


Article

# KAN-Transformer Model for UltraShort-Term Wind Power Prediction Based on EWMA Data Processing

Feng Xing <sup>1</sup>, Yanlong Gao <sup>1</sup>, Lipeng Kang <sup>1</sup>, Mingming Zhang <sup>2</sup> and Caiyan Qin <sup>2,\*</sup><sup>1</sup> School of Electrical Engineering, Liaoning University of Technology, Jinzhou 121001, China<sup>2</sup> School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China

\* Correspondence: qincaiyan@hit.edu.cn

**Abstract:** When using the Transformer model for wind power prediction, the presence of noise in wind power data and the model's final layer relying solely on a simple linear output reduces the model's ability to capture nonlinear relationships, leading to a decrease in prediction accuracy. To address these issues, this paper proposes an ultrashort-term wind power prediction model based on exponential weighted moving average (EWMA) data processing and Kolmogorov–Arnold Network (KAN)-Transformer. First, multiple variable features are smoothed using EWMA, which suppresses noise while preserving the original data trends. Then, the EWMA-processed data is input into the Encoder and Decoder modules of the Transformer model to extract features. The output from the Decoder layer is then passed through the KAN layer, built using a cubic B-spline function, to enhance the model's ability to capture nonlinear relationships, thereby improving the prediction accuracy of the Transformer model for wind power. Finally, experimental analysis is conducted, and it shows that the proposed model achieves the highest prediction accuracy, with a mean absolute error of 4.38 MW, a root mean squared error of 7.37 MW, and a coefficient of determination of 98.73%.

**Keywords:** transformer; wind power prediction; noise; EWMA; nonlinear relationship; KAN



**Citation:** Xing, F.; Gao, Y.; Kang, L.; Zhang, M.; Qin, C. KAN-Transformer Model for UltraShort-Term Wind Power Prediction Based on EWMA Data Processing. *Appl. Sci.* **2024**, *14*, 9630. <https://doi.org/10.3390/app14219630>

Academic Editor: Frede Blaabjerg

Received: 24 September 2024

Revised: 19 October 2024

Accepted: 20 October 2024

Published: 22 October 2024



**Copyright:** © 2024 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) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the past two decades, wind energy has become a key renewable energy source for addressing energy crises and climate change [1]. With fossil fuels gradually depleting and the emergence of new situations emphasizing energy conservation and emission reduction, wind power generation is increasingly receiving attention from countries. However, a major drawback of wind power is that wind is intermittent and highly fluctuating [2], which is not conducive to the safe and stable operation and economic dispatch of the power system. Wind power prediction models can be classified into long-term [3], medium-term [4], short-term [5,6], and ultrashort-term [7,8] predictions based on time scales. Short-term and ultrashort-term predictions are highly correlated with the operation of integrated generating systems in wind farms. High-precision prediction can reduce wind curtailment, helping power plants adjust their short-term generation plans and reserve capacities promptly. Therefore, designing an accurate, fast, and reliable ultrashort-term wind power prediction model is of great significance.

Time series prediction models can generally be categorized into three main types: physical models [9], statistical models [10], and artificial intelligence methods [11]. Physical forecasting models, based on numerical weather prediction (NWP) and aerodynamics, are effective and interpretable in long-term time series forecasting. However, short-term NWP forecasts often lack precision and may not meet forecasting requirements. Statistical methods rely on the statistical analysis of historical time series data to predict wind power generation. These models mainly include autoregressive and moving average (ARMA) [12], autoregressive integrated moving average (ARIMA) [13], and Markov models [14], among others. Statistical models were widely used in wind power time series forecasting in the

past, but with the rapid increase in data volume, they have gradually lost their advantage compared to machine learning methods. In recent years, deep learning has seen rapid development and many deep learning models have been applied in the field of wind speed and wind power prediction. Among them, long short-term memory (LSTM) networks [15,16], gated recurrent units (GRUs) [17,18], and other methods have been widely used in wind power prediction.

Traditional recurrent neural networks and their variants have achieved good results in power prediction, but they have limitations in handling parallel processing and establishing long-term temporal relationships. With the advent of attention mechanisms, the Transformer model was proposed and subsequently widely used [19–21]. Thanks to its powerful ability to model time correlations, attention mechanisms have been widely used in time series prediction tasks, leading to a series of models using the Transformer for wind power prediction. Literature [22] introduced the FEDformer model based on frequency-enhanced Transformer, for wind power generation and wind speed prediction. This model is the entry of Transformer-based models into the field of wind power prediction. In literature [23], a hybrid deep learning model has been proposed that utilizes historical data for calibration. The model incorporates a Transformer network that accesses historical data in the encoder to enhance its predictions with correction information. Literature [24] proposed a super-short-term wind speed prediction model based on complementary ensemble empirical mode decomposition (CEEMD) and the BILTM-Transformer. It used CEEMD decomposition to reduce the instability of wind speed sequences. Literature [25] introduced the Wind Transformer model, which addresses the segmentation problem and predicts the power output of each turbine individually.

However, both the Transformer and its variant models rely on simple linear layers for dimensionality reduction to produce the final prediction, which, to some extent, reduces the model's ability to capture nonlinear relationships. As research in this area has progressed, the emergence of Kolmogorov–Arnold networks (KANs) in 2024 [26] successfully addressed this limitation, improving model prediction accuracy. KAN is an extension of deep learning based on the mathematical principles of the Kolmogorov–Arnold representation theorem (KART). This theorem proves that a multivariate continuous function can be approximated by the summation of a finite number of nonlinear elementary functions. The KAN network initializes several B-spline functions and learns their parameters. By summing these B-spline functions to form nonlinear elementary functions, and then aggregating all the nonlinear elementary functions to map them to the prediction dimension, the model's capacity to capture nonlinear relationships is enhanced. Since KAN networks are adept at capturing nonlinear relationships, they have subsequently been integrated with time-series models. For instance, literature [27] demonstrates that KAN networks, due to their highly accurate modeling of complex nonlinear systems, have proven to be effective in energy management systems, showcasing their practical utility and effectiveness. Another study [28] incorporated KAN into time-series forecasting to improve interpretability and nonlinear relationship modeling, verifying the effectiveness of KAN in the field of time-series prediction.

When forecasting future wind power, the prediction is not only related to past wind power values but also closely linked to other variables such as historical wind speed, wind direction, temperature, and more. Therefore, multiple variable features are often introduced alongside wind power features. However, during this process, the presence of noise and outliers in the multivariable data can reduce the accuracy of the prediction model. To address this, the exponential weighted moving average (EWMA) method is introduced. By controlling the decay factor, EWMA smooths the data while retaining the original trends, thereby reducing the impact of noise on the prediction accuracy. For example, in [29], EWMA analysis is used in the monitoring stage to detect abnormal changes in features. In [30], EWMA assigns greater weight to more recent data, smoothing time-series data and improving performance through this optimized data-mining approach.

In conclusion, when using the Transformer model with point-wise attention mechanisms for wind power prediction, the model’s accuracy is reduced due to noise in wind power data and the inability of the simple linear output layer to capture nonlinear relationships. To address this, this paper proposes a KAN-Transformer model incorporating EWMA data processing. The contributions of this paper are as follows: First, EWMA is employed to denoise the introduced multivariable features, improving the predictability of wind power data. Second, the KAN layer is used to replace the Transformer model’s final linear layer, enhancing the model’s ability to capture nonlinear relationships and thus improving prediction accuracy.

The chapter arrangement is as follows: In Section 2, EWMA is used to reduce the impact of noise on the model’s prediction accuracy. KAN is combined with the Transformer model, where features are extracted through the Encoder–Decoder structure and then passed into the KAN layer to obtain prediction results, enhancing the model’s ability to capture nonlinear relationships. In Section 3, simulation experiments and analysis are conducted. In Section 4, conclusions are summarized.

## 2. Research Methodology and EWMA-KAN-Transformer Model

### 2.1. Method of the Present Research

To clearly show the research methods used in this paper, an overall framework diagram of the research methods is presented in Figure 1.

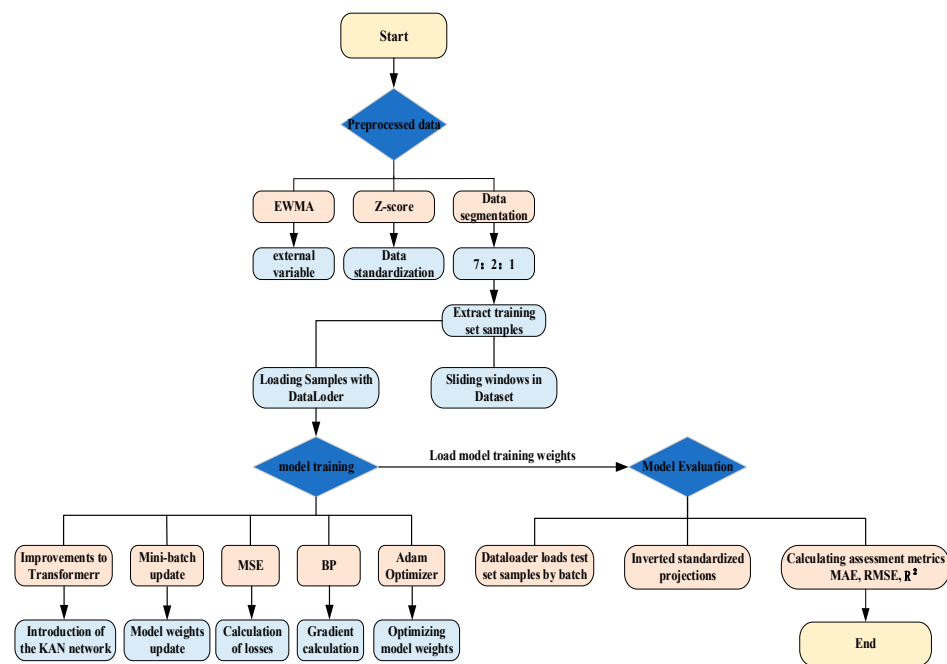


Figure 1. Overall framework diagram of this research.

In Figure 1, the process begins with the preprocessing of the dataset. The collected external variable data are denoised using the EWMA method, followed by Z-score normalization. The data are then divided into training, validation, and test sets in a ratio of 7:2:1. Next, the training set samples are extracted using a sliding window method within the Dataset class to create time-series forecasting samples. A DataLoader class is defined to facilitate data loading for model training. The model training process consists of five main parts: defining the improved KAN-Transformer model, loading data in batches using the Mini-batch method, inputting the loaded data into the KAN-Transformer model to obtain prediction values, calculating the loss by comparing the predicted values with the actual values using mean squared error (MSE), and passing the obtained loss to the backpropagation (BP) network for backward propagation. The model weights are then optimized using

the adaptive moment estimation method (ADAM). Finally, the trained model weights are loaded for testing and evaluation. During the testing phase, data are loaded in batches into the model to obtain prediction results. After denormalizing the prediction results, the root mean squared error (RMSE), mean absolute error (MAE), and coefficient of determination ( $R^2$ ) are calculated to assess the model's performance on the test set.

2.2. Structure of KAN-Transformer Prediction Model Based on EWMA Data Processing

To reduce noise in the wind power data and enhance the ability of the Transformer model to capture nonlinear relationships, this paper proposes an ultrashort-term wind power prediction model based on EWMA data processing and KAN-Transformer. The model is illustrated in Figure 2.

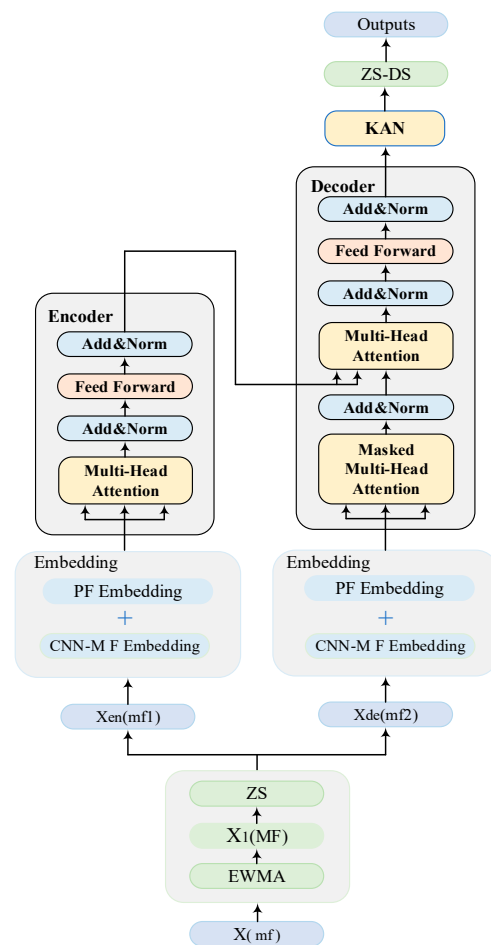


Figure 2. This article proposes a model structure.

In Figure 2, the process begins by applying EWMA data processing to the original data to reduce the impact of noise in the wind power data on model prediction accuracy. The processed data are then used with a sliding window approach to obtain inputs for the Encoder and Decoder, which are embedded through the Embedding layer and then processed through the Encoder–Decoder architecture to extract historical features. Finally, the output from the Decoder layer is fed into the KAN layer, constructed with a cubic B-spline function, to obtain the final prediction result, thereby enhancing the model's ability to capture nonlinear relationships.

The experimental dataset used in this paper comes from a publicly available dataset of a wind power plant with an installed capacity of 200 MW [31]. Data are recorded every fifteen minutes, totaling 35,041 time steps. The feature columns are numbered and named

as shown in Table 1, where columns 1–13 represent the introduced multivariable features, and serial 14 represents the wind power feature.

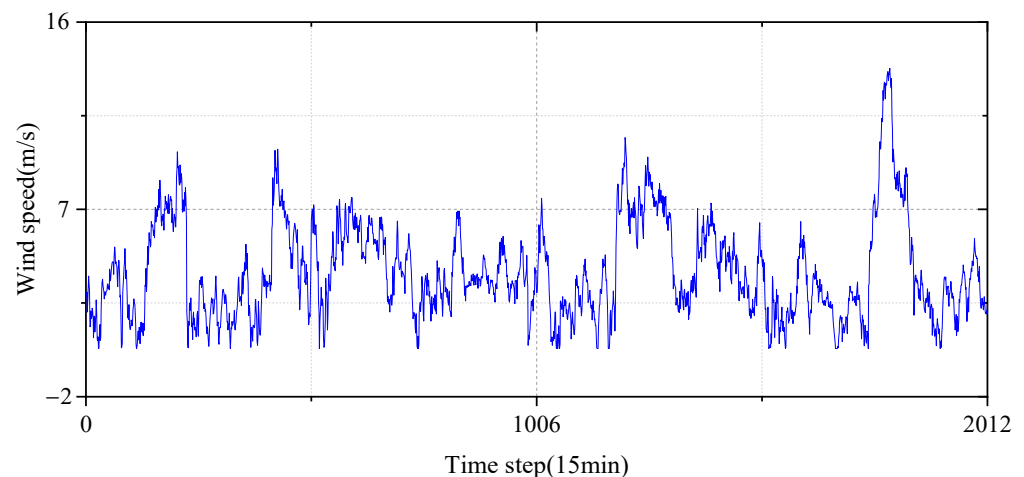
**Table 1.** Feature serial number and feature name.

Serial Number	Feature Name
1	10 m wind speed
2	30 m wind speed
3	50 m wind speed
4	70 m wind speed
5	Hub height wind speed
6	10 m wind direction
7	30 m wind direction
8	50 m wind direction
9	70 m wind direction
10	Hub height wind direction
11	Temperature
12	Pressure
13	Humidity
14	Actual power generation

### 2.2.1. Data Processing Method of EWMA

When predicting future wind power, the forecast is influenced by past wind power features and multiple variable features. To improve predictions, variables such as wind speed, wind direction, and temperature are introduced alongside wind power features. However, the introduction of these multiple variable features can introduce noise, which reduces the accuracy of the model. Therefore, this paper applies the EWMA method for data smoothing on the introduced multivariable features. This method helps to suppress noise and retain data trends, making the smoothed data more reflective of actual changes.

Taking the variable 30 m wind speed in sequence 2 as an example, the original data are plotted as shown in Figure 3.



**Figure 3.** Partial data points from serial number 2.

In Figure 3, the data show frequent fluctuations and unclear trends, which may indicate the presence of noise. The noise in serials 1–13 can reduce the model's prediction accuracy. To address this, EWMA smoothing is applied to serials 1–13. The EWMA is given using Formula (1):

$$\begin{cases} \alpha = \frac{2}{span+1} \\ E_{(t)} = \alpha \times X_{(t)} + (1 - \alpha) \times E_{(t-1)} \end{cases} \quad (1)$$

where *span* is the span parameter that determines the value of  $\alpha$ ,  $X_{(t)}$  represents the current observation value at time  $t$ ,  $E_{(t-1)}$  is the exponential moving average value at the previous time point ( $t - 1$ ), and  $E_{(t)}$  is the exponential moving average value for the current time point  $t$ .

In Formula (1),  $\alpha$  typically ranges between 0 and 1. When  $\alpha$  is larger, the smoothed sequence adapts more quickly to recent trends and fluctuations but retains less long-term trend information. Conversely, when  $\alpha$  is smaller, the smoothed sequence retains more long-term trend information but adapts more slowly to recent trends and fluctuations. For the specific implementation, using data points from serial number 2 as an example, by setting *span* = 10 and an initial value  $E_{(t)}$ ,  $\alpha$  is determined. The observation value  $X_{(2)}$  is then substituted into the formula to get  $E_{(2)}$ . This process is iteratively applied to achieve data smoothing, as illustrated in Figure 4.

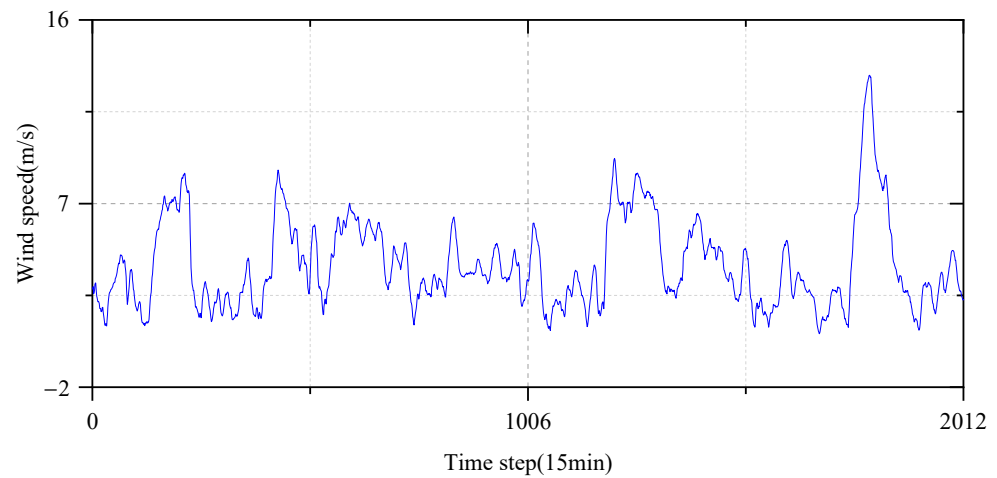


Figure 4. Data points for serial number 2 after EWMA smoothing.

In Figure 4, the smoothed curve is more stable compared to that in Figure 3. By applying EWMA smoothing, noise is suppressed, and the original data trends are retained. The architecture of the EWMA data processing method is shown in Figure 5.

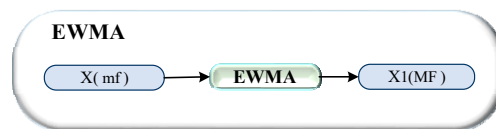


Figure 5. Workflow of EWMA.

In Figure 5,  $X(mf)$  represents the original dataset, where *mf* denotes the concatenation of the introduced multivariable feature serials and the wind power feature serial. The multivariable feature serials within *mf* are processed using EWMA and then concatenated with the original wind power feature serial to form the multivariable channel data *MF*. Through the EWMA process, the original dataset  $X(mf)$  is transformed into a new dataset  $X_1(MF)$ .

### 2.2.2. $X_{en}$ and $X_{de}$

The sliding window method is applied to extract samples from  $X_1(MF)$ , resulting in the Encoder input  $X_{en}(mf1)$ . Additionally, the Informer model’s input partitioning method is used to obtain the Decoder input. The partitioning method of the Informer model is illustrated in Figure 6.

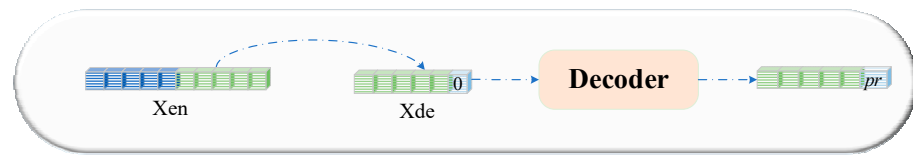


Figure 6. Partitioning method of the Informer model.

In Figure 6, the historical data from the green area of  $X_{en}(mf1)$  are concatenated with data of value 0 to obtain  $X_{de}(mf2)$ . The Decoder layer then transforms the 0 values into predicted values  $pr$ .

### 2.2.3. Embedding Layer

The Embedding layer primarily consists of convolutional neural network multi-channel (CNN-MC) Embedding and positional encoding (PE) Embedding. To enhance the capability of extracting local features, the CNN-MC Embedding uses one-dimensional convolution for dimensional expansion. The CNN-MC Embedding module is illustrated in Figure 7.

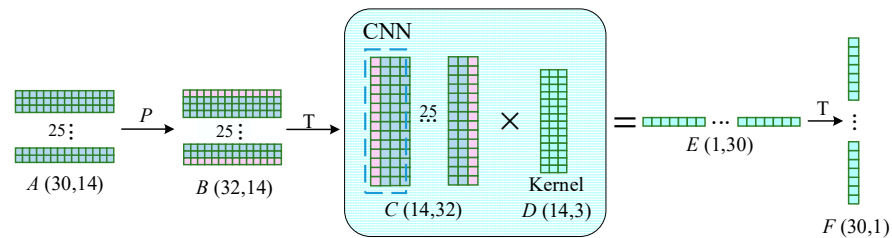


Figure 7. CNN-MC Embedding model.

In Figure 7, dark green represents the original data, pink represents the padding data, and light green represents the convolutional data, for example, a matrix  $X_{en}$  with dimensions (30, 14) is input, where 30 represents the number of data points, and 14 represents the number of features. Padding ( $P$ ) is applied to  $X_{en}$ , resulting in a matrix  $B$  with dimensions (32, 14). The matrix  $B$  is then transposed ( $T$ ), yielding a matrix  $C$  with dimensions (14, 32). Matrix  $C$  is fed into the CNN module, which uses a convolutional kernel  $D$  with dimensions (14, 3). Convolution is performed on  $C$  with a stride of 1 from left to right, resulting in an output matrix  $E$  with dimensions (1, 30). Matrix  $E$  is then transposed ( $T$ ) to produce a matrix  $F$  with dimensions (30, 1), thereby incorporating local information into each data point of  $X_{en}$ . It is important to note the following two points: First, the process described involves using a single convolutional kernel. To expand the dimension of the Embedding layer to 512, 512 convolutional kernels need to be set. Consequently, the final output of the CNN-MC Embedding layer should be (30, 512). Second, Formula (2) should be used to ensure that the number of data points in matrix  $F$  matches the number of data points in matrix  $X_{en}$ .

$$L = H + P - K_n + 1 \tag{2}$$

where  $H$  is the number of rows in the original input matrix,  $P$  is the padding dimension, and  $K_n$  is the width of the convolutional kernel.

PE Embedding creates a zero matrix with the same dimensions as the output of CNN-MC Embedding. The odd columns of the zero matrix are assigned positional information using the cosine function from Formula (3), while the even columns use the sine function from Formula (3) to assign positional information. Finally, the CNN-MC Embedding and PE Embedding are added together to obtain the final output of the Embedding layer, with the last dimension remaining at 512.

$$\begin{cases} PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \end{cases} , \tag{3}$$



where  $pos$  represents the position,  $i$  represents the dimension, and  $d_{model}$  represents the expanded dimension.

### 2.2.4. Encoder Layer

The flow of the Encoder layer is illustrated in Figure 8. The output of the new Embedding layer is fed into three linear layers to obtain  $Q$ ,  $K$ , and  $V$ . Multi-head attention mechanism calculations are performed using Formula (4) to establish temporal correlations and extract relevant features through weighting. The output then passes through an Add & Norm layer, followed by a feedforward neural network composed of CNN-Gelu-CNN for nonlinear transformation, enhancing the model’s ability to capture nonlinear relationships in the data. Finally, another Add & Norm layer processes the data to produce the output of the Encoder layer. It is important to note that the last dimension of the Encoder layer’s input and output remains unchanged at 512.

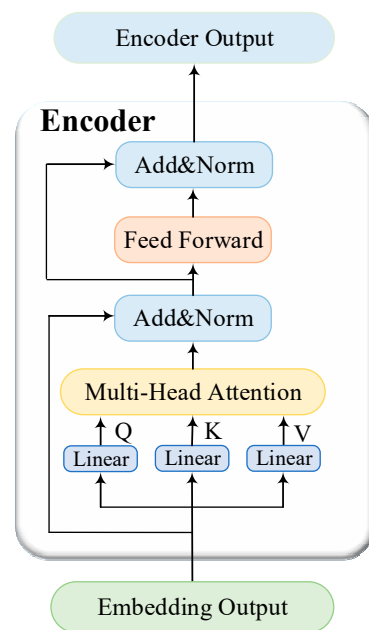


Figure 8. Workflow of the Encoder.

The multi-head attention mechanism algorithm is described using Formula (4):

$$\begin{cases} Score_{(w)} = \text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) , \\ C = Score_{(w)} \times V \end{cases} \quad (4)$$

where  $Q$  represents the query matrix,  $K$  represents the key matrix,  $V$  represents the value matrix, and  $\sqrt{d_k}$  represents the scaling factor for the attention weight range.

### 2.2.5. Decoder Layer

The flow of the Decoder is illustrated in Figure 9. The output from the Embedding layer is passed through three linear layers to obtain  $Q$ ,  $K$ , and  $V$ . To prevent information leakage during inference, a masked multi-head attention mechanism is used to compute correlations, and the output is then fed into an Add & Norm layer. A linear layer generates the query matrix  $Q'$ , while the Encoder’s output is processed through two linear layers to obtain  $K'$  and  $V'$ . The matrices  $Q'$ ,  $K'$ , and  $V'$  are input into the multi-head attention mechanism to facilitate the interaction between the Encoder and Decoder. The output then passes through an Add & Norm layer, a feedforward neural network, and another Add & Norm layer. It is important to note that the last dimension of the Decoder layer’s input and output remains unchanged at 512.



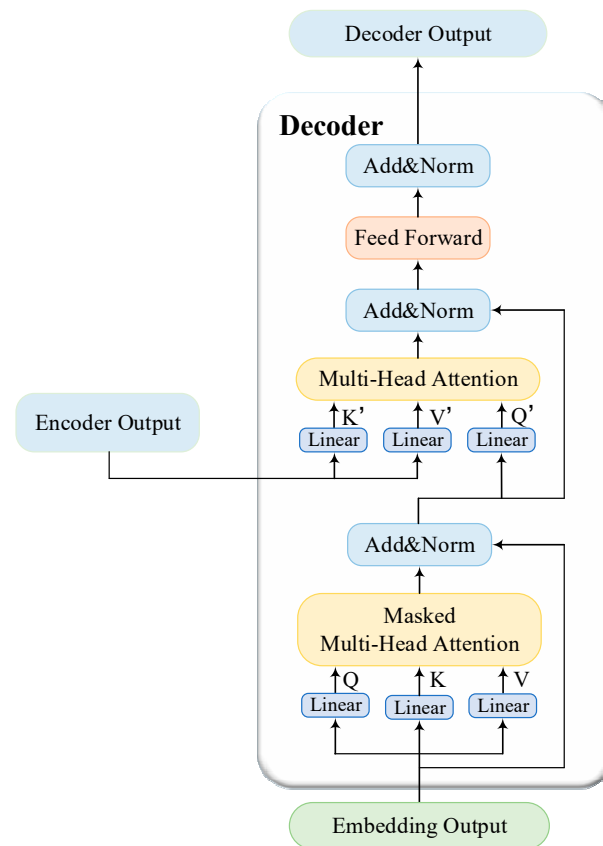


Figure 9. Workflow of Decoder.

### 2.2.6. KAN Layer

In Transformer models and their variants, the final prediction results are typically obtained through a simple linear layer for dimensionality reduction. Using self-learned nonlinear unit functions instead of a simple linear layer, the model can better capture nonlinear relationships and improve prediction accuracy. To address this, the KAN network, which is based on the KART mathematical principle, is introduced. The principle is described using Formula (5).

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left( \sum_{p=1}^n \phi_{q,p}(x_p) \right), \tag{5}$$

where  $q$  represents the number of unit functions,  $p$  represents the number of input units,  $\phi_{q,p}(x_p)$  represents the unary function of the first layer in the KART network, and  $\Phi_q$  represents the unary function of the second layer in the KART network.

Based on Formula (5), it can be concluded that a multivariate function can be obtained by summing a finite number of nonlinear unit functions. In this process, if  $n = 2$ , then the first layer of the KART network will have a fixed number of nodes, specifically  $(2n + 1) = 5$  nodes. The KAN network is a variant of this formula, where the fixed number of  $(2n + 1)$  nodes can be manually set. For ease of analysis, the basic architecture of the KAN network, using the example of predicting a future wind power value, is illustrated in Figure 10.

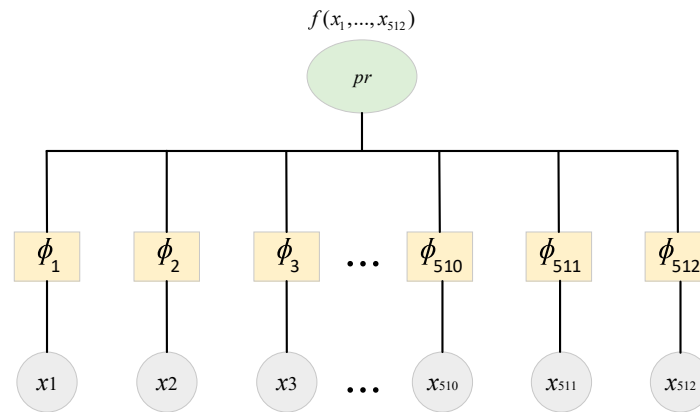


Figure 10. KAN network infrastructure.

In Figure 10, the 512 outputs from the Decoder, denoted as  $x_1$  to  $x_{512}$ , are mapped to 512 unit functions  $\phi_1 \sim \phi_{512}$ . These unit functions are then summed to obtain the multivariate function  $f(x_1, \dots, x_{512})$ , which is used to derive the final prediction result  $pr$ . The core of the KAN network is to learn each of these nonlinear unit functions. The learning process is described using Formula (6):

$$\begin{cases} \phi(x) = w(b(x) + \text{spline}(x)) \\ b(x) = \text{silu}(x) = x / (1 + e^{-x}) \\ \text{spline}(x) = \sum c_i B_i(x) \end{cases}, \tag{6}$$

where  $w$  represents the learnable parameters,  $\text{spline}(x)$  represents the B-spline function, and  $c_i$  represents the learnable parameters.

In Formula (6), each nonlinear unit function  $\phi(x)$  is defined by adding a B-spline function  $\text{spline}(x)$  to the  $\text{silu}(x)$  function and then scaling it by a learnable parameter  $w$  to ensure smoothness of  $\phi(x)$ . The most important part is  $\text{spline}(x)$ , which is the B-spline function; the KAN network uses a cubic B-spline.

To facilitate a clear explanation of the process of learning the nonlinear unit function, the code for the most important component,  $\text{spline}(x)$ , in the KAN network is shown in Figure 11. Additionally, the learning process of  $\text{spline}(x)$  and other components within the KAN network is elaborated in detail, as illustrated in Figure 12.

```
def b_splines(self, x: torch.Tensor):
    assert x.dim() == 2 and x.size(1) == self.in_features

    grid: torch.Tensor = ( # shape (in_features, grid_size + 2 * spline_order + 1)
        self.grid
    ) # (in_features, grid_size + 2 * spline_order + 1)
    x = x.unsqueeze(-1)
    bases = ((x >= grid[:, :-1]) & (x < grid[:, 1:])).to(x.dtype)
    for k in range(1, self.spline_order + 1):
        bases = (
            (x - grid[:, :-(k + 1)])
            / (grid[:, k:-1] - grid[:, :-(k + 1)])
            * bases[:, :, :-1]
        ) + (
            (grid[:, k + 1:] - x)
            / (grid[:, k + 1:] - grid[:, 1:(-k)])
            * bases[:, :, 1:]
        )

    assert bases.size() == (
        x.size(0),
        self.in_features,
        self.grid_size + self.spline_order,
    )
    return bases.contiguous()
```

Figure 11. spline(x) code writing process.

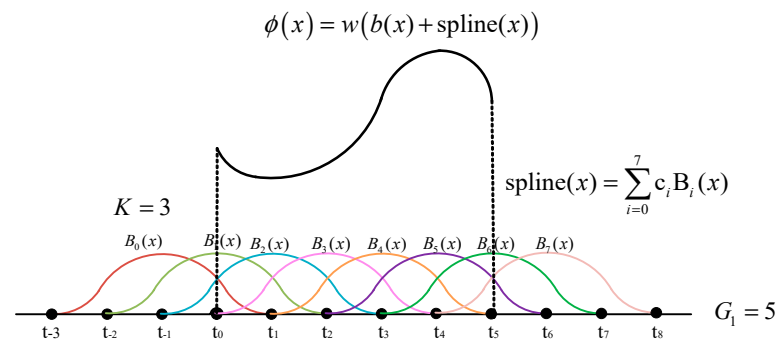


Figure 12. Learning process of nonlinear unit functions.

In Figure 12,  $b(x)$  is a fixed silu function and does not require learning, while  $w$  is a learnable parameter used for scaling. The most critical aspect of the figure is the learning process of  $\text{spline}(x)$ . The process is as follows: First,  $G_1 = 5$  is set, which indicates that there are five grid points where the nonlinear unit function will be formed. Second, seven B-spline functions are initialized with  $K = 3$ . By learning the weights  $c_i$  of these seven B-spline functions, one can control the shape of each B-spline function. These seven learned B-spline functions are summed to obtain  $\text{spline}(x)$ . Then, a smooth curve  $\text{silu}(x)$  is added to  $\text{spline}(x)$  to enhance the smoothness of the unit function. Finally,  $\{\text{spline}(x) + \text{silu}(x)\}$  is multiplied by  $w$  to obtain the final unit function  $\phi(x)$ . It is important to note that the process described is for learning a single unit function. In practical applications, one needs to map the 512 outputs of the Decoder layer to 512 nonlinear unit functions, each learnable. Summing these 512 learned unit functions will yield the final prediction result  $pr$

### 3. Experimental Analysis

#### 3.1. Evaluation Metrics

In the experimental analysis of this paper, the accuracy of the model predictions is measured using three metrics: RMSE, MAE, and  $R^2$ . The calculation formula for RMSE is presented in Equation (7):

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}, \tag{7}$$

where  $Y_i$  represents the actual value of the data,  $\hat{Y}_i$  represents the predicted value output by the model, and  $n$  denotes the total number of prediction samples.

The calculation formula for MAE is presented in Equation (8):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|. \tag{8}$$

The coefficient of determination, denoted as  $R^2$ , is an important metric for assessing the goodness of fit of a time series forecasting model. It indicates the proportion of variance between the predicted values and the actual observed values. The specific calculation method for  $R^2$  is presented in Equation (9):

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y}_i)^2}, \tag{9}$$

where  $\bar{Y}_i$  represents the mean value.

### 3.2. Data and Simulation Settings

This experiment uses data from the 200 MW wind farm, as described in Section 2. To predict the future wind power value based on 30 historical data points, a sliding window method is used to extract samples. The dataset is divided as follows: Training set: 24,498 samples, Validation set: 7008 samples, and Test set: 3504 samples. The model is built and trained using Python 3.9.7 programming language and the PyTorch-gpu 2.1.0 deep learning framework. The hyperparameters for the model are set as shown in Table 2.

**Table 2.** Hyperparameter configuration.

Hyperparameter	Parameter Values
Batch_size	32
Train_epochs	15
Kenel_size	3
D_model	512
H_heads	8
Encoder_layers	2
Decoder_layers	1
Dropout	0.05
Learning_rate	Adaptive Optimization

The test set contains a total of 3504 samples. To clearly present the prediction results, samples are selected at intervals for plotting. MAE, RMSE, and  $R^2$  are used to evaluate and analyze the prediction results of the 3504 test samples. The experimental analysis consists of two parts. The first part is model optimization verification. First, the Transformer model is enhanced with the KAN module to create the KAN-Transformer model. This part validates whether the KAN layer improves the model's prediction accuracy by capturing nonlinear relationships. Next, the original data are processed using EWMA to create the EWMA-KAN-Transformer model. This part verifies whether applying EWMA denoising to the introduced multivariable features enhances the model's prediction accuracy. The second part is a comparison with common time series prediction models. The proposed model is compared with common time series prediction models in terms of prediction curves and test set evaluation metrics. This part assesses whether the proposed model shows advantages over other models in the field of ultrashort-term wind power prediction.

### 3.3. Optimization Analysis of Transformer Model Based on EWMA and KAN

The KAN network is a novel deep-learning algorithm that captures nonlinear relationships by introducing learnable activation functions at the network's edges [26,27]. It has significant advantages in handling complex function fitting and nonlinear patterns, and it has been demonstrated to be an effective method for improving accuracy in the field of time series forecasting [28].

To verify that the KAN network can effectively improve the prediction accuracy of the Transformer model, we replace the final linear layer of the Transformer model with a KAN layer. This modification improves the model's ability to capture nonlinear relationships. The comparison of prediction curves between the Transformer model and the KAN-Transformer model is shown in Figure 13, and the comparison of test set prediction evaluation metrics is shown in Table 3.

Data from 3 days are selected for plotting in Figure 13, and Table 3 shows the results of calculations for the entire test set data. In Figure 13, the orange curve represents the prediction of the Transformer model, while the green curve represents the prediction of the KAN-Transformer model. Table 3 indicates that the KAN-Transformer model reduces MAE and RMSE by 0.41 MW and 0.49 MW, respectively, and increases  $R^2$  by 0.18% compared to the Transformer model. This demonstrates that the KAN layer's ability to self-learn non-linear functions results in higher prediction accuracy than the Transformer model's linear mapping, highlighting the importance of capturing non-linear relationships.

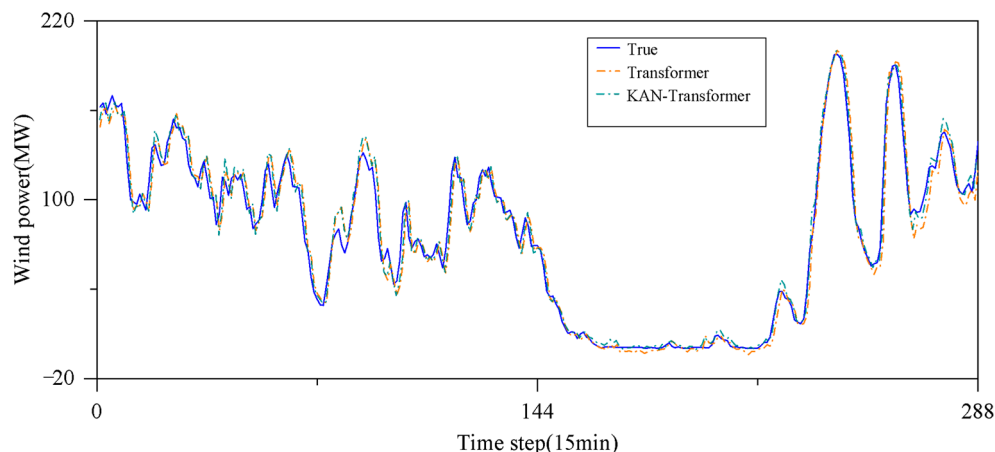


Figure 13. Comparison of prediction curves between Transformer and KAN-Transformer models.

Table 3. Comparison of test set evaluation metrics between Transformer and KAN-Transformer models.

Model	MAE/MW	RMSE/MW	R <sup>2</sup> /%
Transformer	4.87	8.21	98.43
KAN-Transformer	4.46	7.72	98.61

EWMA, with its exponential weighting average characteristic, helps to smooth time series data and can effectively address issues related to noise and outliers in the data [29,30], which is particularly important in applications such as time series forecasting. In the context of wind power forecasting, noise from external variables like wind speed and wind direction can directly impact the accuracy of power predictions. By applying EWMA smoothing, we can preprocess these external variables, providing the model with more stable feature inputs. This paper first introduces the KAN network based on the Transformer time series forecasting model and then combines it with EWMA to suppress noise in the data. To validate that the combination of the two achieves the highest prediction accuracy, we compare the KAN-Transformer with the EWMA-KAN-Transformer model. The comparison of prediction curves between the KAN-Transformer model and the EWMA-KAN-Transformer model is shown in Figure 14, and the evaluation metrics for the test set are compared in Table 4.

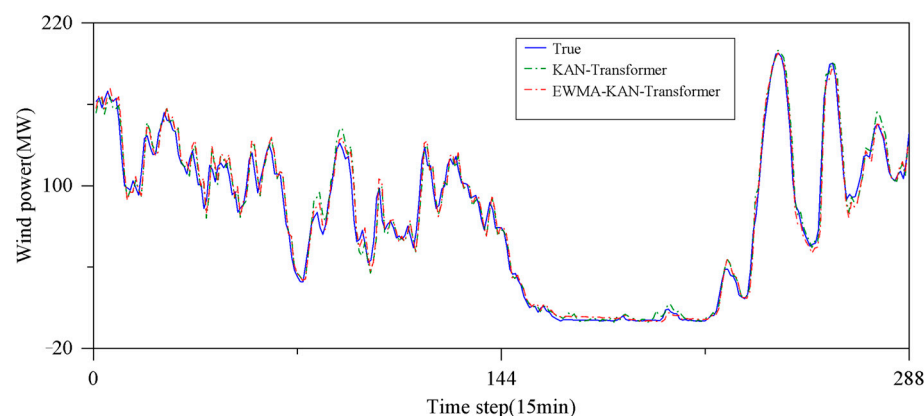


Figure 14. Comparison of prediction curves between the KAN-Transformer model and the EWMA-KAN-Transformer model.

**Table 4.** Comparison of test set evaluation metrics between the KAN-Transformer model and the EWMA-KAN-Transformer model.

Model	MAE/MW	RMSE/MW	R <sup>2</sup> /%
KAN-Transformer	4.46	7.72	98.61
EWMA-KAN-Transformer	4.38	7.37	98.73

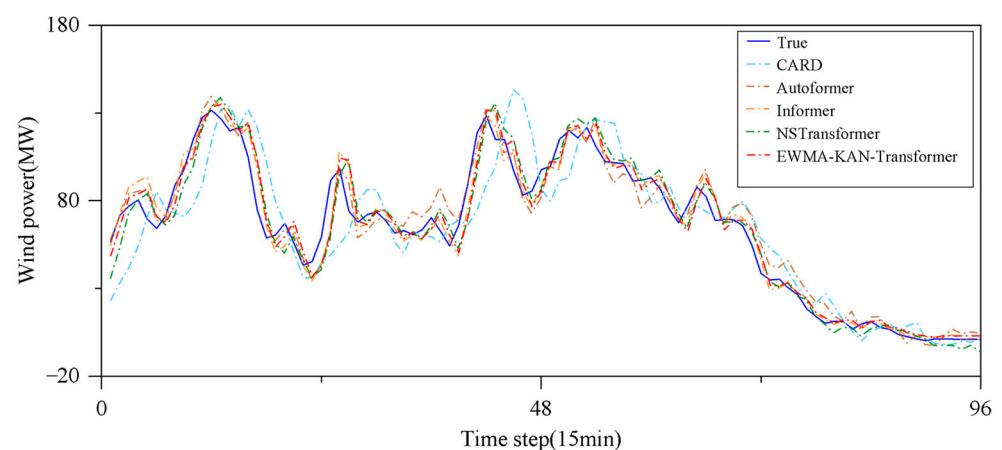
Data from 3 days is selected in Figure 14 for plotting, and Table 4 shows the results calculated from the entire test set data. In Table 4, the KAN-Transformer model has a reduction in MAE and RMSE by 0.08 MW and 0.35 MW, respectively, and an increase in R<sup>2</sup> by 0.12% compared to the EWMA-KAN-Transformer model. This confirms that applying EWMA for denoising when introducing multiple variable features improves the model's prediction accuracy for future wind power values. From Tables 3 and 4, it can be seen that the EWMA-KAN Transformer model reduces MAE and RMSE by 0.49 MW and 0.84 MW, respectively, and increases R<sup>2</sup> by 0.3% compared to the Transformer model.

### 3.4. Comparison and Analysis of the Proposed Model with Other Models

The EWMA data-processed KAN-Transformer short-term wind power forecasting model is compared with Transformer variant models, traditional machine learning algorithms, and recurrent neural network variants such as LSTM and GRU series models to verify the advantages of the proposed model in the field of short-term wind power time series forecasting. During the verification process, data from day 1 are drawn and selected, and the results are analyzed in the table as the computed results from the entire test set data.

#### 3.4.1. Comparison of the Proposed Model with Transformer Variant Models

The proposed model is compared with CARD, Autoformer, Informer, NSTransformer, and Autoformer models. The comparison of model prediction curves is shown in Figure 15, and the evaluation metrics for the test set are compared in Table 5.

**Figure 15.** Comparison of prediction curves of Transformer variant models and EWMA-KAN-Transformer model.

In Figure 15, red represents the model proposed in this article, and the blue solid line represents the actual wind power data. From Figure 15, it is evident that the model proposed in this article has the highest degree of fitting with real wind power data in terms of prediction results. In Table 5, the EWMA-KAN-Transformer model shows a reduction in MAE and RMSE by 4.05 MW and 7.96 MW and an increase in R<sup>2</sup> by 4.19% compared to the CARD model. Compared to the Autoformer model, the MAE and RMSE are reduced by 1.42 MW and 1.48 MW, and R<sup>2</sup> is increased by 0.55%. Compared to the Informer model, the MAE and RMSE are reduced by 0.38 MW and 1.30 MW, and R<sup>2</sup> is increased by 0.48%. Compared to the NSTransformer model, the MAE and RMSE are reduced by 0.05 MW

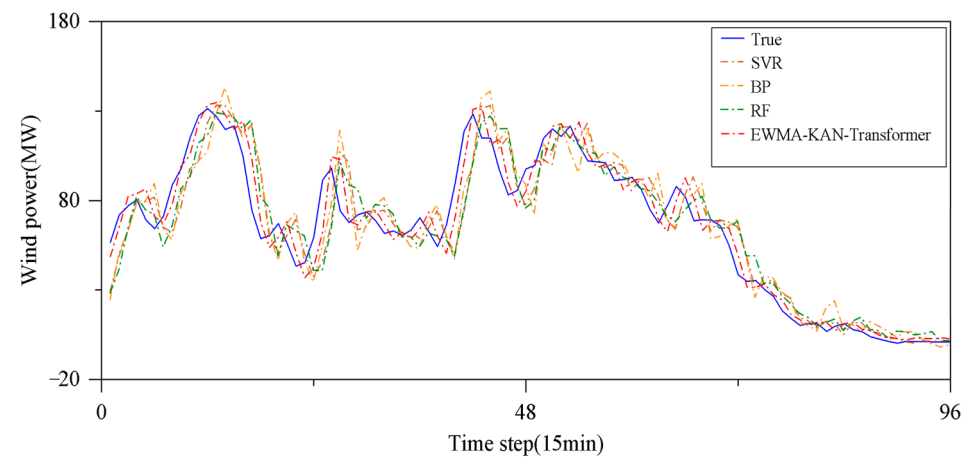
and 0.79 MW, and  $R^2$  is increased by 0.28%. This demonstrates that the proposed model achieves the highest prediction accuracy among the four Transformer variant models.

**Table 5.** Comparison of test set evaluation metrics between the Transformer variant models and the EWMA-KAN-Transformer model.

Model	MAE/MW	RMSE/MW	$R^2/\%$
CARD	8.43	15.33	94.54
Autoformer	5.80	8.85	98.18
Informer	4.76	8.67	98.25
NSTransformer	4.43	8.16	98.45
EWMA-KAN-Transformer	4.38	7.37	98.73

### 3.4.2. Comparison of the Proposed Model with Traditional Machine Learning Algorithms

In machine learning-based time series forecasting algorithms, SVR can handle the nonlinear issues in the data, BP effectively trains deep networks using the backpropagation algorithm, and RF enhances prediction robustness by aggregating multiple decision trees. Therefore, these three methods are commonly used for wind power forecasting [32–34]. The model proposed in this paper is a time series forecasting model based on deep learning, aiming to suppress noise and enhance the model's ability to capture nonlinear relationships through EWMA and KAN. Compared to traditional algorithms, this model can automatically learn deep features of the data, making it superior when dealing with high-dimensional and nonlinear characteristics while also exhibiting greater generalization capability. To verify the superiority of the proposed model over traditional machine learning algorithms, it is compared with traditional machine learning algorithms such as SVR, BP, and RF. The comparison of model prediction curves is shown in Figure 16, and the test set evaluation metrics are shown in Table 6.



**Figure 16.** Comparison of prediction curves between the EWMA-KAN-Transformer model and traditional machine learning algorithms.

**Table 6.** Comparison of test set evaluation metrics between the EWMA-KAN-Transformer model and traditional machine learning algorithms.

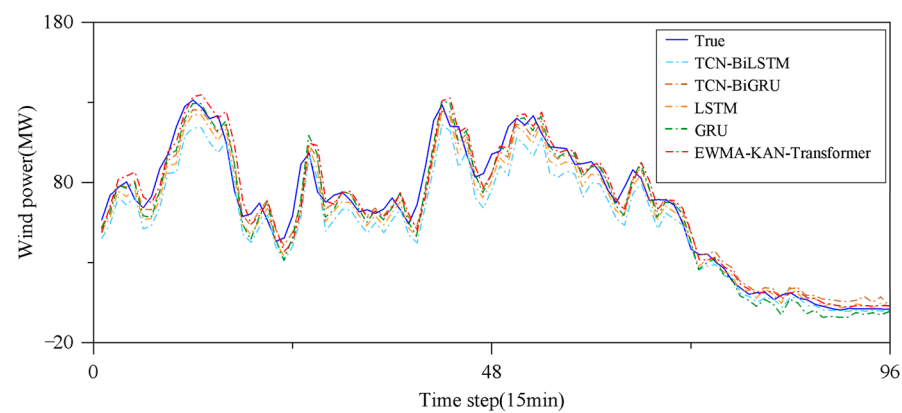
Model	MAE/MW	RMSE/MW	$R^2/\%$
SVR	4.55	17.52	92.88
BP	7.00	12.86	96.16
RF	4.55	8.47	98.33
EWMA-KAN-Transformer	4.38	7.37	98.73



From Figure 16, it is evident that the model proposed in this paper has the best prediction accuracy compared to traditional machine learning algorithms. In Table 6, the proposed EWMA-KAN-Transformer model shows improvements compared to the SVR model; MAE and RMSE are reduced by 0.17 MW and 10.15 MW, respectively, while  $R^2$  increases by 5.85%. Compared to the BP model, MAE and RMSE are reduced by 2.62 MW and 5.49 MW, respectively, with an increase of 2.57% in  $R^2$ . Compared to the RF model, MAE and RMSE are reduced by 0.17 MW and 1.10 MW, respectively, and  $R^2$  increases by 0.40%. Thus, the proposed model demonstrates the highest prediction accuracy compared to the four traditional machine learning algorithms.

### 3.4.3. Comparison Between the Proposed Model and LSTM and GRU Series Models

The proposed model is compared with the TCN-BiLSTM, TCN-BiGRU, LSTM, and GRU models. The comparison of the model prediction curves is shown in Figure 17, and the evaluation metrics for the test set are shown in Table 7.



**Figure 17.** Comparison of prediction curves between the EWMA-KAN-Transformer model and LSTM and GRU series models.

**Table 7.** Comparison of test set evaluation metrics between the EWMA-KAN-Transformer model and LSTM and GRU series models.

Model	MAE/MW	RMSE/MW	$R^2$ /%
TCN-BiLSTM	8.54	13.15	95.97
TCN-BiGRU	6.66	8.96	98.13
LSTM	5.57	8.56	98.29
GRU	6.24	8.36	98.37
EWMA-KAN-Transformer	4.38	7.37	98.73

From Figure 17, it can be observed that the model proposed in this article has the highest prediction accuracy. In Table 7, the EWMA-KAN-Transformer model shows the following improvements compared to other models. Compared to the TCN-BiLSTM model, the MAE and RMSE are reduced by 4.16 MW and 5.78 MW, respectively, and  $R^2$  increases by 2.76%. Compared to the TCN-BiGRU model, the MAE and RMSE are reduced by 2.28 MW and 1.59 MW, respectively, and  $R^2$  increases by 0.60%. Compared to the LSTM model, the MAE and RMSE are reduced by 1.19 MW and 1.19 MW, respectively, and  $R^2$  increases by 0.44%. Compared to the GRU model, the MAE and RMSE are reduced by 1.86 MW and 0.99 MW, respectively, and  $R^2$  increases by 0.36%. This demonstrates that the proposed model achieves higher prediction accuracy compared to the GRU and LSTM series models.

## 4. Conclusions

In order to enhance grid dispatch flexibility, improve the absorption capacity of renewable energy, reduce wind curtailment, and strengthen grid stability, this paper proposes a

short-term wind power forecasting model based on the EWMA-KAN-Transformer. This paper addresses the issues of data noise and the limited ability of the model to capture nonlinear relationships due to the use of a simple linear layer in the final prediction step. Firstly, EWMA data processing is applied to reduce the impact of noise from the introduced multivariate features on the model's prediction accuracy. Secondly, the KAN (Kernelized Attention Network) is integrated into the Transformer model. By learning the parameters of cubic B-splines, non-linear unit functions are obtained. These non-linear unit functions are then summed to form a multi-dimensional continuous function, which is used to generate the final prediction, thus enhancing the model's ability to capture non-linear relationships. Finally, experimental analysis leads to the following conclusions.

1. The KAN-Transformer model reduced the MAE and RMSE by 0.41 MW and 0.49 MW, respectively, compared to the Transformer model. Additionally, the  $R^2$  improved by 0.18%. This validates that incorporating the KAN network in the final layer of the Transformer model enhances its predictive accuracy by capturing nonlinear relationships.
2. By applying EWMA to the original wind power data to remove some of the noise, the EWMA-KAN-Transformer model reduced the MAE and RMSE by 0.08 MW and 0.35 MW, respectively, compared to the KAN-Transformer model. Additionally, the  $R^2$  improved by 0.12%. This demonstrates that when incorporating multiple variables, using EWMA to remove noise can mitigate the impact of noise in wind power data on the model's predictive accuracy.
3. When comparing the proposed model with three types of models, the EWMA-KAN-Transformer model demonstrated the highest prediction accuracy, with MAE and RMSE values of 4.38 MW and 7.37 MW, respectively, and an  $R^2$  value of 98.73%. This confirms that the proposed model is more advantageous compared to other wind power forecasting models.
4. The proposed model uses EWMA for noise reduction, which has the advantage of denoising without altering the data trend by adjusting the decay factor. However, finding the optimal decay factor requires ongoing experimentation to enhance the model's prediction accuracy. In future research, employing a rational optimization method to determine the most suitable decay factor could potentially further improve the model's prediction accuracy.

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

**Funding:** This research was in part supported by the 2024 Fundamental Research Project (No. LJ212410154011) of the Educational Department of Liaoning Province, in part by National Natural Science Foundation of China (No. 52406079), in part by the Start-up Funding for Newly Introduced Talents in Shenzhen (CA11409031), and in part by Guangdong Science and Technology Department through Guangdong-Hong Kong-Macao Joint Innovation Program (No. 2024A0505040006).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The original contributions presented in this study are included in this article; further inquiries can be directed to the corresponding author.

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

## Nomenclature

### Abbreviations

ADAM	Adaptive moment estimation method
ARIMA	Auto-regressive integrated moving average
ARMA	Auto-regressive and moving average
BP	Backpropagation
CEEMD	Complementary ensemble empirical mode decomposition
CNN-MC	Convolutional neural network multi-channel
EWMA	Exponential weighted moving average
GRU	Gated recurrent units
KAN	Kolmogorov–Arnold networks
KART	Kolmogorov–Arnold representation theorem
LSTM	Long short-term memory networks
MAE	Mean absolute error
MSE	Mean squared error
NWP	Numerical weather prediction
PE	Positional encoding
RMSE	Root mean squared error

### Symbols

$c_i$	The learnable parameters
$d_{\text{model}}$	The expanded dimension
$E_{(t-1)}$	The exponential moving average value at the previous time point ( $t - 1$ )
$E_{(t)}$	The exponential moving average value for the current time point $t$
$H$	The number of rows in the original input matrix
$i$	Dimension
$K$	Key matrix
$K_n$	The width of the convolutional kernel
$n$	The sample size
$p$	The number of input units
$P$	The padding dimension
$pr$	Model prediction results
$pos$	Position
$q$	The number of unit functions
$Q$	Query matrix
$R^2$	The coefficient of determination
$spline(x)$	The b-spline function
$V$	Value matrix
$w$	The learnable parameters
$X_{(t)}$	The current observation value at time $t$
$Y_i$	The actual values
$\hat{Y}_i$	The predicted values
$\phi_{q,p}(x_p)$	The unary function of the first layer in the KART network
$\phi_q$	The unary function of the second layer in the KART network

## References

- Cheng, L.L.; Zang, H.X.; Xu, Y.; Wei, Z.N.; Sun, G.Q. Augmented Convolutional Network for Wind Power Prediction: A New Recurrent Architecture Design with Spatial-Temporal Image Inputs. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6981–6993. [[CrossRef](#)]
- An, J.Q.; Yin, F.; Wu, M.; She, J.H.; Chen, X. Multisource Wind Speed Fusion Method for Short-Term Wind Power Prediction. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5927–5937. [[CrossRef](#)]
- Ran, M.H.; Huang, J.D.; Qian, W.Y.; Zou, T.T.; Ji, C.Y. EMD-based gray combined forecasting model—Application to long-term forecasting of wind power generation. *Heliyon* **2023**, *9*, e18053. [[CrossRef](#)] [[PubMed](#)]
- Zhang, W.J.; Di, L.; Miao, G.X.; Liu, W.; Hua, Y.P.; Shen, L.L.; Wang, Q.; Zhu, G.R. Research on Medium and Long-Term Electrical Power Prediction of Wind Farm Based on GA-BP Algorithm. In Proceedings of the 2023 9th International Conference on Electrical Engineering, Control and Robotics (EECR), Wuhan, China, 24–26 February 2023; pp. 110–114.
- Yang, Z.M.; Peng, X.S.; Song, J.J.; Duan, R.Q.; Jiang, Y.; Liu, S.Q. Short-Term Wind Power Prediction Based on Multi-Parameters Similarity Wind Process Matching and Weighed-Voting-Based Deep Learning Model Selection. *IEEE Trans. Power Syst.* **2024**, *39*, 2129–2142. [[CrossRef](#)]
- Li, H.C.; Liu, L.Q.; He, Q.S. A Spatiotemporal Coupling Calculation-Based Short-Term Wind Farm Cluster Power Prediction Method. *IEEE Access* **2023**, *11*, 131418–131434. [[CrossRef](#)]

7. Xing, F.; Song, X.Y.; Wang, Y.B.; Qin, C.Y. A New Combined Prediction Model for Ultra-Short-Term Wind Power Based on Variational Mode Decomposition and Gradient Boosting Regression Tree. *Sustainability* **2023**, *15*, 11026. [[CrossRef](#)]
8. Sun, Y.; Yang, J.J.; Zhang, X.T.; Hou, K.Y.; Hu, J.Y.; Yao, G.Z. An Ultra-Short-Term Wind Power Forecasting Model Based on EMD-EncoderForest-TCN. *IEEE Access* **2024**, *12*, 60058–60069. [[CrossRef](#)]
9. Zhang, J.H.; Wang, Y.Y.; Zhou, G.P.; Wang, L.; Li, B.; Li, K. Integrating Physical and Data-Driven System Frequency Response Modelling for Wind-PV-Thermal Power Systems. *IEEE Trans. Power Syst.* **2024**, *39*, 217–228. [[CrossRef](#)]
10. Ashraf, M.; Raza, B.; Arshad, M.; Ahmed, A.; Zaidi, S.S.H. A Hybrid Statistical Model for Ultra Short Term Wind Speed Prediction. In Proceedings of the 2023 7th International Multi-Topic ICT Conference (IMTIC), Jamshoro, Pakistan, 10–12 May 2023; pp. 1–8.
11. Li, M.L.; Yang, M.; Yu, Y.X.; Li, P.; Wu, Q.W. Short-Term Wind Power Forecast Based on Continuous Conditional Random Field. *IEEE Trans. Power Syst.* **2024**, *39*, 2185–2197. [[CrossRef](#)]
12. Zhu, X.T.; Chen, D.J.; Wang, L. Research on Wind Power Prediction Based on EMD-AttLSTM-ARMA. In Proceedings of the 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 26–28 May 2023; pp. 904–909.
13. Kumari, S.; Sreekumar, S.; Singh, S.; Kothari, D.P. Comparison Among ARIMA, ANN, and SVR Models for Wind Power Deviation Charge Reduction. In Proceedings of the 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India, 26–27 May 2022; pp. 551–557.
14. Li, M.L.; Yang, M.; Yu, Y.X.; Lee, W.-J. A Wind Speed Correction Method Based on Modified Hidden Markov Model for Enhancing Wind Power Forecast. *IEEE Trans. Ind. Appl.* **2022**, *58*, 656–666. [[CrossRef](#)]
15. Zhang, H.T.; Zhu, Q.H.; Li, W.J.; Li, X.F.; Xie, C.Q.; Xiang, C.Y.; Fan, J.H. Ultra Short Term Power Forecasting of Wind Generation Based on Improved LSTM. In Proceedings of the 2023 3rd Power System and Green Energy Conference (PSGEC), Shanghai, China, 24–26 August 2023; pp. 176–180.
16. Gao, W.B.; Wang, G.X.; Liang, X.L.; Hu, Z.W. A STAM-LSTM model for wind power prediction with feature selection. *Energy* **2024**, *296*, 131030.
17. Yu, G.Z.; Lu, L.; Tang, B.; Wang, S.Y.; Chung, C.Y. Ultra-Short-Term Wind Power Subsection Forecasting Method Based on Extreme Weather. *IEEE Trans. Power Syst.* **2023**, *38*, 5045–5056. [[CrossRef](#)]
18. Li, C.S.; Tang, G.; Xue, X.M.; Saeed, A.; Hu, X. Short-Term Wind Speed Interval Prediction Based on Ensemble GRU Model. *IEEE Trans. Sustain. Energy* **2020**, *11*, 1370–1380. [[CrossRef](#)]
19. Zhu, Y.H.; Sun, X.Y.; Wang, M.; Huang, H. Multi-Modal Feature Pyramid Transformer for RGB-Infrared Object Detection. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9984–9995. [[CrossRef](#)]
20. Ramana, K.; Srivastava, G.; Kumar, M.R.; Gadekallu, T.R.; Lin, J.C.W.; Alazab, M.; Iwendi, C. A Vision Transformer Approach for Traffic Congestion Prediction in Urban Areas. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3922–3934. [[CrossRef](#)]
21. Hong, D.F.; Han, Z.; Yao, J.; Gao, L.R.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking Hyperspectral Image Classification with Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5518615. [[CrossRef](#)]
22. Deng, B.X.; Wu, Y.F.; Liu, S.J.; Xu, Z.L. Wind Speed Forecasting for Wind Power Production Based on Frequency-Enhanced Transformer. In Proceedings of the 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Shanghai, China, 28–30 October 2022; pp. 151–155.
23. Han, Y.C.; Tong, X.Q. Day-Ahead Wind Power Prediction Based on Corrected Transformer Network. In Proceedings of the 2023 6th Asia Conference on Energy and Electrical Engineering (ACEEE), Chengdu, China, 21–23 July 2023; pp. 404–408.
24. Jiang, B.X.; Liu, Y.; Xie, H. Super Short-Term Wind Speed Prediction Based on CEEMD Decomposition and BiLSTM-Transformer Model. In Proceedings of the 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 29–31 January 2023; pp. 876–882.
25. Li, R.; Zhang, J.C.; Zhao, X.W. Deep Learning-Based Wind Farm Power Prediction Using Transformer Network. In Proceedings of the 2022 European Control Conference (ECC), London, UK, 12–15 July 2022; pp. 1018–1023.
26. Liu, Z.M.; Wang, Y.X.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljagic, M.; Hou, T.Y.; Tegmark, M. KAN: Kolmogorov-Arnold Networks. *arXiv* **2024**, arXiv:2404.19756.
27. Sulaiman, M.H.; Mustafa, Z.; Saealal, M.S.; Saari, M.M.; Ahmad, A.Z. Utilizing the Kolmogorov-Arnold Networks for Chiller Energy Consumption Prediction in Commercial Building. *J. Build. Eng.* **2024**, *96*, 110475. [[CrossRef](#)]
28. Han, X.; Zhang, X.F.; Wu, Y.L.; Zhang, Z.D.; Wu, Z. KAN4TSF: Are KAN and KAN-based Models Effective for Time Series Forecasting? *arXiv* **2024**, arXiv:2408.11306.
29. Guo, J.C.; Song, X.W.; Liu, C.; Zhang, Y.F.; Guo, S.J.; Wu, J.X.; Cai, C.; Li, Q.A. Research on the Icing Diagnosis of Wind Turbine Blades Based on FS-XGBoost-EWMA. *Energy Eng.* **2024**, *121*, 1739–1758. [[CrossRef](#)]
30. Nouhitehrani, S.; Caro, E.; Juan, J. Computation of Prediction Intervals of Wind Energy Based on the EWMA and BOA Techniques. *Sustain. Energy Techn.* **2024**, *66*, 103806. [[CrossRef](#)]
31. 200 MW Wind Power Data. Available online: [https://blog.csdn.net/2301\\_79953585/article/details/142982461](https://blog.csdn.net/2301_79953585/article/details/142982461) (accessed on 17 October 2024).
32. Lan, Y.; Wang, N.Z.; Lin, X.Y. Ultra-short term prediction of wind power based on PSO-SVR algorithm. In Proceedings of the 2023 IEEE PELS Students and Young Professionals Symposium (SYPS), Shanghai, China, 27–29 August 2023; pp. 1–5.

33. Qin, H.; Huang, L.J.; Li, K.; Cheng, G.H. Short-Term Offshore Wind Power Prediction based on VMD-SE-BP Neural Network Model. In Proceedings of the 2024 IEEE 2nd International Conference on Power Science and Technology (ICPST), Dali, China, 9–11 May 2024; pp. 1476–1481.
34. Li, Z.; Zhou, S.H.; Yu, Y.X.; Shang, Y.; Gao, Z.Q. Short-Term Wind Power Prediction Model Based on WRF-RF Model. In Proceedings of the 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 26–28 April 2023; pp. 599–604.

**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.