agriculture

MDPI

Article

AMSformer: A Transformer for Grain Storage Temperature Prediction Using Adaptive Multi-Scale Feature Fusion

Qinghui Zhang [1], Weixiang Zhang [1], Quanzhen Huang [2,*], Chenxia Wan [1] and Zhihui Li [1]

[1] College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China; zqh131@163.com (Q.Z.); zwx19980222@163.com (W.Z.); wanchenxia@haut.edu.cn (C.W.); lizhihui@haut.edu.cn (Z.L.)
[2] School of Electrical and Information Engineering, Henan University of Engineering, Zhengzhou 451191, China
* Correspondence: huangquanzhen666@126.com; Tel.: +86-15890181395

Abstract: Grain storage temperature prediction is crucial for silo safety and can effectively prevent mold and mildew caused by increasing grain temperature and condensation due to decreasing grain temperature. However, current prediction methods lead to information redundancy when capturing temporal and spatial dependencies, which diminishes prediction accuracy. To tackle this issue, this paper introduces an adaptive multi-scale feature fusion transformer model (AMSformer). Firstly, the model utilizes the adaptive channel attention (ACA) mechanism to adjust the weights of different channels according to the input data characteristics and suppress irrelevant or redundant channels. Secondly, AMSformer employs the multi-scale attention mechanism (MSA) to more accurately capture dependencies at different time scales. Finally, the ACA and MSA layers are integrated by a hierarchical encoder (HED) to efficiently utilize adaptive multi-scale information, enhancing prediction accuracy. In this study, actual grain temperature data and six publicly available datasets are used for validation and performance comparison with nine existing models. The results demonstrate that AMSformer outperforms in 36 out of the 58 test cases, highlighting its significant advantages in prediction accuracy and efficiency.

Keywords: grain storage temperature prediction; information redundancy; adaptive channel attention mechanism; multi-scale attention mechanism; hierarchical encoder

Academic Editor: Xiaoshuai Wang

Received: 27 October 2024
Revised: 21 December 2024
Accepted: 26 December 2024
Published: 29 December 2024

Citation: Zhang, Q.; Zhang, W.; Huang, Q.; Wan, C.; Li, Z. AMSformer: A Transformer for Grain Storage Temperature Prediction Using Adaptive Multi-Scale Feature Fusion. *Agriculture* **2025**, *15*, 58. https://doi.org/10.3390/agriculture15010058

1. Introduction

China is the world's most populous developing country and the world's largest food producer, consumer, and importer [1], with 20% of the world's population and less than 9% of its arable land [2]. China's food supply faces challenges due to limited food production resources and growing domestic food consumption. According to the Food and Agriculture Organization of the United Nations (Oliveira), the loss of food exceeds 35 billion kilograms per year, with the largest percentage of loss occurring during the storage stage, at a rate of about 7% [3]. This means that the loss of food in the storage stage is 2.45 billion kilograms, and such a huge loss has had a serious impact on China, significantly increasing the burden on the land. In addition to this, China's food production is faced with challenges such as rising costs of agricultural production, localization of water use [4], and arable land constraints. To address these challenges, there is a need to shift the focus from increasing food production to reducing food losses. Reducing losses during the grain storage phase is one of the most realistic and effective ways to ensure food security in China [5]. The grain storage period is most affected by temperature; too low a temperature will lead to condensation phenomena, while too high a temperature will lead to the rapid reproduction

of pests and molds, posing a serious threat to food security. Therefore, to prevent the hazards of grain storage, it is essential to predict the temperature change of grain storage in advance. In Chinese grain silos, real-time monitoring of the temperature of grain storage piles has been initially realized by deploying sensor arrays in grain silos [6]. However, the judgment of the future trend of grain temperature often relies on the experience of the staff, which may lead to errors in judgment and potentially cause irreparable consequences to the safety of grain storage. Recent studies have shown promising applications of agricultural intelligence models in precision agriculture, such as precision irrigation, pest and disease prediction, and crop growth management [7]. However, these models mainly focus on field management, and research on intelligent predictive modeling for the storage phase of grain bins is still scarce. To this end, we propose AMSformer, which utilizes the transformer structure of multi-scale feature fusion to solve the problem of redundant information in the prediction of grain bin temperature and improve the generalization ability of the model. Therefore, accurate grain temperature prediction is crucial to ensure grain storage safety.

In the field of grain temperature prediction, three challenges still stand in the way of achieving higher prediction efficiency and accuracy. (1) Existing models struggle to efficiently and accurately predict multi-sensor data due to the single output limitations. (2) Most models overlook the spatial topology of the sensor network, restricting a comprehensive examination of temperature variations in stored grains. (3) The discrete distribution of sensors results in the inability to construct a continuous temperature field, hindering comprehensive visualization. However, with the advancement of deep learning, several neural network models have been proposed for stored grain temperature prediction. Transformer-based methods, in particular, have shown great potential due to their ability to capture long-term temporal dependence (trans-temporal dependence) [8]. Besides trans-temporal dependence, trans-dimensional dependence is also crucial for temperature prediction. That is, for a given dimension, information from related series in other dimensions may enhance the prediction. For instance, SageFormer utilizes a series perceptual graph structure to efficiently capture and model dependencies between dimensions [9], and Crossformer introduces dimension segmentation (DSW) embedding and a two-stage attention layer (TSA) to effectively capture cross-temporal and cross-dimensional dependencies [10]. However, existing models are prone to redundant information when capturing cross-time and cross-dimensional dependencies, while redundant information could disrupt the model training process, limiting the model's effectiveness in practical applications and reducing the accuracy of forecasts.

To address the problems mentioned above, this paper proposes an AMSformer model that explicitly exploits cross-dimensional dependencies and mitigates channel redundant information. We design the adaptive channel attention (ACA) mechanism based on DSW embedding, which suppresses the information of irrelevant channels and reduces the model's attention to redundant information. Then, we propose the multi-scale attention (MSA) mechanism, which can flexibly adapt to the changes of different time scales in the time series data. Whether it is the long-term trend, seasonal changes, or short-term fluctuations, the MSA can efficiently capture the important features in the data, thus improving the accuracy of the prediction. The contributions of this paper are as follows:

(1) This paper delves into the existing variants of transformer-based temperature prediction and finds that these models bring redundant information when utilizing cross-dimensional dependencies, which, if not handled, can impact the accuracy of grain temperature prediction.
(2) An adaptive channel attention mechanism and a multi-scale attention mechanism are designed. The former is able to adaptively adjust the weights of different channels according to the characteristics of the input data while suppressing those irrelevant

or redundant channels. The latter is used to capture cross-time dependencies more accurately, and by computing attention at different time scales, the model is able to understand the features and structures in the data more comprehensively, thus improving the accuracy and generalization ability of the prediction.

(3) We utilize a hierarchical encoder to feature-fuse the adaptive channel attention mechanism and the multi-scale attention mechanism, which realizes the effective use of adaptive multi-scale information. Experimental results show that our model achieves state-of-the-art performance on both real-world datasets and synthetic datasets.

2. Related Works

This paper reviews previous storage temperature prediction as well as multivariate time series prediction models. In order to detail the benefits of AMSformer for storage temperature prediction, this section briefly describes the core features of each method, sample implementations, and the strengths and weaknesses of the key findings.

Temperature prediction models can be broadly categorized into statistical models and neural network models. Statistical models can be categorized into vector autoregressive (VAR) models [11] and vector autoregressive moving average (VARMA) models, which assume a linear relationship between the variables and linear prediction by the past values of that variable and other variables. However, as the amount of data increases, deep learning shows better performance than statistical models [12]. For example, Ge et al. used multiple convolutional kernels with shared weights based on convolutional neural networks (CNNs) to capture temperature features at different locations [13], making full use of the temperature information around the target point. Qu et al. combined multiple outputs and spatiotemporal modeling with graph convolutional networks (GCNs) and transformers [14], where GCNs captured the spatial correlation of the sensors in the grain silo and the sensor network, and transformers captured the long-term and short-term temporal features and described the temporal dependencies. Mao et al. proposed temperature prediction algorithms with gated recurrent units (GRU) and multivariate linear regression (MLP), as well as wavelet filtering techniques [15], which efficiently deal with the problem of data sparsity and noise. It can be seen that grain storage temperature is influenced by a variety of complex factors intertwined. In order to achieve more accurate temperature prediction, we can skillfully use the multivariate time series prediction model. This method can comprehensively consider the dynamic relationship of multiple variables over time to grasp the trend of grain storage temperature more comprehensively and improve the accuracy of prediction.

Multivariate time series (MTS) prediction, also known as multivariate time prediction, focuses on the presence of multiple time-dependent variables in a system. Fan et al. combine temporal convolutional networks (TCNs) with spatio-temporal attention mechanisms to capture spatial and temporal dependencies [16]. Jiang et al. decomposed time series into temporal and spatial terms to integrate global and local multivariate information [17]. Li et al. modeled topological relationships between instances using the BERT model and attention mechanism [18]. Jin et al. combined graph neural networks (GNN) and attention mechanisms to capture spatial dependencies through hierarchical signal decomposition on graphs [19]. Lu et al. proposed the complementary time series (CATS), which generates the complementary time series from the original time series and merges the relationships between the series for prediction [20]. Miao et al. designed a progressive quadratic decomposition architecture to extract time series patterns, learn to represent the spatial topology through graph structure, and gate the augmented representation input to GNNML to integrate temporal and spatial information [21]. Guo et al. proposed a matrix attention mechanism that constructs a frequency domain module and a time domain module with

equal weighting of the previous data points to capture the local dynamics and the long-term change patterns, respectively [22]. Wang et al. utilized two self-attention strategies, spatial and temporal self-attention, to focus on the most relevant information in a time series, the former for discovering dependencies between variables and the latter for capturing relationships between historical observations [23]. These models capture both temporal and dimensional dependencies but do not take into account the fact that dimensional dependencies already potentially include the influence of historical data, which can affect the accuracy of MTS predictions.

While transformers are widely used in natural language processing (NLP), vision (CV), and speech processing, variants based on the transformer model for MTS prediction show great potential. Informer [24] exploits the sparsity of the attention scores through KL scatter estimation and proposes the ProbSparse self-attention and distillation techniques. Autoformer [25] renovates transformers into a deep decomposition architecture and concatenates autocorrelation mechanisms. Pyraformer [26] proposes a pyramid attention module to achieve linear time and space complexity. FEDformer [27] argues that the time series has a sparse representation in the frequency domain and proposes frequency domain augmentation structures. Preformer [28] divides the embedded feature vector sequence into multiple segments and utilizes segment-based correlation attention for prediction. STFormer [29] combines a two-stage transformer, which captures spatio-temporal relationships and solves the noise problem, and an adaptive spatio-temporal graph structure, which solves the problem of disordered data. These models mainly focus on capturing temporal dependencies and spatial dependencies, often ignoring the information redundancy between spatio-temporal dependencies. Different from the above approaches, we propose AMSformer, which utilizes adaptive channels and multi-scale feature fusion to suppress redundant channels and capture cross-scale dependencies more accurately.

3. Methodology

In grain storage temperature prediction, the assumption denotes the past 1-T time steps of the temperature datasets, which we aim to predict, where τ denotes the future time step to be predicted, T denotes the past time step, and D denotes the data dimension. In order to utilize the cross-dimensional dependence and reduce the impact of information redundancy, we make the following contributions: in Section 3.1, we collect temperature and humidity, as well as air temperature and air humidity data inside the silo by arranging temperature measurement cables and temperature and humidity sensors inside the silo, and we access these data through a centralized data acquisition system for processing to remove the outliers and missing values. The processed data are stored in a database to construct the grain temperature prediction dataset. In Section 3.2, the adaptive channel attention (ACA) mechanism is used to extract global features, capture global dependencies between channels, and mitigate information redundancy. In Section 3.3, the multi-scale attention layer (MSA) is proposed to capture multi-level information to efficiently capture the dependencies between different time dimensions. In Section 3.4, a hierarchical encoder–decoder module is constructed utilizing the ACA mechanism and the MSA layer to integrate features from different scales in order to generate the final output sequence.

3.1. Datasets

The dimensions of the grain silo involved in this study are 42 m long and 24 m wide, the height of the grain stacking line is 6 m, and the total silo capacity is 4838.4 tons. The cable layout consists of 10 columns and 6 rows, with 7 layers of sensors in each row and column, and the total number of cables is unknown. The sensor buried line standard is 0.5 m from the wall; the top sensor is 0.5 m from the grain surface; the distance between

the sensor columns is 4.56 m; the row spacing is 4.6 m; and the layer height is 0.78 m. The distribution of the temperature-sensitive cables is shown in Figure 1.

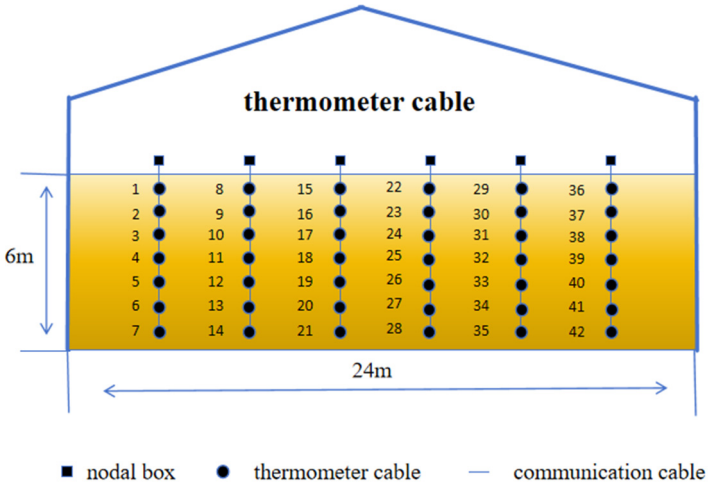


Figure 1. In the grain silo, temperature data from 420 sensors were collected over a time span of 365 days, from 1 January 2021 to 31 December 2021, with each sensor collecting 6 data points per day, for a total of one session of data collection. Sensor acquisition of data may lead to the problem of missing data because the trend of grain storage temperature data is relatively smooth. It is assumed that the data points before and after the missing values have a linear relationship, and the missing values are estimated using linear equations to interpolate the missing data and correct the anomalous data. The data from 42 sensors in one of the cross sections, as well as bin temperature, bin humidity, air temperature, air humidity, and other data, are selected for integration, and the final tensor size represents 46 sensors collecting 6 data points per day for 365 days, ultimately obtaining its own dataset.

3.2. Adaptive Channel Attention

To motivate our approach, we first analyze the embedding methods of the transformer-based models previously used for MTS prediction. Informer and autoformer, among others, embed data points of the same time step into a vector:

$$X_t \rightarrow h_t,\ X_t \in R^D,\ h_t \in R^{d_{model}} \quad (1)$$

where X_t denotes all data points in D-dimension with step size of t. In this way, the inputs $X_{1:T}$ are embedded into the t-vectors {h1, h2, ...; h_T}. Dependencies between t-vectors are then captured for prediction, while cross-dimensional dependencies are not explicitly captured during embedding, which limits their predictive power. Crossformer proposed dimensional segmentation embedding, where nearby points on each dimension are divided into segments of length L and then embedded into the following:

$$S_i^d = \left[X_i^d, X_{i+1}^d, \ldots, X_{i+l-1}^d\right] \quad (2)$$

where S_i^d is the i-th segment of length 1 in the d dimension. Each segment is then embedded into a vector: $h_i = Ws_i + b$ using linear projection and positional embedding. A two-dimensional vector array is obtained, where h_i denotes a univariate time-series segment, explicitly capturing cross-dimensional dependencies. However, data of different dimensions can also introduce information redundancy. To address this problem, we propose the adaptive channel attention (ACA) mechanism, which aims to automatically learn and adjust the weights of different channels based on the characteristics of the input data in order to capture and utilize the important features in the data more efficiently. This attention mechanism allows the model to dynamically adjust the weights of the channels

when dealing with different samples, which improves the model's representational and generalization performance.

Suppose we have a multivariate time series dataset where each sample contains multiple channels of time series data. We can represent these data as a three-dimensional tensor $X \in \mathbb{R}^{N\times T\times D}$, where N is the number of samples, T is the time series length, and D is the number of channels. First, we need to compute the adaptive channel attention weights. Assuming we have q attention heads, we start with a parameter transformation for each attention head:

$$Q_i = XW_i^Q + b_i^Q, K_i = XW_i^K + b_i^K, V_i = XW_i^V + b_i^V \tag{3}$$

where W_i^Q, W_i^K, W_i^V is the parameter matrix and b_i^Q, b_i^K, b_i^V is the bias vector corresponding to the i attention head. Attention weights are then calculated for each attention head:

$$A_i = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) \tag{4}$$

where d_k is the dimension of the key vector. Attention weights are applied to a linear combination of each value to generate the attention output: $O_i = A_i V_i$. We connect the outputs of each head together to form the attention output tensor:

$$Y = \text{concat}(O_1, O_2, \ldots, O_q)W^O \tag{5}$$

where W^O is the output weight matrix. Finally, the attention output is used as an input to the dimensional segmented embedding (DSW), yielding a two-dimensional array of vectors. The illustrative architecture of the adaptive channel attention mechanism is shown in Figure 2.

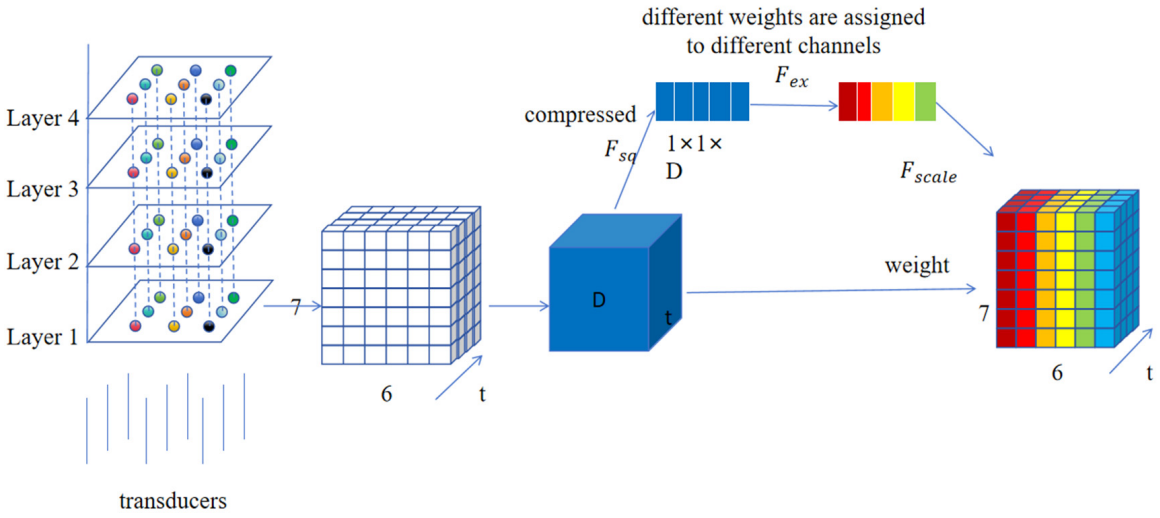


Figure 2. The left side of this figure shows the positional relationship between the sensors. The stored grain temperature data collected by the sensors and the spatial information of the data are used as the input features, then the global spatial information of the input features is compressed. After compression, the excitation part of the input features learns the dimensions of the channels and obtains the weights of the individual channels; finally, the input features are multiplied by the weights to obtain the final output feature map.

3.3. Multi-Scale Attention Mechanisms

The output of the dimensional segmented embedding is multiple univariate time series segments, denoted as H: $H \in R^{i\times d}$, where i is the number of time steps and d is the output dimension of the dimensional segmented embedding. Consider that time-series

patterns for storage temperature prediction often contain multiple different time scales, such as daily, weekly, monthly, and other cycle-specific patterns. For feature representation at each time scale, the multiple attention mechanism is applied. Suppose there is an attention head at the s-th scale. For an attention head h at the s-th scale, the attention weights are calculated as follows: $A^{(s,h)} = \text{softmax}\left(\frac{1}{\sqrt{d_k}}\left(Q^{(s,h)}\right)\left(K^{(s,h)}\right)^T\right)V^{(s,h)}$, where $Q^{(s,h)} = Z^{(s)}W_Q^{(s,h)}$, $K^{(s,h)} = Z^{(s)}W_Q^{(s,h)}$, $V^{(s,h)} = Z^{(s)}W_V^{(s,h)}$ is the query, key, and value matrix obtained by linear permutation of the input feature $Z^{(s)}$; $W_Q^{(s,h)} \in R^{M\times d_k}$, $W_K^{(s,h)} \in R^{M\times d_k}$, $W_V^{(s,h)} \in R^{M\times d_v}$ is the weight matrix obtained by learning, d_k is the dimension of the query/key, and d_v is the dimension of the value; $\sqrt{d_k}$ is the normalization factor used to scale the range of values of the dot product attention; and the softmax function is used to normalize the attention weights obtained. After computing the multi-scale attention weights $A^{(s,h)}$, the feature representations at different time scales are weighted and fused with the corresponding attention weights. The fusion result $Z^{(s)}$ at the s scale is computed as follows: $Z^{(s)} = \sum_{h=1}^{H_s} A^{(s,h)}V^{(s,h)}$. The fusion results $Z^{(s)}$ at all scales are stitched together to obtain the final output of the multi-scale attention mechanism. The illustrative architecture of the multi-scale attention mechanism is shown in Figure 3.

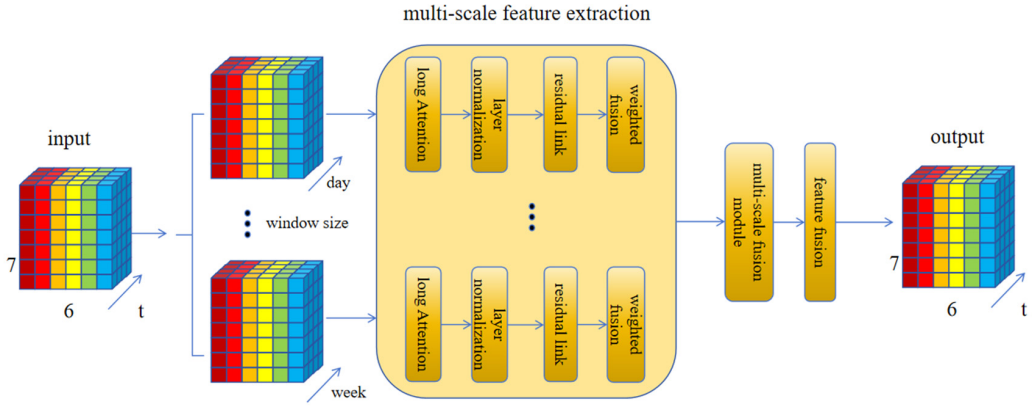


Figure 3. The architecture of the multi-scale attention mechanism for time series analysis, which contains a multi-head attention mechanism, residual connections, layer normalization, and a multi-scale fusion module, aiming to effectively capture and process the information in time series data. First, the input data are shaped as (batch_size, time_steps, input_dim), and each time step contains multiple features. Then, multi-scale feature extraction is performed through different time windows (e.g., 1 day, 1 week, and 1 month); then, the multi-head attention mechanism is applied to compute the importance weight of each time scale, and the specific steps include mapping the input features as query (Q), key (K), and value (V) vectors, calculating the attention weights and performing weighted summation. Subsequently, layer normalization is performed on the output of the multi-head attention mechanism to improve the stability of the model training. Residual connections are added between the input of the multi-head attention mechanism and the output of the layer normalization to prevent the gradient from disappearing. The weighted summation is performed on the features at each time scale according to the attention weights to ensure the features are effectively integrated. Stitching and processing of the weighted fused features of all time scales is performed through the multi-scale fusion module to form a comprehensive feature representation. Further processing the output of the multi-scale fusion module is performed through the feature fusion layer to form the final feature representation.

3.4. Hierarchical Encoder–Decoder

In the layered encoder, we focus on how to effectively integrate the adaptive channel attention (ACA) and multi-scale attention (MSA) layers. In integrating the ACA and MSA layers, we devise an effective strategy that allows the two to work together and fully utilize their respective advantages. The ACA layer is first applied to capture channel correlations

in the input data, then the output of the ACA is used as the input to the MSA layer. Doing so allows the MSA layer to better understand the multi-scale features of the input data, thus improving the model's characterization capability. The specific architecture of the hierarchical encoder is designed as a sequence of stacked modules, each containing an ACA layer and an MSA layer. Such an architecture enables feature extraction of input data at multiple levels and efficiently integrates information from the ACA and MSA layers.

Specifically, each module can contain the following steps: adaptive channel attention (ACA) layer: used to capture the channel correlations in the input data and weigh the important information between channels to improve the model's understanding of the input data; multi-scale attention (MSA) layer: utilizes the output of the adaptive channel attention layer as the input to capture the feature information at different times through the mechanism of multi-head attention to achieve the effective fusion and interaction of features at different scales. Residual connection and layer normalization: in order to enhance the gradient propagation and training effect of the model, residual connection and layer normalization operations can be added in each module to improve the stability and convergence speed of the model.

Through the hierarchical encoder–decoder (HED) structure in the hierarchical encoder, we can realize the effective utilization of adaptive multi-scale information. Specifically, the ACA layer can help the model automatically learn the channel correlations in the input data, which makes the model pay more attention to the important feature channels, while the MSA layer can capture the correlations of the features at different scales, promoting the effective fusion of feature information. Through this combination, the model is able to better understand the feature structure of the input data and achieve better performance in prediction tasks. The illustrative architecture of the layered encoder is shown in Figure 4.

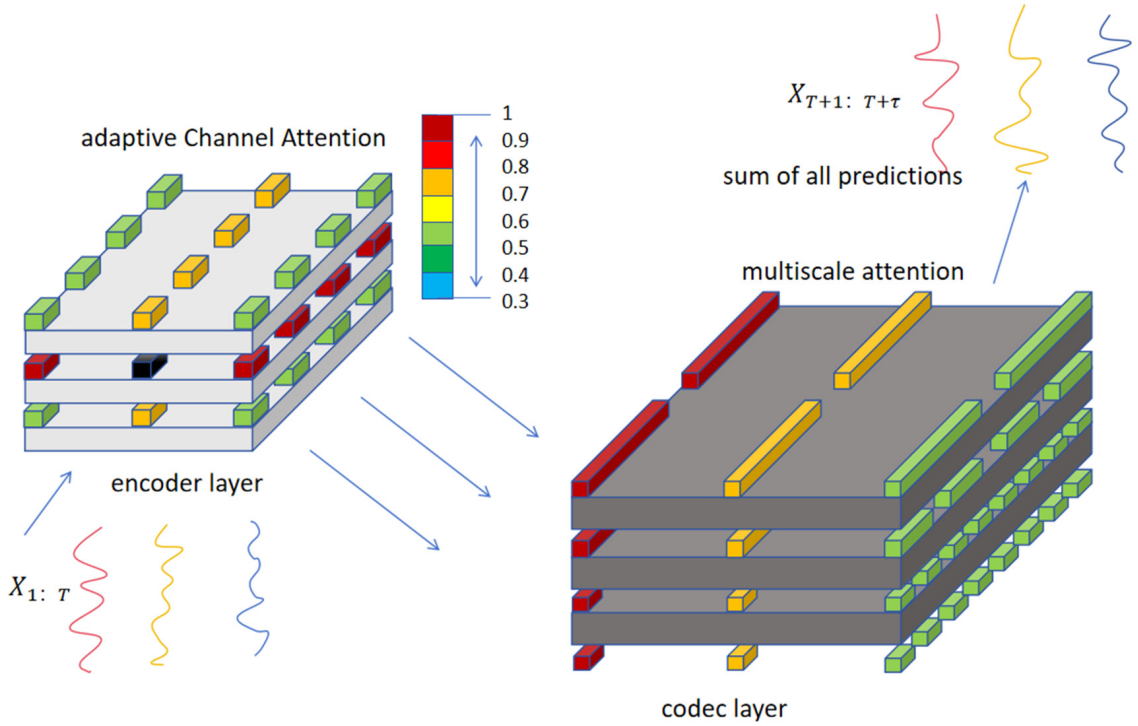


Figure 4. The hierarchical encoder integrates adaptive channel attention and multi-scale attention. The input historical temperature data are filtered for important channels by adaptive channel attention, then the multi-scale attention mechanism is utilized to capture the dependencies between different time scales, and the obtained results are summed to obtain the final output.

4. Experiments

4.1. Spatial and Temporal Correlation Analysis

In the time dimension, the autocorrelation function is used to measure the correlation of time series at different time lags and to determine how one time series affects another. For the time series dataset, the autocorrelation coefficients for the intervals are calculated as follows:

$$\rho_l = \frac{\sum_{t=l+1}^{T}(r_t - \bar{r})(r_{t-l} - \bar{r})}{\sum_{t=1}^{T}(r_t - \bar{r})^2}, 0 \leq l \leq T - 1 \quad (6)$$

where r_t and r_{t-l} denote the grain temperature observations at time t and $t - l$, respectively, r is the average temperature of the sensor over 365 days, and l is the time span. This equation is used to calculate the autocorrelation coefficient corresponding to the temperature data for each layer at different lag times. In Figure 5, as the lag time increases, it is observed that the autocorrelation coefficient decreases at a relatively slow rate from 1 to 0. Since six data points are collected per day, it can be seen that this effect gradually decreases to zero over a period of more than three months, at which time seasonal effects play a dominant role, with different seasons negatively correlating with each other.

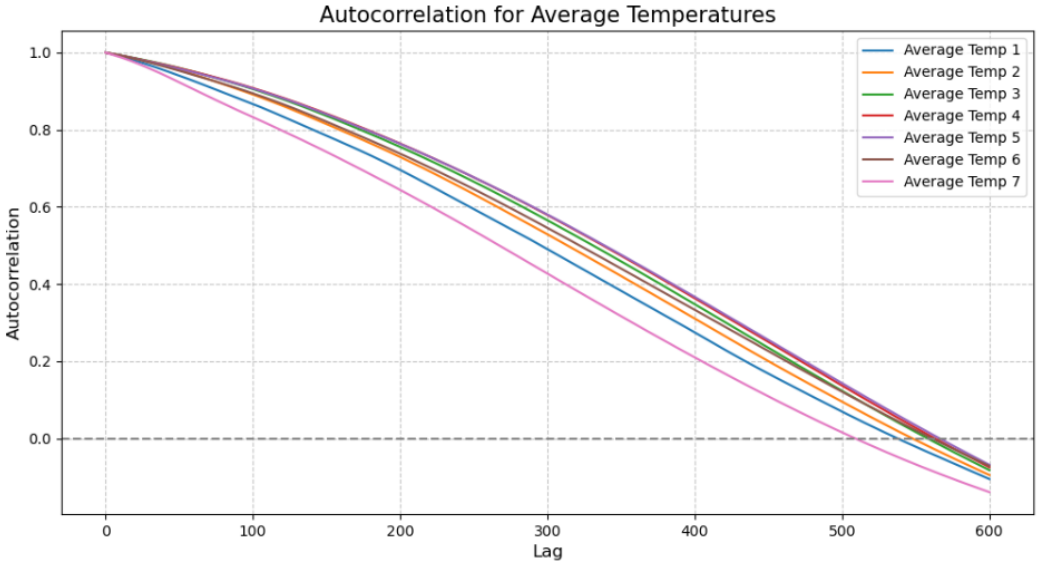


Figure 5. Temporal correlation analysis. The y-axis is the autocorrelation coefficient, which is stronger as it approaches 1 and weaker as it approaches 0, and the x-axis is the time variation.

In the spatial dimension, the global Moran index is used to judge the correlation between spatially adjacent regions to explain that the value of a region's variable is not only determined by its own characteristics, but also influenced by its neighboring regions. Moran I > 0 indicates positive spatial correlation, and the larger the value, the stronger the spatial correlation. The formula is as follows:

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}} \bullet \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \quad (7)$$

where n denotes the number of subregions divided, x_i and x_j are the temperature values of sensors i and j, x is the average of the temperatures of all subregions, and W is the weight matrix of the sensor neighborhoods. W values of 0 or 1 indicate whether a sensor is disconnected or not, respectively. After constructing the coordinate system, the correlation heat map is shown in Figure 6.

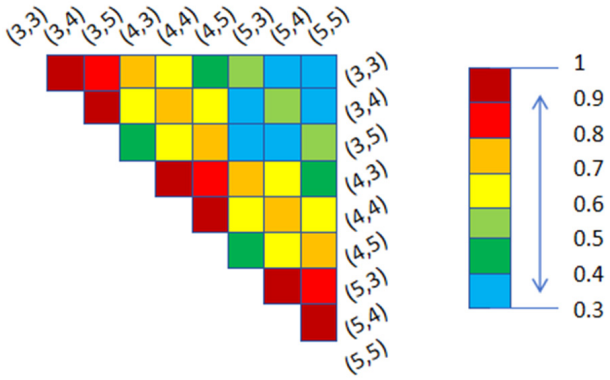
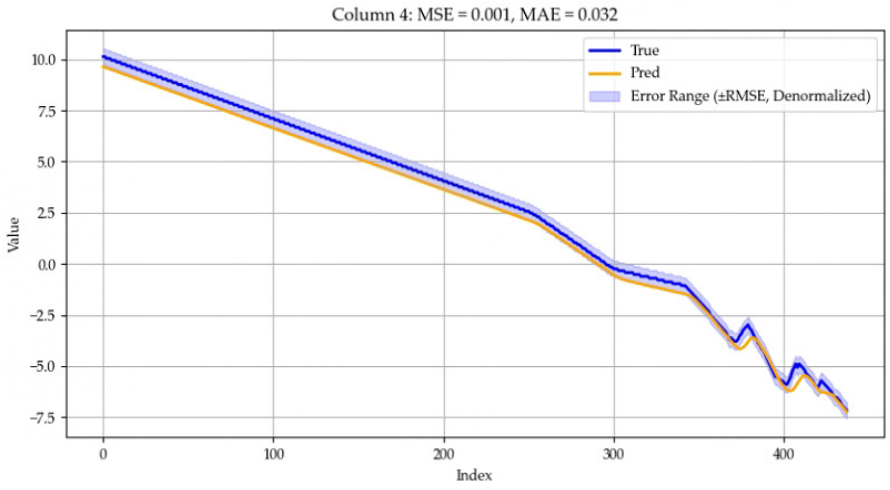


Figure 6. The results shown in the above figure illustrate that the spatial correlation is negatively correlated with the distance between the sensors, and the correlation between (3,3) and (3,4) is stronger than the correlation between (3,3) and (4,3). There is a relatively weak correlation between the layers. From this, it can be obtained that the temperature difference in the horizontal direction is smaller, while the temperature difference in the vertical direction is larger. In conclusion, the above analysis verifies the spatial correlation between temperatures.

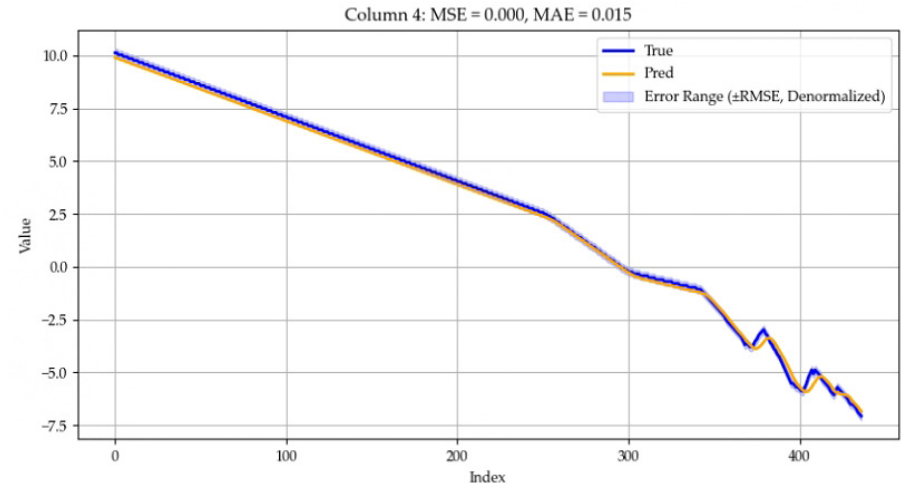
4.2. Experimental Setup and Results

The dataset is partitioned into a training set, a validation set, and a test set with a partition ratio of 0.7:0.1:0.2. The hidden layer dimension is set to 256, the number of multiple notes is set to 4, the segment length is set to 6, the input length is T, the output length is τ, and the window size is set to 2. The training is performed using the Adam optimizer, with the batch size set to 32, the number of training rounds set to 20, and the learning rate set to 1×10^{-4}. The training process is stopped early if the validation loss is not reduced within three cycles. We use Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluation metrics, and all the experiments are repeated five times, with the average of the five experiments used as the reporting metric. The error range was also set, and to make it easier to understand the error range, we used the square root of the MSE, RMSE. The magnitude of the RMSE was consistent with the original data. The results of different predicted lengths for a particular sensor in the grain bin are shown in Figure 7, where the y-axis is in degrees Celsius, and True and Pred represent the actual and predicted temperatures, respectively.

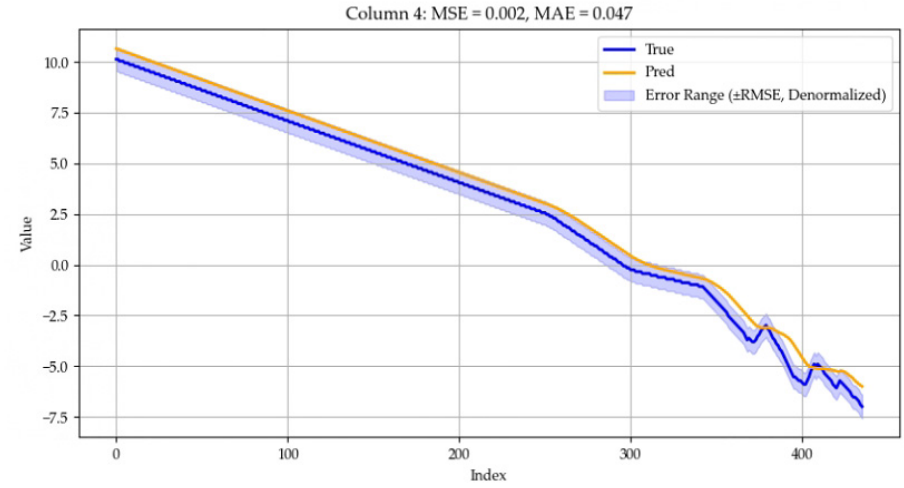


(**a**)

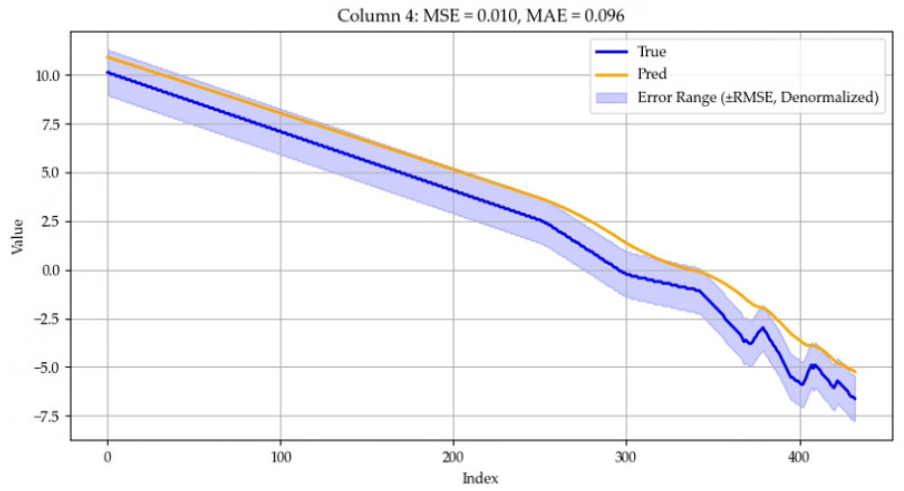
Figure 7. *Cont.*



(**b**)

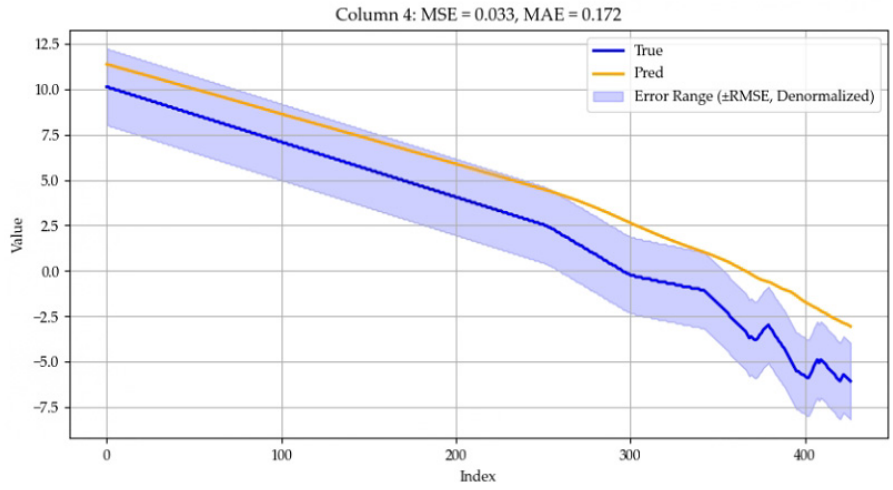


(**c**)

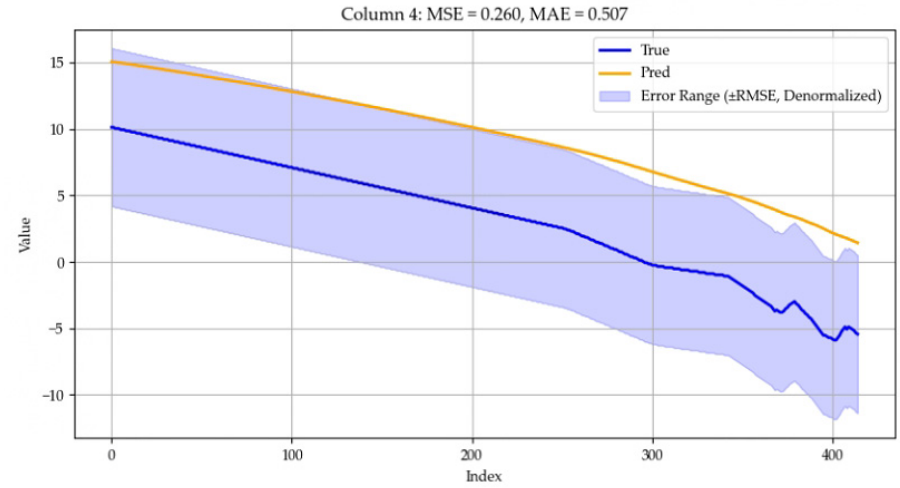


(**d**)

Figure 7. *Cont*.



(**e**)



(**f**)

Figure 7. (**a**). Predicting the next 1 data point, MSE is 0.001 and MAE is 0.032. (**b**). Predicting the next 2 data points, MSE is 0.000 and MAE is 0.015. (**c**). Predicting the next 3 data points, MSE is 0.002 and MAE is 0.047. (**d**). Predicting the next 6 data points, MSE is 0.010 and MAE is 0.096. (**e**). Predicting the next 12 data points, MSE is 0.033 and MAE is 0.172. (**f**). Predicting the next 24 data points, MSE is 0.260 and MAE is 0.507.

4.3. Comparative Analysis of Experiments

We chose six publicly available datasets to validate the model in this paper: 1. ETTh1 (electricity transformer temperature—hourly); 2. ETTm1 (electricity transformer temperature—minute); 3. WTH (weather); 4. ECL (electricity consumption load); 5. ILI (influenza-like illness); 6. traffic. These datasets were selected to comprehensively evaluate the performance of the AMSformer model on different types of time-series data, including high-frequency data, seasonal variations, and joint forecasts of multidimensional data. The first four datasets have the same tra/val/test split as the informer, and the last two datasets are split according to the autoformer's ratio of 0.7:0.1:0.2. This split ratio is widely used in time series forecasting tasks and can ensure that the model can use sufficient historical data during training while also ensuring the independence of the validation and test sets to avoid overfitting. During the training of the AMSformer model, we used a grid search to tune the key hyperparameters to ensure optimal performance of the model on different datasets. The following are the key hyperparameters we tuned: 1. Learning rate: We tested different learning rates to find the optimal value that minimizes the loss during the training

process. We selected the range of [1×10^{-3}, 5×10^{-4}, 1×10^{-4}, 5×10^{-5}] and selected the optimal learning rate based on the performance on the validation set. 2. Hidden layer dimensions: we tried different dimensions of hidden layers (e.g., 64, 128, 256) and evaluated the impact of each setting on the model performance through the validation set. 3. Batch sizes: we tried different batch sizes (e.g., 16, 32, 64), and finally chose a batch size of 32, as this performed optimally in most of the experiments. 4. Number of multi-head attention heads: We chose different numbers of heads of [2,4,8] and adjusted the dimensionality of the multi-head attention to optimize the information extraction process. After validation, four heads were chosen as the optimal configuration.

We performed multiple rounds of testing on the validation set via a grid search combined with cross-validation to select the optimal hyperparameter combination. For each dataset, we performed five experiments, calculated the average performance metrics (e.g., MSE and MAE) for each experiment, and finally selected the hyperparameter combination with the smallest validation error.

We use the following popular MTS prediction models as baselines: LSTMa [30], LSTNet, MTGNN [31], transformer, informer, autoformer, pyraformer, FEDformer, and crossformer. We use the same setup as above with the same setups: the training/validation/testing sets were normalized using the mean and standard deviation of the training set for zero-mean normalization. On each dataset, we evaluated the performance over a changing future window size τ and used the mean squared error (MSE) and mean absolute error (MAE) as evaluation metrics.

As shown in Table 1, our model showed a leading performance on most datasets as well as different prediction length settings, with optimal results in 36 out of a total of 58 cases. On the ETTh1 dataset, AMSformer predicts seasonal and trend changes more accurately than other models such as LSTM by capturing the correlation of multidimensional time series data more efficiently. In a 24 h forecasting task on the WTH dataset, AMSformer accurately captures daytime temperature variations through MSA, while the traditional transformer may have missed some important features. In the traffic dataset, AMSformer outperforms pyraformer and FEDformer in cross-level information fusion. The ILI dataset has a small amount of data and weak temporal features, which causes AMSformer's multi-scale attention mechanism and channel attention mechanism to fail to give full play to its advantages. In addition to this, our model generally outperforms the other baselines in short-term forecasting.

Table 1. MSE/MAE for different models with different prediction lengths. At the top of the table are the different models, and on the left side are the publicly available datasets and the different prediction lengths. Underlining indicates that the previous model is the best, and bolding indicates that it is better compared to the previous results.

Models		**LSTMa**		**LSTnet**		**MTGNN**		**Transformer**		**Informer**		**Autoformer**		**Pyraformer**		**FEDformer**		**Crossformer**		**AMSformer**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	0.650	0.624	1.293	0.901	0.336	0.393	0.620	0.577	0.577	0.549	0.439	0.440	0.493	0.507	0.318	0.384	0.305	0.367	**0.299**	**0.365**
	48	0.720	0.675	1.456	0.960	0.386	0.429	0.692	0.671	0.685	0.625	0.429	0.442	0.554	0.544	0.342	0.396	0.352	0.394	0.347	0.395
	168	1.212	0.867	1.997	1.214	0.466	0.474	0.947	0.797	0.931	0.752	0.493	0.479	0.781	0.675	0.412	0.449	0.410	0.441	0.413	**0.441**
	336	1.424	0.994	2.655	1.369	0.736	0.643	1.094	0.813	1.128	0.873	0.509	0.492	0.912	0.747	0.456	0.474	0.440	0.461	0.444	0.462
	720	1.960	1.322	2.143	1.380	0.916	0.750	1.241	0.917	1.215	0.896	0.539	0.537	0.993	0.792	0.521	0.515	0.519	0.524	**0.510**	**0.520**
ETTm1	24	0.621	0.629	1.968	1.170	0.260	0.324	0.306	0.371	0.323	0.369	0.410	0.428	0.310	0.371	0.290	0.364	0.211	0.293	**0.199**	**0.283**
	48	1.392	0.939	1.999	1.215	0.386	0.408	0.465	0.470	0.494	0.503	0.485	0.464	0.465	0.464	0.342	0.396	0.300	0.352	**0.288**	**0.343**
	96	1.339	0.913	2.762	1.542	0.428	0.446	0.681	0.612	0.678	0.614	0.502	0.476	0.520	0.504	0.366	0.412	0.320	0.373	**0.317**	**0.364**
	288	1.740	1.124	1.257	2.076	0.469	0.488	1.162	0.879	1.056	0.786	0.604	0.522	0.729	0.657	0.398	0.433	0.404	0.427	**0.392**	**0.418**
	672	2.736	1.555	1.917	2.941	0.620	0.571	1.231	1.103	1.192	0.926	0.607	0.530	0.980	0.678	0.455	0.464	0.569	0.528	0.529	0.510
WTH	24	0.546	0.570	0.615	0.545	0.307	0.356	0.349	0.397	0.335	0.381	0.363	0.396	0.301	0.359	0.357	0.412	0.294	0.343	**0.292**	**0.343**
	48	0.829	0.677	0.660	0.589	0.388	0.422	0.386	0.433	0.395	0.459	0.456	0.462	0.376	0.421	0.428	0.458	0.370	0.411	**0.366**	**0.407**
	168	1.038	0.835	0.748	0.647	0.498	0.512	0.613	0.582	0.608	0.567	0.574	0.548	0.519	0.521	0.564	0.541	0.473	0.494	**0.469**	**0.490**
	336	1.657	1.059	0.782	0.683	0.506	0.523	0.707	0.634	0.702	0.620	0.600	0.571	0.539	0.543	0.533	0.536	0.495	0.515	0.498	0.517
	720	1.536	1.109	0.851	0.757	0.510	0.527	0.834	0.741	0.831	0.731	0.587	0.570	0.547	0.553	0.562	0.557	0.526	0.542	0.521	0.537
ECL	48	0.486	0.572	0.369	0.445	0.173	0.280	0.334	0.399	0.344	0.393	0.241	0.351	0.478	0.471	0.229	0.338	0.156	0.255	**0.150**	**0.254**
	168	0.574	0.602	0.394	0.476	0.236	0.320	0.353	0.420	0.368	0.424	0.299	0.387	0.452	0.455	0.263	0.361	0.231	0.309	**0.223**	**0.307**
	336	0.886	0.795	0.419	0.477	0.328	0.373	0.381	0.439	0.381	0.431	0.375	0.428	0.463	0.456	0.305	0.386	0.323	0.369	**0.301**	**0.353**
	720	1.676	1.095	0.556	0.565	0.422	0.410	0.391	0.438	0.406	0.443	0.377	0.434	0.480	0.461	0.372	0.434	0.404	0.423	0.408	0.426
	960	1.591	1.128	0.605	0.599	0.471	0.451	0.492	0.550	0.460	0.548	0.366	0.426	0.550	0.489	0.393	0.449	0.433	0.438	0.436	0.441
ILI	24	4.220	1.335	4.975	1.660	4.265	1.387	3.954	1.323	4.588	1.462	3.101	1.238	3.970	1.338	2.687	1.147	3.041	1.186	2.926	**1.139**
	36	4.771	1.427	5.322	1.659	4.777	1.496	4.167	1.360	4.845	1.496	3.397	1.270	4.377	1.410	2.887	1.160	3.406	1.232	3.154	**1.160**
	48	4.945	1.462	5.425	1.632	5.333	1.592	4.746	1.463	4.865	1.516	2.947	1.203	4.811	1.503	2.797	1.155	3.459	1.221	3.256	1.158
	60	5.176	1.504	5.477	1.675	5.070	1.552	5.219	1.553	5.212	1.576	3.019	1.202	5.204	1.588	2.809	1.163	3.640	1.305	3.396	1.208
Traffic	24	0.668	0.378	0.648	0.403	0.506	0.278	0.597	0.332	0.608	0.334	0.550	0.363	0.606	0.338	0.562	0.375	0.491	0.274	**0.481**	**0.270**
	24	0.668	0.378	0.648	0.403	0.506	0.278	0.597	0.332	0.608	0.334	0.550	0.363	0.606	0.338	0.562	0.375	0.491	0.274	**0.481**	**0.270**
	48	0.709	0.400	0.709	0.425	0.512	0.298	0.658	0.369	0.644	0.359	0.595	0.376	0.619	0.346	0.567	0.374	0.519	0.295	**0.506**	**0.285**
	168	0.900	0.523	0.713	0.435	0.521	0.319	0.664	0.363	0.660	0.391	0.649	0.407	0.635	0.347	0.607	0.385	0.513	0.289	**0.512**	**0.287**
	336	1.067	0.599	0.741	0.451	0.540	0.335	0.654	0.358	0.747	0.405	0.624	0.388	0.641	0.347	0.624	0.389	0.530	0.300	**0.528**	**0.299**
	720	1.461	0.787	0.768	0.474	0.557	0.343	0.685	0.370	0.792	0.430	0.674	0.417	0.670	0.364	0.623	0.378	0.573	0.313	0.572	**0.310**

4.4. Ablation Experiment

In our approach, there are three components: the ACA layer, the MSA layer, and the HED. We use the transformer as a baseline to compare the two ablation versions: (1) ACA, (2) MSA, (3) ACA + MSA.

We analyze the results shown in Table 2. (1) When only the ACA layer is retained, the model is able to capture local contextual information. The ACA layer mainly enhances the local attention mechanism, which makes the model outperform the baseline transformer to a certain extent, but its overall performance enhancement is limited due to the lack of support for multi-scale information aggregation and hierarchical decoding. (2) When only the MSA layer is retained, the model is able to aggregate information at different scales, enhancing the ability to understand the global context. Although the MSA layer improves the aggregation of multi-scale information, without the local context aggregation of the ACA layer and the hierarchical decoding of HED, the model still has limited performance enhancement. (3) With the addition of the MSA layer to the ACA layer, the model is able to capture local context information as well as to aggregate information at different scales. This version significantly improves the overall performance of the model, indicating that the ACA layer and the MSA layer play an important synergistic role in information aggregation and context understanding. (4) The full model integrates the ACA layer and the MSA layer into the HED, which enables the model to better utilize the hierarchically embedded information in the decoding stage. The addition of the HED significantly improves the model's ability to capture details and overall performance, resulting in the best model performance being optimal.

Table 2. Ablation studies of the ETTh1 dataset, preserving the results of the ablation experiments for the ACA layer, MSA layer, ACA + MSA and ACA + MSA + HED.

Models		**Transformer**		**ACA**		**MSA**		**ACA + MSA**		**ACA + MSA + HED**	
Metric		**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	24	0.620	0.577	0.612	0.571	0.615	0.573	0.331	0.387	0.299	0.365
	48	0.692	0.671	0.684	0.665	0.687	0.669	0.381	0.423	0.347	0.395
	168	0.947	0.797	0.941	0.791	0.932	0.783	0.461	0.465	0.413	0.441
	336	1.094	0.813	1.089	0.806	1.074	0.793	0.728	0.632	0.444	0.462
	720	1.241	0.917	1.238	0.914	1.219	0.897	0.869	0.706	0.510	0.520

5. Conclusions

We propose AMSformer, a transformer-based model that utilizes adaptive multi-scale feature fusion for grain storage temperature prediction. Specifically, adaptive channel attention (ACA) assigns different weights to each channel, allowing the model to adaptively focus on those channels that are more important to the prediction task. A multi-scale attention (MSA) mechanism was designed to compute attention weights at different time scales and weighted fusion of features at different scales. Then, a hierarchical encoder–decoder (HED) was designed to synthesize the temporal and dimensional dependencies using ACA and MSA. The results show that it is very effective.

We analyze the limitations of our work. First, although AMSformer has achieved significant results in grain storage temperature prediction, its performance may be affected by the quality and quantity of data. In practice, the presence of noise or missing values in the dataset may have some impact on the prediction results of the model. Secondly, the complexity and computational cost of the model are also issues to be considered. AMSformer employs multi-scale attention and adaptive channel attention mechanisms, which increase the complexity and computational requirements of the model. With limited resources, the structure and parameters of the model may need to be further optimized to reduce computational costs and improve efficiency.

To address these limitations, we plan to conduct further exploration and improvement in our future work. First, we will investigate how to deal with noise and missing values in the data more effectively to improve the robustness and generalization ability of the model. Second, we will try to optimize the structure and parameters of the model to reduce the computational cost and improve the prediction performance. Adaptive filtering methods, such as Kalman filtering, are introduced to dynamically estimate and correct the noisy data. The introduction of masked self-encoders will allow the model to directly process input data containing missing values. The introduction of sparse attention mechanisms will also be explored to further reduce the computational complexity, and we will use mixed-precision training to speed up computation and reduce graphics memory requirements. In addition, we will explore the application of AMSformer to other related fields to verify its generalization and scalability.

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

Funding: This research was funded by the Zhongyuan Science and Technology Innovation Leading Talent Program (244200510024), Development and Promotion Project of Henan Province (Grant No. 242102211002), and the High-Level Talent Research Start-up Fund Project of Henan University of Technology (2023BS040).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cao, B.; Tang, L.; Hu, B.; Zhao, X. Global Grain Crisis and China's Grain Security. *Int. Econ. Rev.* **2021**, *2*, 9–21.
2. Liu, Z.; Zhong, H.; Li, Y.; Wen, Q.; Liu, X.; Jian, Y. Change in grain production in China and its impacts on spatial supply and demand distributions in recent two decades. *J. Nat. Resour.* **2021**, *36*, 1413–1425. [CrossRef]
3. Oliveira, L.F.P.; Moreira, A.P.; Silva, M.F. Advances in agriculture robotics: A state-of-the-art review and challenges ahead. *Robotics* **2021**, *10*, 52. [CrossRef]
4. Heller, M.C.; Willits-Smith, A.; Mahon, T.; Keoleian, G.A.; Rose, D. Individual US diets show wide variation in water scarcity footprints. *Nat. Food* **2021**, *2*, 255–263. [CrossRef] [PubMed]
5. Chen, X.; Wu, L.; Shan, L.; Zang, Q. Main factors affecting post-harvest grain loss during the sales process: A survey in nine provinces of China. *Sustainability* **2018**, *10*, 661. [CrossRef]
6. Zhao, L.; Wang, J.; Li, Z.; Hou, M.; Dong, G.; Liu, T.; Sun, T.; Grattan, K.T. Quasi-distributed fiber optic temperature and humidity sensor system for monitoring of grain storage in granaries. *IEEE Sens. Journal.* **2020**, *20*, 9226–9233. [CrossRef]
7. SS, V.C.; Hareendran, A.; Albaaji, G.F. Precision farming for sustainability: An agricultural intelligence model. *Comput. Electron. Agric.* **2024**, *226*, 109386. [CrossRef]
8. Chen, W.; Wang, W.; Peng, B.; Wen, Q.; Zhou, T.; Sun, L. Learning to rotate: Quaternion transformer for complicated periodical time series forecasting. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 146–156.
9. Zhang, Z.; Wang, X.; Gu, Y. Sageformer: Series-aware graph-enhanced transformers for multivariate time series forecasting. *arXiv* **2023**, arXiv:2307.01616.
10. Zhang, Y.; Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
11. Duan, S.; Yang, W.; Wang, X.; Mao, S.; Zhang, Y. Grain pile temperature forecasting from weather factors: A support vector regression approach. In Proceedings of the 2019 IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 1–13 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 255–260.

12. Qi, Y.; Li, Q.; Karimian, H.; Liu, D. A hybrid model for spatiotemporal forecasting of PM 2.5 based on graph convolutional neural network and long short-term memory. *Sci. Total Environ.* **2019**, *664*, 1–10. [CrossRef] [PubMed]
13. Ge, L.; Chen, C.; Li, Y.; Mo, T.; Li, W. A CNN-based temperature prediction approach for grain storage. *Int. J. Internet Manuf. Serv.* **2020**, *7*, 345–357. [CrossRef]
14. Qu, Z.; Zhang, Y.; Hong, C.; Zhang, C.; Dai, Z.; Zhao, Y.; Gu, Z. Temperature forecasting of grain in storage: A multi-output and spatiotemporal approach based on deep learning. *Comput. Electron. Agric.* **2023**, *208*, 107785. [CrossRef]
15. Mao, B.; Tao, S.; Li, B. Grain Temperature Prediction Based on GRU Deep Fusion Model. *Int. J. Inf. Technol. Decis. Mak.* **2024**, 1–19. [CrossRef]
16. Fan, J.; Zhang, K.; Huang, Y.; Zhu, Y.; Chen, B. Parallel spatio-temporal attention-based TCN for multivariate time series prediction. *Neural Comput. Appl.* **2023**, *35*, 13109–13118. [CrossRef]
17. Jiang, Z.; Ning, Z.; Miao, H.; Wang, L. STDNet: A Spatio-Temporal Decomposition Neural Network for Multivariate Time Series Forecasting. *Tsinghua Sci. Technol.* **2024**, *29*, 1232–1247. [CrossRef]
18. Li, X.; Luo, S.; Pan, L.; Wu, Z. Adapt to small-scale and long-term time series forecasting with enhanced multidimensional correlation. *Expert Syst. Appl.* **2024**, *238*, 122203. [CrossRef]
19. Kim, J.; Lee, H.; Yu, S.; Hwang, U.; Jung, W.; Yoon, K. Hierarchical Joint Graph Learning and Multivariate Time Series Forecasting. *IEEE Access* **2023**, *11*, 118386–118394. [CrossRef]
20. Lu, J.; Han, X.; Sun, Y.; Yang, S. CATS: Enhancing Multivariate Time Series Forecasting by Constructing Auxiliary Time Series as Exogenous Variables. *arXiv* **2024**, arXiv:2403.01673.
21. Miao, H.; Zhang, Y.; Ning, Z.; Jiang, Z.; Wang, L. TDG4MSF: A temporal decomposition enhanced graph neural network for multivariate time series forecasting. *Appl. Intell.* **2023**, *53*, 28254–28267. [CrossRef]
22. Guo, K.; Yu, X. Long-Term Forecasting Using MAMTF: A Matrix Attention Model Based on the Time and Frequency Domains. *Appl. Sci.* **2024**, *14*, 2893. [CrossRef]
23. Wang, D.; Chen, C. Spatiotemporal Self-Attention-Based LSTNet for Multivariate Time Series Prediction. *Int. J. Intell. Syst.* **2023**, *2023*, 9523230. [CrossRef]
24. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 11106–11115.
25. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
26. Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A.X.; Dustdar, S. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In Proceedings of the Tenth International Conference on Learning Representations (ICLR 2022) New Orleans, Louisiana, USA, 25–29 April 2022.
27. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Proceedings of the 39th International Conference on Machine Learning (PMLR), Baltimore, MD, USA, 17–23 July 2022; pp. 27268–27286.
28. Du, D.; Su, B.; Wei, Z. Preformer: Predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
29. Xiao, Y.; Liu, Z.; Yin, H.; Wang, X.; Zhang, Y. STFormer: A dual-stage transformer model utilizing spatio-temporal graph embedding for multivariate time series forecasting. *J. Intell. Fuzzy Syst.* **2024**, *46*, 6951–6967. [CrossRef]
30. Bahdanau, D. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
31. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the dots: Multivariate time series forecasting with graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, San Diego, CA, USA, 6–10 July 2020; pp. 753–763.