original time series, all of the datasets were divided chronologically into training and testing subsets at a ratio of 7:3. The earliest 70% of the observations were used for model training, while the remaining 30% were reserved for testing. No random shuffling was performed during data partitioning. Furthermore, sliding-window samples were generated independently within each subset to ensure that future observations were never used during model training, thereby effectively preventing information leakage. Additionally, to eliminate the influence of differences in the feature scales and reduce the computational complexity, all features were normalized using the following equation:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (19)$$

where x_{min} and x_{max} represent the minimum and maximum values of the corresponding feature among all samples, respectively, and x_{scaled} denotes the normalized value.

4.2. Evaluation Metrics

To comprehensively evaluate the performance of the compared models, we selected the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) as evaluation metrics. The MSE measures the discrepancy between the predicted values and actual values, where a smaller value indicates higher prediction accuracy. The RMSE provides a more intuitive interpretation of the prediction errors and facilitates error comparison. The MAE is less sensitive to outliers, and a lower MAE indicates a smaller average prediction error. R^2 is used to evaluate the goodness of fit of a regression model; a value closer to 1 indicates better fitting performance.

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2, \quad (20)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}, \quad (21)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}_i|, \quad (22)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad (23)$$

where x_i and $\hat{x}_i$ denote the actual value and predicted value of the i -th sample, respectively, $\bar{x}$ represents the mean of the actual values, and n denotes the total number of samples.

4.3. Experimental Setup and Parameter Settings

The experimental environment used for model training in this study is configured as follows. The operating system is Windows 11. The hardware platform is equipped with an Intel(R) Core(TM) i7-14700HX processor, an NVIDIA GeForce RTX 4060 laptop GPU, and 32 GB of memory. The software environment consists of Python 3.10.8, the PyTorch 2.6.0 deep learning framework, and the CUDA 12.6 computing platform. To ensure the statistical reliability of the experimental results, all the quantitative evaluation metrics reported in this study are averaged over 30 independent runs.

To systematically evaluate the predictive performance and structural advantages of the proposed MAFM, comparative analyses were conducted from three perspectives: comparisons with machine learning methods, comparisons with mainstream deep learning models, and comparisons of model variants through ablation studies. For the machine learning methods, SVR and XGBoost were selected as the baseline models. SVR is effective for modelling small-sample and nonlinear data, whereas XGBoost achieves excellent prediction accuracy and computational efficiency through its efficient ensemble learning mechanism. For the deep learning category, three representative time series forecasting models, namely, the LSTM network, temporal convolutional network (TCN), and transformer, were selected for comparison. The classic LSTM model was employed as a reference to evaluate the performance improvements of the proposed model, whereas the TCN and transformer served as advanced deep learning benchmarks.

Furthermore, to verify the effectiveness of the components incorporated in the MAFM, two ablation models, namely, CNN-BiLSTM and CNN-LSTM-Attention, were constructed. These experiments provide additional evidence regarding the contributions of the attention mechanism and the bidirectional recurrent architecture to the prediction performance and global information capture capability of the proposed model.

4.4. Parameter Settings

Owing to the large hyperparameter search space of hybrid models, an orthogonal experimental design was adopted to ensure that the MAFM achieves optimal forecasting performance across different application scenarios. The objective of the orthogonal experiment in this study was to efficiently identify an appropriate hyperparameter combination for the proposed MAFM rather than to investigate the statistical significance of individual factors. Therefore, only the three hyperparameters that have the greatest impact on the optimization convergence, feature representation capability, and attention modelling, namely, the learning rate (L_r), hidden dimension (H_d), and number of attention heads (A_h), were included in the orthogonal design. The other architectural parameters were fixed according to the model design principles and preliminary exploratory experiments to ensure fair comparisons and to avoid unnecessary increases in computational complexity. Subset D_1 of the environmental dataset and subset D_3 of the financial dataset were selected for the orthogonal experiments, which were conducted using the $L_{16}(4^3)$ orthogonal array. The results are presented in Tables 2 and 3, where the optimal parameter combinations are highlighted in bold, corresponding to the 5th and 11th settings, respectively. The remaining parameters of the MAFM were configured as follows. The number of input channels was set equal to the input feature dimension (5 or 11), while the number of output channels was fixed at 32. The convolution kernel size was set to 1, and the number of BiLSTM layers was set to 1. During training, the input dimension was set to 5 or 11 according to the dataset, and the output dimension was set to 1. The batch size was 32, the sliding window length (*time step*) was 10, and the number of training epochs was 100. The learning rate was set to 0.0001 or 0.0002 depending on the dataset.

The models included in the comparative experiments were the SVR, XGBoost, LSTM, TCN, and transformer models. Their parameter settings were adopted from the corresponding literature, while the shared parameters were kept as consistent as possible with those used in the MAFM. The detailed parameter configurations are listed in Table 4.

Table 2. Experimental results of the proposed MAFM on D_1 for different parameter combinations.

No.	L_r	H_d	A_h	MSE	RMSE	MAE	R^2
1	0.00005	32	2	0.0017	0.0412	0.0296	0.6926
2	0.00005	64	4	0.0011	0.0332	0.0235	0.7944
3	0.00005	128	8	0.0017	0.0412	0.0296	0.6905
4	0.00005	256	16	0.0014	0.0374	0.0263	0.745
5	0.0001	32	4	0.0005	0.0224	0.0145	0.9068
6	0.0001	64	2	0.0008	0.0283	0.0192	0.8505
7	0.0001	128	16	0.0016	0.04	0.0281	0.7151
8	0.0001	256	8	0.0014	0.0374	0.0262	0.7506
9	0.0002	32	8	0.0016	0.04	0.0246	0.7178
10	0.0002	64	16	0.0013	0.0361	0.0257	0.765
11	0.0002	128	2	0.001	0.0316	0.0235	0.8096
12	0.0002	256	4	0.0018	0.0424	0.0297	0.6715
13	0.0004	32	16	0.0009	0.03	0.0192	0.8449
14	0.0004	64	8	0.0013	0.0361	0.0237	0.7626
15	0.0004	128	4	0.0029	0.0539	0.0287	0.4752
16	0.0004	256	2	0.0042	0.0648	0.046	0.2266

Table 3. Experimental results of the proposed MAFM on D_3 for different parameter combinations.

No.	L_r	H_d	A_h	MSE	RMSE	MAE	R^2
1	0.00005	32	2	0.0035	0.0592	0.0479	0.7243
2	0.00005	64	4	0.0031	0.0557	0.0447	0.7561
3	0.00005	128	8	0.0029	0.0539	0.0431	0.7719
4	0.00005	256	16	0.0025	0.05	0.0398	0.8012
5	0.0001	32	4	0.0032	0.0566	0.0464	0.7453
6	0.0001	64	2	0.0021	0.0458	0.0359	0.8308
7	0.0001	128	16	0.0021	0.0458	0.0354	0.8354
8	0.0001	256	8	0.0019	0.0436	0.0334	0.8483
9	0.0002	32	8	0.0021	0.0458	0.0357	0.8353
10	0.0002	64	16	0.0019	0.0436	0.0331	0.8493
11	0.0002	128	2	0.0012	0.0346	0.0264	0.9057
12	0.0002	256	4	0.0024	0.049	0.0385	0.8108
13	0.0004	32	16	0.002	0.0447	0.0345	0.843
14	0.0004	64	8	0.0015	0.0387	0.0295	0.8778
15	0.0004	128	4	0.0029	0.0539	0.0451	0.7678
16	0.0004	256	2	0.0019	0.0436	0.0342	0.848

Table 4. Parameter settings of the comparison models.

Model	Parameters or Conditions
SVR	$kernel = RBF, cost(c) = 10, \gamma = 0.1$
XGBoost	$learning\ rate = 0.3$
LSTM	$hidden\ layers = 128$
TCN	$kernel\ size = 2, layers = 3, heads = 4$
Transformer	$channels \in [64, 64, 64], Hidden\ layers = 64$ $heads = 4, encoding\ layers = 1,$ $dimension = 128$

4.5. Performance Comparison

4.5.1. Environmental Data Forecasting Results

In the $PM_{2.5}$ concentration forecasting task, the proposed MAFM achieved the best performance on both environmental datasets, D_1 and D_2 (Table 5). On dataset D_1 , the MSE of the MAFM was reduced by 44.4% compared with that of the best-performing baseline model, the LSTM network, while the R^2 value of the MAFM reached 0.9068. On dataset D_2 ,

the MSE of the MAFM was reduced by 8.3% compared with that of the best baseline model, with an R^2 value of 0.881. These results are significantly better than those of traditional machine learning models, such as SVR and XGBoost, as well as those of mainstream deep learning models, including the LSTM, TCN, and transformer models. Furthermore, the convergence curves in Figure 6 and the prediction fitting curves in Figure 7 indicate that compared with the other models, the MAFM converges faster during training with smaller loss fluctuations. Moreover, the MAFM can fit extreme values to a certain extent. In other words, compared with the comparison models, the MAFM achieves higher fitting accuracy in the peak, trough, and sharply fluctuating regions of the $PM_{2.5}$ concentration. This improvement is directly related to the architecture of the proposed model. The multi-head self-attention-enhanced fusion mechanism effectively addresses the limitations of existing models in high-dimensional feature extraction and enables dynamic adaptation to critical information, such as peaks and troughs in long sequences, demonstrating its applicability in complex environmental monitoring scenarios.

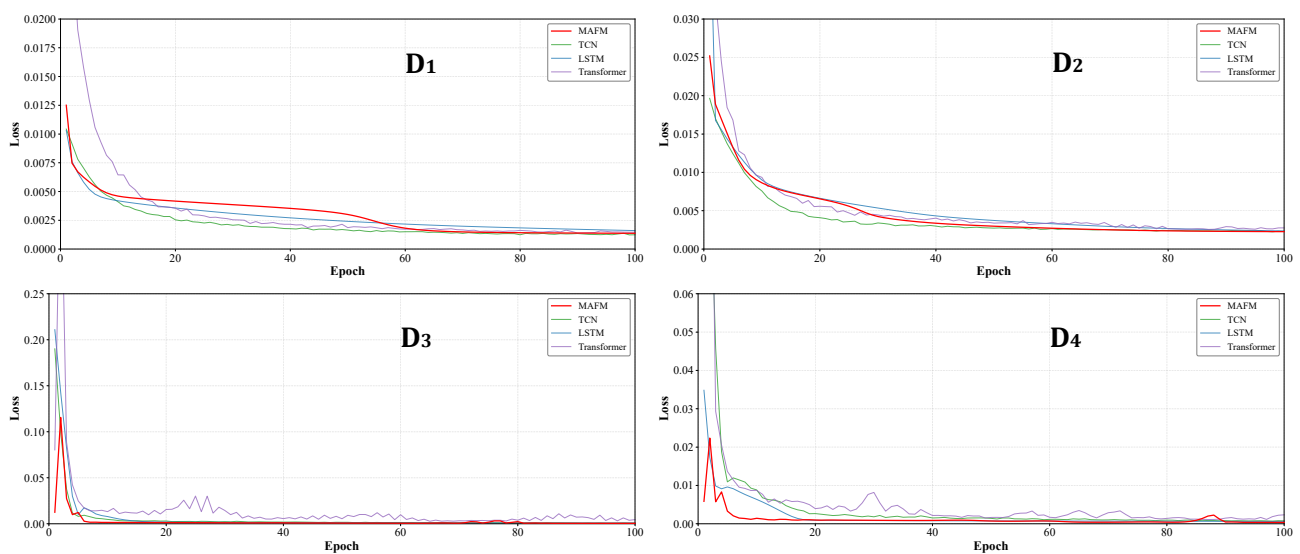


Figure 6. Convergence curves of the four deep learning models.

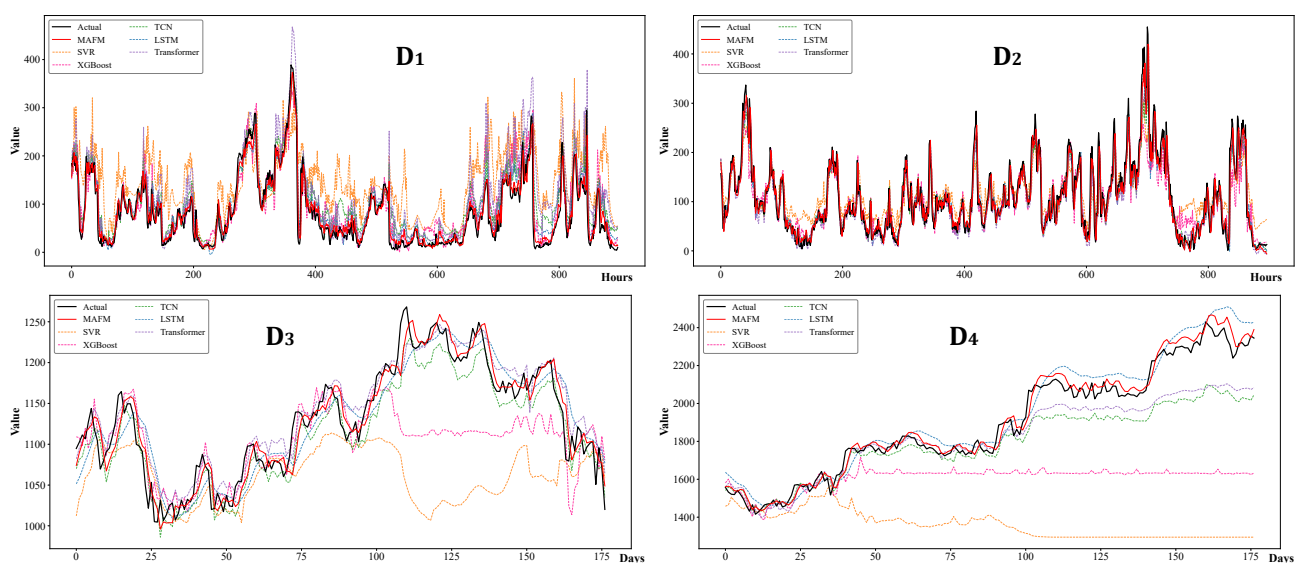


Figure 7. Time series prediction results obtained by the all comparison models.

Table 5. Experimental results produced by all the competitive models in the PM_{2.5} concentration forecasting task.

Model	D_1				D_2			
	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2
SVR	0.0073	0.0854	0.0708	0.3373	0.0087	0.0933	0.0732	0.6847
XGBoost	0.0011	0.0332	0.0231	0.7998	0.0062	0.0787	0.0514	0.7756
LSTM	0.0009	0.03	0.0221	0.834	0.0037	0.0608	0.042	0.8647
TCN	0.0013	0.0361	0.0277	0.7644	0.0036	0.06	0.0405	0.8696
Transformer	0.0026	0.051	0.0357	0.5243	0.0037	0.0608	0.0432	0.867
MAFM	0.0005	0.0224	0.0145	0.9068	0.0033	0.0574	0.0398	0.881

4.5.2. Financial Data Forecasting Results

For the Google stock price forecasting task, Figure 7 shows that D_3 exhibits relatively stable price fluctuations with a smoother overall trend, whereas D_4 contains more pronounced upward and downward movements as well as trend reversal points, which makes it more challenging to predict. The results in Table 6 highlight the superiority of the MAFM. On datasets D_3 and D_4 , the MSE values of the MAFM were reduced by 29.4% and 65.5%, respectively, compared with those of the best-performing baseline model. In particular, the R^2 value of the MAFM on dataset D_4 reached 0.9758. The traditional machine learning models performed poorly on such highly volatile time series data, whereas the baseline deep learning models such as the LSTM network and TCN exhibited notable trend lag and turning-point deviations. The visualizations in Figures 6 and 7 show that the MAFM converges steadily during training without obvious oscillations and accurately captures both long-term increasing trends and short-term fluctuations in stock prices. This superior performance across different scenarios stems from the fusion mechanism of the MAFM, which combines the global dependency modelling capability of the self-attention mechanism with the local feature extraction capability of the temporal convolution. As a result, the proposed model preserves global contextual information in long sequences while increasing its sensitivity to short-term abnormal fluctuations. The consistent performance gains achieved in both the environmental and financial forecasting tasks demonstrate the effectiveness and strong generalizability of the proposed model, which provides an efficient and robust solution for time series forecasting across multiple domains.

Table 6. Experimental results produced by all the competitive models in the Google stock price prediction task.

Model	D_3				D_4			
	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2
SVR	0.0268	0.1637	0.1208	0.1197	0.0209	0.4572	0.3856	0.5173
XGBoost	0.0090	0.0949	0.0692	0.2905	0.0751	0.2740	0.2114	0.3838
LSTM	0.0025	0.05	0.0388	0.8052	0.0029	0.0539	0.0443	0.9292
TCN	0.0017	0.0412	0.0349	0.8618	0.0116	0.1077	0.0787	0.717
Transformer	0.0018	0.0424	0.0322	0.8604	0.0081	0.09	0.0642	0.8025
MAFM	0.0012	0.0346	0.0264	0.9057	0.001	0.0316	0.0252	0.9758

4.6. Ablation Experiment

To verify the effectiveness of the multi-head self-attention-enhanced fusion architecture of the proposed MAFM, three representative hybrid time series models were selected as variants for comparative experiments: BiLSTM+Attention, CNN+BiLSTM, and CNN+LSTM+Attention. All the variant models employed the same optimal hyperparameters, training strategy, and dataset partitioning scheme as the MAFM did. Experiments

were conducted on datasets D_1 and D_3 , and the results are presented in Tables 7 and 8. The corresponding prediction fitting curves are shown in Figures 8 and 9.

The experimental results indicate that the MAFM consistently outperforms all the hybrid variant models on both heterogeneous datasets. In the $PM_{2.5}$ forecasting task, the MAFM achieved an MSE of 0.0005 and an R^2 value of 0.9068. Compared with the best-performing variant, CNN+BiLSTM, the MSE of the complete MAFM was reduced by 28.6%, and the fitting accuracy was substantially increased. In contrast, the performance of the CNN+LSTM+Attention variant decreased significantly, with an R^2 value of only 0.5638, indicating that a simple serial stacking structure may lead to information loss during feature transmission, while the attention mechanism at the output stage cannot effectively utilize the local and temporal features extracted by the preceding layers. In the Google stock price forecasting task, the MAFM also demonstrated superior performance, achieving an MSE of 0.0012 and an R^2 value of 0.9057. Compared with the best-performing variant, CNN+BiLSTM, the MSE of the complete MAFM was reduced by 29.4%, demonstrating the adaptability of the proposed fusion architecture to highly volatile financial time series data.

Further analysis of the prediction fitting curves reveals that the CNN+BiLSTM variant notably underestimates extreme $PM_{2.5}$ peaks, whereas the BiLSTM+Attention variant produces relatively large deviations in regions with continuous fluctuations (Figure 8). Furthermore, the prediction results of the CNN+LSTM+Attention variant differ substantially from the actual sequence. In contrast, the prediction curve of the complete MAFM closely matches the ground truth, particularly at the abrupt change points at approximately 400 and 800 h, where its fitting accuracy is markedly higher than that of the other variants. Furthermore, as shown in Figure 9, near the stock price peak around day 125 and the trough around day 175, all the hybrid variants exhibit varying degrees of prediction lag and amplitude deviation, whereas the complete MAFM accurately captures both the trend turning points and fluctuation magnitudes.

Table 7. Experimental results produced by different hybrid schemes in the $PM_{2.5}$ concentration forecasting task.

Hybrid Scheme	MSE	RMSE	MAE	R^2
BiLSTM+Attention	0.0012	0.0346	0.024	0.779
CNN+BiLSTM	0.0007	0.0265	0.017	0.878
CNN+LSTM+Attention	0.0024	0.049	0.0341	0.5638
MAFM	0.0005	0.0224	0.0145	0.9068

Table 8. Experimental results produced by different hybrid schemes in the Google stock price prediction task.

Hybrid Scheme	MSE	RMSE	MAE	R^2
BiLSTM+Attention	0.0035	0.0592	0.0494	0.7192
CNN+BiLSTM	0.017	0.0412	0.0319	0.8632
CNN+LSTM+Attention	0.0028	0.0529	0.0407	0.7754
MAFM	0.0012	0.0346	0.0264	0.9057

The results of the ablation experiments further support the rationale for the proposed model design. Neither the model with the simple serial fusion of local and temporal features nor the model with the straightforward addition of an attention mechanism at the output stage can simultaneously and accurately extract local high-dimensional features and effectively model long-range dependencies. Owing to its multi-head attention-enhanced fusion architecture, the proposed MAFM can extract temporal features at different scales in parallel, and a multi-head self-attention mechanism is employed for dynamic weighting and

global feature fusion. This design effectively alleviates the information loss associated with serial architectures and addresses the limited focus of conventional attention mechanisms on critical local features, thereby explaining the superior performance of the MAFM and further demonstrating the rationality and necessity of the proposed model architecture.

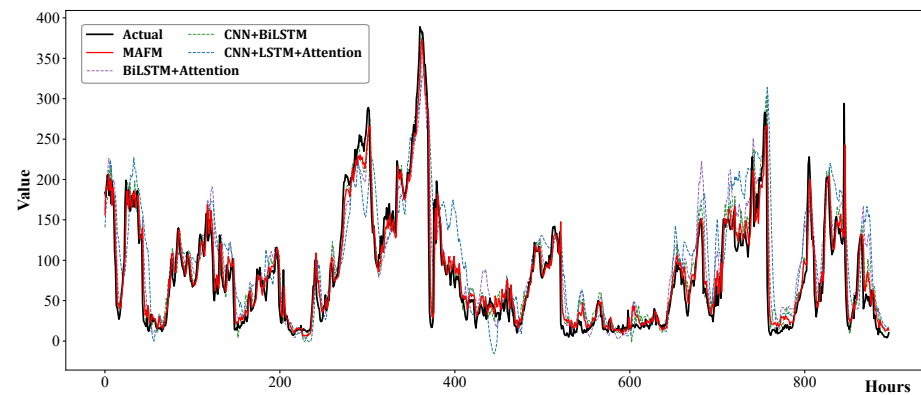


Figure 8. PM_{2.5} concentration forecasting results of the different hybrid schemes.

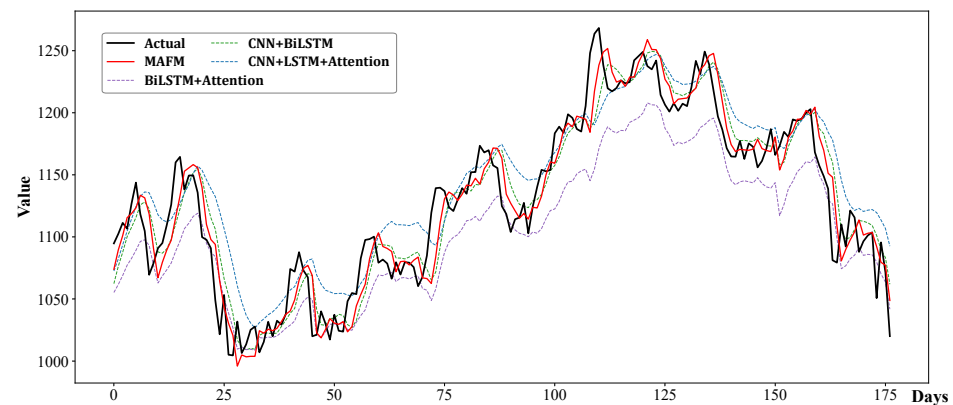


Figure 9. Google stock price prediction results of the different hybrid schemes.

5. Conclusions

To address the challenges of high-dimensional feature extraction, limited prediction accuracy, and insufficient long-sequence dependency modelling in short-term time series forecasting, this study presents a multi-head attention fusion model (MAFM). In the proposed model, the pooling layer is removed from the traditional CNN architecture, and pointwise convolutions are employed for feature fusion to extract local temporal features. Additionally, a BiLSTM network is utilized to capture bidirectional dependencies from both past and future contexts, while a multi-head attention mechanism is introduced to perform weighted aggregation of relevant information. To evaluate the effectiveness of the proposed model, comprehensive experiments were conducted with two forecasting tasks: Beijing PM_{2.5} concentration prediction and Google stock price prediction. The results demonstrate that the MAFM consistently outperforms traditional machine learning models and mainstream deep learning models across all four subsets. Furthermore, the ablation studies confirm the contributions of the CNN, BiLSTM, and attention modules to the overall model performance. Compared with the transformer-based models that stack multiple encoder layers, the proposed MAFM adopts a relatively lightweight architecture consisting of a single 1D convolution layer, one BiLSTM layer, and one multi-head attention module. Therefore, although the attention mechanism introduces additional computational overhead compared with a standalone BiLSTM, the overall computational complexity remains moderate and suitable for short-term time series forecasting tasks.

Despite these promising results, the MAFM still has several limitations. As the input dimensionality increases, the number of model parameters increases substantially, resulting in higher computational costs and reduced efficiency. In addition, the MAFM operates as an end-to-end forecasting model and provides limited interpretability. Moreover, from the perspective of dataset diversity, the proposed model is evaluated on only two types of time series datasets. Therefore, the current experimental results are not sufficient in terms of fully demonstrating the generalization capability of the MAFM. In future work, we will investigate GNNs and adaptive graph learning techniques to integrate temporal modelling with graph-based feature interaction learning. Such an extension is expected to better capture complex relationships among heterogeneous variables while further improving the generalization capability of the proposed framework and focusing on evaluating its generalization capability and robustness in a wider range of application scenarios and on more representative public datasets.

Author Contributions: Conceptualization, Z.S. and Y.Z.; methodology, Z.S. and Y.Z.; resources, Y.Z. and Z.L.; software, Z.S.; writing—original draft, Z.S.; writing—review and editing, Z.S., Y.Z. and C.T.; visualization, Y.Z.; validation, Y.Z. and Z.L.; formal analysis, L.T., Y.Z. and C.C.; investigation, Z.L.; data curation, C.C.; project administration, Z.S.; supervision, Y.Z. and L.T.; funding acquisition, Z.S. and C.T. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partially supported by the Talent Development Project of Taizhou University (No. TZXY2018QDJJ006), the Young Science and Technology Talent Support Project of Taizhou, and the National Natural Science Foundation of China (Grant No. 62473225).

Informed Consent Statement: Informed consent was obtained from all the subjects involved in the study.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Lim, B.; Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2021**, *379*, 20200209. [\[CrossRef\]](#)
2. Ceravolo, R.; Coletta, G.; Miraglia, G.; Palma, F. Statistical correlation between environmental time series and data from long-term monitoring of buildings. *Mech. Syst. Signal Process.* **2021**, *152*, 107460. [\[CrossRef\]](#)
3. Yang, H.; Yu, J.; Xu, W.; Wu, Y.; Lei, X.; Ye, J.; Geng, J.; Ding, Z. Long-time series ecological environment quality monitoring and cause analysis in the Dianchi Lake Basin, China. *Ecol. Indic.* **2023**, *148*, 110084. [\[CrossRef\]](#)
4. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [\[CrossRef\]](#)
5. Tang, Y.; Song, Z.; Zhu, Y.; Yuan, H.; Hou, M.; Ji, J.; Tang, C.; Li, J. A survey on machine learning models for financial time series forecasting. *Neurocomputing* **2022**, *512*, 363–380. [\[CrossRef\]](#)
6. Benidis, K.; Rangapuram, S.S.; Flunkert, V.; Wang, Y.; Maddix, D.; Turkmen, C.; Gasthaus, J.; Bohlke-Schneider, M.; Salinas, D.; Stella, L.; et al. Deep learning for time series forecasting: Tutorial and literature survey. *Acm Comput. Surv.* **2022**, *55*, 1–36. [\[CrossRef\]](#)
7. Das, R.; Middya, A.I.; Roy, S. High granular and short term time series forecasting of PM_{2.5} air pollutant—a comparative review. *Artif. Intell. Rev.* **2022**, *55*, 1253–1287.
8. Zhou, T.; Gao, S.; Wang, J.; Chu, C.; Todo, Y.; Tang, Z. Financial time series prediction using a dendritic neuron model. *Knowl.-Based Syst.* **2016**, *105*, 214–224. [\[CrossRef\]](#)
9. Song, Z.; Song, S.; Song, S.; Tan, L.; Tang, C.; Ji, J. Evaluating a novel incremental-input neural network for multivariate air temperature forecasting. *Eng. Appl. Artif. Intell.* **2026**, *170*, 114203. [\[CrossRef\]](#)
10. Wu, L.Y.; He, W.T.; Zeeshan, M.; Zhou, Y.; Zhang, Y.T.; Liang, L.X.; Huang, J.W.; Zhou, J.X.; Zhao, K.; Bao, W.W.; et al. Incidence of respiratory diseases associated with per-and polyfluoroalkyl substances (PFAS) in PM_{2.5}: New evidence from a population-based survey of Pearl River Delta (PRD), China. *J. Hazard. Mater.* **2025**, *494*, 138485. [\[CrossRef\]](#) [\[PubMed\]](#)

11. Gürkaynak, R.; Karasoy-Can, H.G.; Lee, S.S. Stock market's assessment of monetary policy transmission: The cash flow effect. *J. Financ.* **2022**, *77*, 2375–2421. [[CrossRef](#)]
12. Kim, J.; Kim, H.; Kim, H.; Lee, D.; Yoon, S. A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges. *Artif. Intell. Rev.* **2025**, *58*, 216. [[CrossRef](#)]
13. Lin, C.; Weng, K.; Lin, Y.; Zhang, T.; He, Q.; Su, Y. Time series prediction of dam deformation using a hybrid STL–CNN–GRU model based on sparrow search algorithm optimization. *Appl. Sci.* **2022**, *12*, 11951. [[CrossRef](#)]
14. Zhang, L.; Cai, Y.; Huang, H.; Li, A.; Yang, L.; Zhou, C. A CNN–LSTM model for soil organic carbon content prediction with long time series of MODIS-based phenological variables. *Remote Sens.* **2022**, *14*, 4441. [[CrossRef](#)]
15. Kirisci, M.; Cagcag Yolcu, O. A new CNN-based model for financial time series: TAIEX and FTSE stocks forecasting. *Neural Process. Lett.* **2022**, *54*, 3357–3374. [[CrossRef](#)]
16. Livieris, I.E.; Pintelas, E.; Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. *Neural Comput. Appl.* **2020**, *32*, 17351–17360. [[CrossRef](#)]
17. Tian, G.; Xu, Y.; Ma, X.; Li, X.; Zhang, C. MDWConv: CNN based on multi-scale atrous pyramid and depthwise separable convolution for long time series forecasting. *Neural Netw.* **2025**, *185*, 107139. [[CrossRef](#)] [[PubMed](#)]
18. Corradini, F.; Gerosa, F.; Gori, M.; Lucheroni, C.; Piangerelli, M.; Zannotti, M. A systematic literature review of spatio-temporal graph neural network models for time series forecasting and classification. *Neural Netw.* **2025**, *195*, 108269. [[CrossRef](#)] [[PubMed](#)]
19. Hou, M.; Liu, Z.; Sa, G.; Wang, Y.; Sun, J.; Li, Z.; Tan, J. Parallel multi-scale dynamic graph neural network for multivariate time series forecasting. *Pattern Recognit.* **2025**, *158*, 111037. [[CrossRef](#)]
20. Qu, K.; Xue, S.; Zheng, X.; Yan, D.; Cao, H. Learning dynamic inter-farm dependencies for wind power forecasting via adaptive sparse graph attention network. *Renew. Energy* **2025**, *258*, 124969.
21. Wang, Y.; Yang, Z.; Ma, J.; Jin, Q. A wind speed forecasting framework for multiple turbines based on adaptive gate mechanism enhanced multi-graph attention networks. *Appl. Energy* **2024**, *372*, 123777. [[CrossRef](#)]
22. Huang, H.; Chen, Y.; Lin, C.; Qin, Y. A multi-graph attention network for joint information extraction. *Expert Syst. Appl.* **2025**, *297*, 129350. [[CrossRef](#)]
23. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Washington, DC, USA, 2023; Volume 37, pp. 11121–11128.
24. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. itransformer: Inverted transformers are effective for time series forecasting. In *Proceedings of the International Conference on Learning Representations*; OpenReview.net: Amherst, MA, USA, 2024; Volume 2024, pp. 11116–11140.
25. Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [[CrossRef](#)]
26. Contreras, J.; Espinola, R.; Nogales, F.J.; Conejo, A.J. ARIMA models to predict next-day electricity prices. *IEEE Trans. Power Syst.* **2003**, *18*, 1014–1020. [[CrossRef](#)]
27. Siarni-Namini, S.; Tavakoli, N.; Namin, A.S. A comparison of ARIMA and LSTM in forecasting time series. In *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1394–1401.
28. Liu, C.; Hoi, S.C.; Zhao, P.; Sun, J. Online arima algorithms for time series prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Washington, DC, USA, 2016; Volume 30.
29. Dimri, T.; Ahmad, S.; Sharif, M. Time series analysis of climate variables using seasonal ARIMA approach. *J. Earth Syst. Sci.* **2020**, *129*, 149. [[CrossRef](#)]
30. Modarres, R.; Ouarda, T.B. Generalized autoregressive conditional heteroscedasticity modelling of hydrologic time series. *Hydrol. Process.* **2013**, *27*, 3174–3191.
31. Kontopoulou, V.I.; Panagopoulos, A.D.; Kakkos, I.; Matsopoulos, G.K. A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet* **2023**, *15*, 255. [[CrossRef](#)]
32. Herzen, J.; Lässig, F.; Piazzetta, S.G.; Neuer, T.; Tafti, L.; Raille, G.; Van Pottelbergh, T.; Pasieka, M.; Skrodzki, A.; Huguenin, N.; et al. Darts: User-friendly modern machine learning for time series. *J. Mach. Learn. Res.* **2022**, *23*, 1–6.
33. Liu, G.; Zhong, K.; Li, H.; Chen, T.; Wang, Y. A state of art review on time series forecasting with machine learning for environmental parameters in agricultural greenhouses. *Inf. Process. Agric.* **2024**, *11*, 143–162. [[CrossRef](#)]
34. Sapankevych, N.I.; Sankar, R. Time series prediction using support vector machines: A survey. *IEEE Comput. Intell. Mag.* **2009**, *4*, 24–38. [[CrossRef](#)]
35. Lin, G.; Lin, A.; Gu, D. Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient. *Inf. Sci.* **2022**, *608*, 517–531. [[CrossRef](#)]
36. Zhang, Y.; Shi, X.; Zhang, S.; Abraham, A. A xgboost-based lane change prediction on time series data using feature engineering for autopilot vehicles. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 19187–19200. [[CrossRef](#)]

37. Jiang, W.; Lin, P.; Liang, Y.; Gao, H.; Zhang, D.; Hu, G. A novel hybrid deep learning model for multi-step wind speed forecasting considering pairwise dependencies among multiple atmospheric variables. *Energy* **2023**, *285*, 129408. [\[CrossRef\]](#)
38. Fan, J.; Wang, X.; Wu, L.; Zhou, H.; Zhang, F.; Yu, X.; Lu, X.; Xiang, Y. Comparison of Support Vector Machine and Extreme Gradient Boosting for predicting daily global solar radiation using temperature and precipitation in humid subtropical climates: A case study in China. *Energy Convers. Manag.* **2018**, *164*, 102–111. [\[CrossRef\]](#)
39. Li, G.; Jung, J.J. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Inf. Fusion* **2023**, *91*, 93–102. [\[CrossRef\]](#)
40. Barrera-Animas, A.Y.; Oyedele, L.O.; Bilal, M.; Akinosho, T.D.; Delgado, J.M.D.; Akanbi, L.A. Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Mach. Learn. Appl.* **2022**, *7*, 100204. [\[CrossRef\]](#)
41. Hall, T.; Rasheed, K. A survey of machine learning methods for time series prediction. *Appl. Sci.* **2025**, *15*, 5957. [\[CrossRef\]](#)
42. Torres, J.F.; Hadjout, D.; Sebaa, A.; Martínez-Álvarez, F.; Troncoso, A. Deep learning for time series forecasting: A survey. *Big Data* **2021**, *9*, 3–21. [\[CrossRef\]](#) [\[PubMed\]](#)
43. Izadi, M.A.; Hajizadeh, E. Time series prediction for cryptocurrency markets with transformer and parallel convolutional neural networks. *Appl. Soft Comput.* **2025**, *177*, 113229. [\[CrossRef\]](#)
44. Özdoğan, İ.; Yıldız, O.; Boran, F.E. A hybrid fuzzy-CNN model with feature extraction from MACD-based trends for stock market movement forecasting. *Expert Syst. Appl.* **2025**, *296*, 129068.
45. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. In *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*; IEEE: Piscataway, NJ, USA, 2019; pp. 3285–3292.
46. Li, Y.; Zhu, Z.; Kong, D.; Han, H.; Zhao, Y. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowl.-Based Syst.* **2019**, *181*, 104785. [\[CrossRef\]](#)
47. Nikseresht, A. Dynamic hydrogen demand forecasting using hybrid time series models: Insights for renewable energy systems. *Renew. Energy* **2025**, *244*, 122737. [\[CrossRef\]](#)
48. Lu, W.; Li, J.; Li, Y.; Sun, A.; Wang, J. A CNN-LSTM-based model to forecast stock prices. *Complexity* **2020**, *2020*, 6622927. [\[CrossRef\]](#)
49. Nsengiyumva, E.; Mung'atu, J.K.; Ruranga, C. A comparative study of multivariate CNN, BiLSTM and hybrid CNN–BiLSTM models for forecasting foreign exchange rate using deep learning. *Cogent Econ. Financ.* **2025**, *13*, 2526148. [\[CrossRef\]](#)
50. Romia, A.; Huang, Q. Attention-Enhanced CNN-LSTM Models for Forecasting EV Fast-Charging Load at Public Stations. *IEEE Trans. Ind. Appl.* **2026**, 1–14. [\[CrossRef\]](#)
51. Jiang, Q.; Lin, Z.; Hu, J.; Liu, X. Application of the VMD-CNN-BiLSTM-Attention Model in Daily Price Forecasting of NYMEX Natural Gas Futures. *Appl. Sci.* **2025**, *15*, 11169. [\[CrossRef\]](#)
52. Zhang, Y.M.; Wang, H. Multi-head attention-based probabilistic CNN-BiLSTM for day-ahead wind speed forecasting. *Energy* **2023**, *278*, 127865. [\[CrossRef\]](#)
53. Jin, X.B.; Ma, H.; Xie, J.Y.; Kong, J.; Deveci, M.; Kadry, S. Ada-STGMAT: An adaptive spatio-temporal graph multi-attention network for intelligent time series forecasting in smart cities. *Expert Syst. Appl.* **2025**, *269*, 126428. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.