One Improved Wasserstein GAN with Gradient Penalty for Grain Consumption Prediction

Pei Li, Chunhua Zhu*

Abstract—**Prediction of grain consumption is crucial for analyzing the changing trend and balancing the grain supply and demand in China. Recently, the use of generative adversarial networks (GAN) to capture the distribution of historical data for generating future data has gained attention in time-series prediction. In order to enhance prediction performance and address model instability, an improved Wasserstein GAN with gradient penalty, referred to as IWGAN-GP, is proposed. The IWGAN-GP utilizes a bidirectional long short-term memory neural network (BiLSTM) as the generator and a convolutional neural network (CNN) as the discriminator, combining the memory capabilities of LSTM with the nonlinear feature extraction capabilities of CNN. Specifically, the loss function of the generator incorporates the mean square error (MSE) between real and generated samples to optimize the LSTM network, while the loss function of the discriminator includes the L_1 norm as the gradient penalty term to enhance sparsity and robustness, in contrast to the L_2 norm used in existing WGAN-GP models. Experimental results on grain consumption data from 1981 to 2020 demonstrate that the proposed IWGAN-GP improves prediction accuracy compared to BiLSTM, GAN, and WGAN-GP models.**

Index Terms—**grain consumption prediction, BiLSTM, WGAN-GP, L_1 norm; EEMD**

I. Introduction

Accurately understanding the evolving trend of grain consumption is crucial for strengthening the macro strategic control of grain and ensuring the balance of grain supply and demand. Grain consumption data typically follows a time series pattern. Among deep learning techniques, long short-term memory (LSTM) [1, 2, 3], gate recurrent unit (GRU) [4], and Transformer [5], have shown promise in time series prediction. The emerging generative adversarial networks (GAN) performs better to explore time series prediction [6]. GAN consists of a generator and a discriminator, which will be trained against each other [7]. GAN has been effective in image generation but nor yet in time series prediction. Given the linearity and stationarity of grain consumption data, GAN for time series prediction is theoretically feasible. In 2021, an emotion-guided stock price prediction model using conditional generative adversarial network (CGAN) is introduced [6]. The CGAN incorporates LSTM and multilayer perceptron (MLP) in the generator and discriminator, respectively, along with the emotional information from daily tweets as a conditional input, enhancing stock prediction accuracy. Additionally, researchers have proposed various GAN-based models for financial time serious prediction. Lin H. C et al. [8] proposes a Wasserstein GAN with gradient penalty (WGAN-GP) network with a GRU as the generator and a convolutional neural network (CNN) as the discriminator to investigate whether adversarial systems can help improve time series prediction performance. Experiments have demonstrated that the adversarial network outperforms the traditional LSTM network. Wang Jing et al. [9] introduces an empirical mode decomposition generative adversarial network (EMD-WGAN) for financial time series prediction. The generator consists of empirical mode decomposition (EMD) and multiple LSTM networks, while the discriminator adopts CNN. The data after EMD exhibit similar frequency and good regularity, reducing the complexity of the generated model and enhancing the prediction accuracy [10]. Furthermore, the EMD-WGAN utilizes the loss function of WGAN-GP to address the instability of the original GAN [11]. Shuntaro Takahashi et al. [12] proposes the financial time series model FIN-GAN, where the generator and discriminator combine MLP and CNN. Therefore, GAN shows potential advantages in time series prediction. However, for specific applications, the network structure, loss function, etc., have an impact on prediction performance, which is also related to the characteristics of nonlinearity and non-stationarity in historical data series.

To enhance prediction performance and address model instability in grain consumption prediction, an improved WGAN-GP (IWGAN-GP) is proposed. The IWGAN-GP incorporates bidirectional long short-term memory neural network (BiLSTM) as generators and CNN as the discriminator, with a new loss function introducing mean square error (MSE) to optimize the generator and ensure generated data closely resemble real data. Furthermore, the discriminator loss function in WGAN-GP is modified to use L1 norm instead of L2 norm, enhancing sparsity and robustness while reducing model complexity.

II. The proposed IWGAN-GP

The proposed IWGAN-GP architecture is illustrated in Figure 1, consisting of a data preprocessing module for data smoothing and a GAN for predicting future data. The generator of IWGAN-GP utilizes BiLSTM to generate the predicted value $\tilde{x}_{t+1}^{'}$ at time t+1; then, it combines real data

Manuscript received January 10, 2024; revised June 22, 2024.

This work was supported in part by Open subject of Scientific research platform in Grain Information Processing Center under Grant KFJJ2022011; The Innovative Funds Plan of Henan University of Technology under Grant 2022ZKCJ13.

Pei Li is an Associate Professor in Zhengzhou Tourism College, Zhengzhou, Henan, China. (e-mail: lipei@zztrc.edu.cn).

Chunhua Zhu is a Professor in Henan University of Technology, Zhengzhou, Henan, China. (corresponding author: 186-2371-6908; e-mail: zhuchunhua@haut.edu.cn).

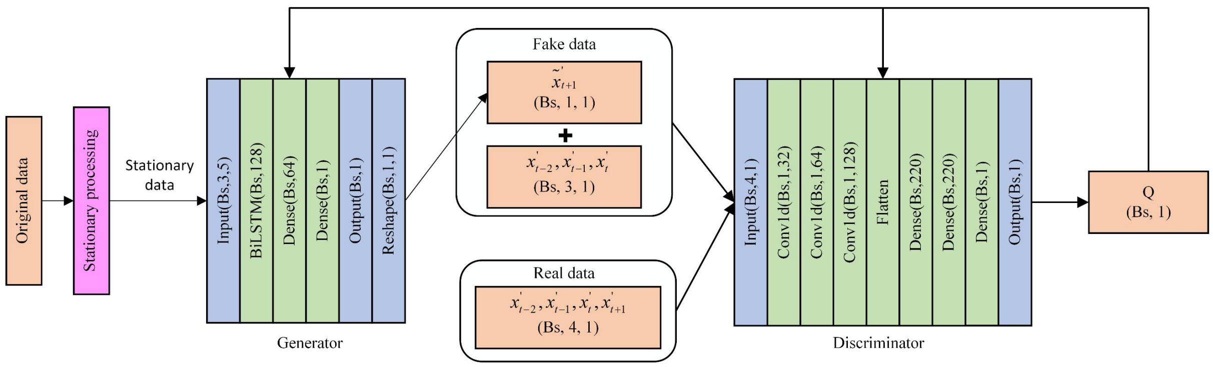


Fig. 1 The proposed IWGAN-GP architecture.

x'_{t-2}, x'_{t-1}, x'_t at the first three points with $\tilde{x}'_{t+1}$ to produce fake data $x'_{t-2}, x'_{t-1}, x'_t, \tilde{x}'_{t+1}$. similarly, real data $x'_{t-2}, x'_{t-1}, x'_t, x'_{t+1}$ is denoted as data at time *t*. Both fake data and real data are inputted into the discriminator to capture the correlation and time series information between x'_{t+1} 、 $\tilde{x}'_{t+1}$ and x'_{t-2}, x'_{t-1}, x'_t .The discriminator of IWGAN-GP employs CNN, where its output value Q represents the discrepancy between real data and generated data. The generator and discriminator are trained through alternating iterations until the output value of the discriminator converges close to zero or fluctuates slightly around zero. The " + " in Figure 1 signifies vector concatenation.

A. *Stabilization processing*

The dataset used in this study is sourced from the official website of the National Bureau of Statistics and the China Statistical Yearbook. China's grain consumption is categorized into food grain and non-food grain [13], with food grain including rations and feed grains, and non-food grain including industrial grain, seed grain and loss grain. The trend chart depicting China's total grain consumption from 1981 to 2020 is illustrated in Figure 2, with the ordinate unit being 10,000 tons. The stationarity of grain consumption data is evaluated using the Augmented Dickey-Fuller (ADF) unit root test method, with the calculated significance test statistics indicating non-stationary with a P value of 0.887 [14]. Globally, the grain consumption displays the characteristics of non-linearity, non-stationary, and an overall increasing trend, as evident from Figure 2.

To address the non-linearity and non-stationary of the grain consumption data, stabilization processing is crucial to ensure the processed data with the same distribution. A stationary time series refers to patterns in the series that remain constant over time, which are essential for subsequent prediction. Normalization and ensemble empirical mode decomposition (EEMD) are employed to smooth the original data, as depicted in Figure 3. Normalization restricts the data with a specific range (e.g., [0,1]) to mitigate the adverse effects of singular sample data. Max-min normalization is utilized to preprocess the original time series, defined as:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

where $X = (x_1, x_2, \ldots x_n)$ represents the historical grain consumption series, $X_{\min}$ and $X_{\max}$ represent the minimum and maximum value of X , respectively, and $X' = (x'_1, x'_2, \ldots, x'_n)$ is the normalized series. Considering the non-stationarity in X' , EEMD is used to smooth X' .

The EEMD algorithm, proposed by Huang [15], involves multiple empirical mode decomposition with Gaussian white noise to adaptively decompose the signal and avoid the mode mixing. The EEMD decomposition process includes:

1) Providing sequence X' and the number of processing times *m*;
2) Adding *m* groups of random white noise $W_1, W_2, \ldots, W_m$ to sequence X' to form $X'_1, X'_2, \ldots, X'_m$;
$$X'_j = X' + W_j \quad (j = 1, 2, \ldots, m) \tag{2}$$
3) Decomposing $X'_1, X'_2, \ldots, X'_m$ using EMD to obtain a series of Intrinsic Mode Function (IMF) components $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$;
4) Averaging the corresponding IMF components to perform EEMD decomposition.
$$a_i = \frac{1}{m} \sum_{j=1}^{m} a_{i,j} (i = 1, 2, \ldots, N; j = 1, 2, \ldots, m) \tag{3}$$

where N is the number of IMF components after decomposition. As shown in Figure 3, the sequence X' undergoes EEMD to obtain four IMF components and one residual component RES, so $N = 5$. Let $\mathbf{E} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5)^T_{n*5} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n)$, then $\mathbf{e}_n$ can be considered as the five features associated with x'_n . Before inputting into the generator, these data need to be reshaped into three dimensions, including total number of samples, time step and feature number. In the experiment, the data from the previous three years will be used to predict the grain consumption for the next year, with the three-dimensional values are *n*-3, 3, 5, respectively.

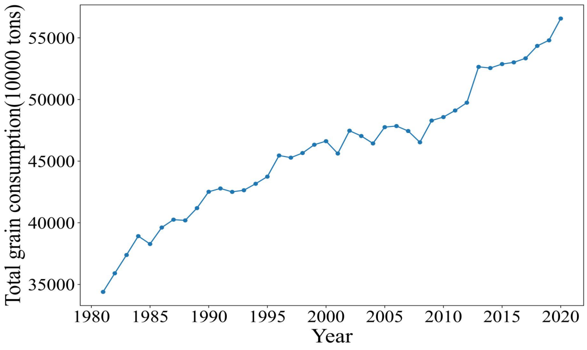


Fig. 2 Trend of total grain consumption (1981 to 2020).

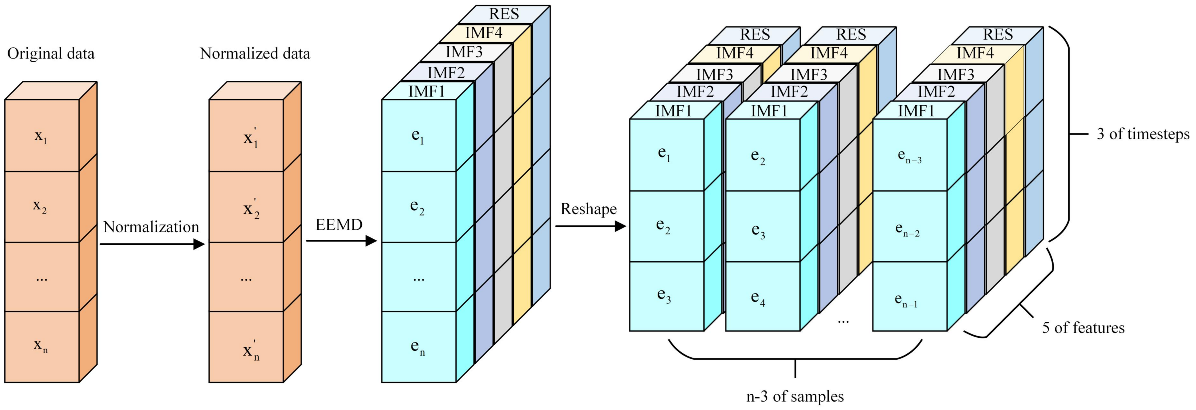


Fig. 3 The stabilization processing

B. The Timing generator based on BiLSTM

As previously mentioned, the generator utilizes BiLSTM, a type of recurrent neural network (RNN) that is able to retain information from previous time points to calculate the information at the current time point [16]. RNNs have shown great performance in handling time series data. However, RNNs suffer from a significant long-term dependency issue [3], where the neural network nodes may fail to capture features from earlier time points after multiple computation stages due to gradient disappearance and gradient explosion caused by cyclic multiplication of RNN weight matrix. This is a main challenge in training RNN models. To address these issues, LSTM introduces a gate structure to regulate the storage and forgetting of features, including the forget gate $\mathbf{f}_t$, the input fate $\mathbf{i}_t$ and the output gate $\mathbf{o}_t$ [17]. The LSTM update can be summarized as follows:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1};\mathbf{k}_t]+\mathbf{b}_f) \tag{4}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1};\mathbf{k}_t]+\mathbf{b}_i) \tag{5}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1};\mathbf{k}_t]+\mathbf{b}_o) \tag{6}$$

$$\mathbf{s}_t = \mathbf{f}_t \odot \mathbf{s}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_s[\mathbf{h}_{t-1};\mathbf{k}_t]+\mathbf{b}_s) \tag{7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{s}_t) \tag{8}$$

where $[\mathbf{h}_{t-1};\mathbf{k}_t]$ is a concatenation of the previous hidden state $\mathbf{h}_{t-1}$ and the current input $\mathbf{k}_t$; $\mathbf{s}_t$ is a memory cell at time t; $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_o$, $\mathbf{W}_s$ and $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_o$, $\mathbf{b}_s$ are learning parameters; σ and $\odot$ represent a logistic sigmoid function and element-wise multiplication, respectively.

In theory, reversing data in the dataset can help LSTM uncover pattern related to grain consumption. Therefore, the generator implements BiLSTM.

A single-layer BiLSTM essentially consists of two LSTMs: one processes the forward sequence, and the other processes the reverse sequence. The outputs of the two LSTMs are then concatenated. As illustrated in Figure 4, the forward LSTM generates an output vector after three-time steps, while the reverse LSTM produces another output after three-time steps. These two output vectors are combined to form the final output of the BiLSTM [18, 19].

The generator includes a BiLSTM layer with 128 neurons, followed by two Dense layers with 64 and 1 neurons, respectively. The number of neurons in the last layer matches the output step. Figure 4 depicts the generation process of a single sample input into the generator and defines the overall output of the generator as $\tilde{X}' = (\tilde{X}'_4, \tilde{X}'_5, ..., \tilde{X}'_n)$.

C. CNN-based Discriminator

CNN can automatically extract deep features and obtain widely application in image classification or text analysis. For instance, when processing an image of a dog, the initial convolution layer detects edges, followed by the detection of shapes in the second layer, and the identification of specific features like the nose in the third layer. In the context of time series data, individual data points form small trends, which in turn contribute to larger patterns, ultimately revealing consumption trends. The ability of CNN to identify patterns can be leveraged to extract valuable information on grain consumption fluctuations [20]. Additionally, CNN is adept at handling data with local spatial correlations. In contrast to scattered data points, adjacent data points exhibit stronger correlations, a phenomenon also observed in time series data.Given that grain consumption data points are collected annually, it is reasonable to assume that data points from consecutive years will exhibit stronger correlations [21].

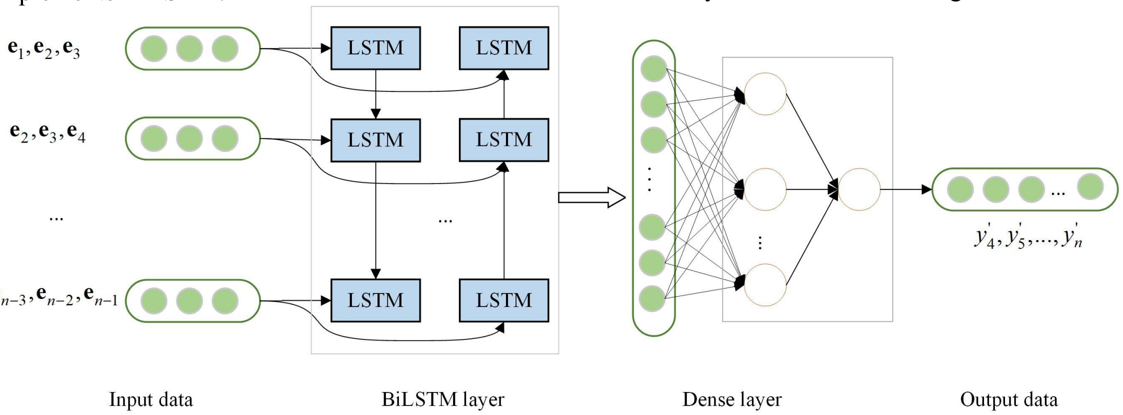


Fig. 4 The timing generator based on BiLSTM

The discriminator in the proposed IWGAN-GP consists of three one-dimensional (1D) convolutional layers, each with 3, 2, and 1 convolution kernels respectively, with a uniform step size of 2. To maintain the size of the features during convolution, the zero-padding method is applied. For one-dimensional data, P rows of zeros are added before the first row and after the last row, with the number of filled rows P determined by

$$n_{output} = (n_{input} - K + 2 * P) / S + 1 \tag{9}$$

where input and output sizes are denoted by n_{input} and n_{output}, respectively. From Figure 1, $n_{input} = 4$, K is the convolution kernel, P is the number of filled rows on each side, S is the step size. In the TensorFlow implementation, the output shape is calculated by

$$n_{output} = \left\lceil \frac{n_{input}}{S} \right\rceil \tag{10}$$

where $\lceil \ \rceil$ represents rounding up. Following the convolutional layers, three additional dense layers with 220, and 1 neuron are included. The Rectified Linear Unit (ReLU) serves as the activation function between these layers, except for the final layer. Figure 5 is for a visual representation of a single sample input into the CNN.

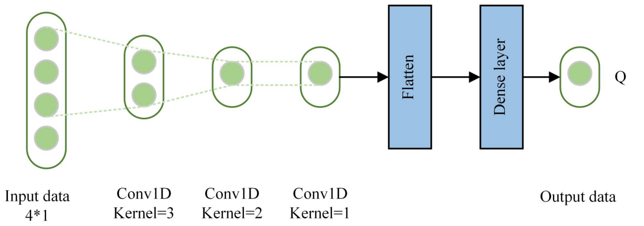


Fig. 5 The discriminator based on CNN.

D. The Improved Wasserstein Distance Loss

The loss function of the original GAN is based on KL-JS divergence. When training, cross-entropy loss is utilized to minimize the difference between the real data distribution and the generated data distribution, which is equivalent to minimize the KL-JS divergence. The objective function for the discriminator is defined as

$$-E_{x\sim P_r}[\lg D(x)] - E_{x\sim P_g}[\lg(1 - D(x))] \tag{11}$$

and the loss function for the generator is

$$-E_{x\sim P_g}[\lg(D(x))] \tag{12}$$

where E denotes expectation, P_g represents the generated data distribution, and P_r is real data distribution. A significant issue with JS divergence arises when the two distributions have minimal or no overlap, leading to JS divergence being fixed at a constant value of log2. This results in the gradient descent information of the generator being 0 [22], rendering it unreasonable to represent the loss function using this information. WGAN-GP is introduced to stabilize and enhance the training of GAN [11]. WGAN-GP employs Wasserstein distance (or Earth-Mover distance (EMD)), which calculates the minimum cost of transporting mass to convert one data distribution to another. The Wasserstein distance between P_r and P_g is mathematically defined [23] as

$$W(P_r, P_g) = \inf_{\gamma\in\Pi(P_r,P_g)} E_{(x,y)\sim\gamma}[\| x - y \|] \tag{13}$$

where $\Pi(P_r, P_g)$ denotes the set of all joint distributions between P_r and P_g, Π contains all the possible transport plans γ. By utilizing the Kantorovich-Rubinstein duality, the calculation can be simplified to

$$W(P_r, P_g) = \sup_{\|f\|_L \le 1} E_{x\sim P_r}[f(x)] - E_{x\sim P_g}[f(x)] \tag{14}$$

where sup is the least upper bound and f is a 1-Lipschitz function following Lipschitz constraint as

$$| f(x_1) - f(x_2) | \le | x_1 - x_2 | \tag{15}$$

The WGAN-GP introduces a gradient penalty to enforce the Lipschitz constraint. A function f is considered as1-Lipschitz if its gradients have a norm at most $1(\|\nabla f\|_2 \le 1)$ everywhere. If the gradient norm deviates from the target norm value of 1, the model will be penalized. Comparison to the basic GAN, the WGAN-GP network lacks the sigmoid function and outputs a scalar score rather than a probability. This score indicates the authenticity of the input data. In the discriminator and generator, the loss function is represented respectively as

$$E_{x\sim P_g}[D(x)] - E_{x\sim P_r}[D(x)] + \lambda E_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_x D(x)\|_2 - 1)^2] \tag{16}$$

$$-E_{x\sim P_g}[D(x)] \tag{17}$$

where $\hat{x}$ represents the intermediate value between the real and generated sample space. The proposed IWGAN-GP introduces two enhancements to the basic WGAN-GP loss function. Firstly, it adopts the L_1 norm instead of the L_2 norm in the gradient penalty section of the discriminator loss function. The L_1 norm exhibits better tolerance to outliers compared to the L_2 norm. This is because the L_2 norm squares the error, the proposed IWGAN-GP will be more sensitive to outliers, sacrificing many normal samples. When the dataset contains outliers, L_1-norm is more effective than L_2-norm. Additionally, L_1 regularization leads to sparser parameter settings compared to L_2 regularization, reducing model complexity. The discriminator loss function in the proposed IWGAN-GP is:

$$E_{x\sim P_g}[D(x)] - E_{x\sim P_r}[D(x)] + \lambda E_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_x D(x)\|_1 - 1)^2] \tag{18}$$

Secondly, the Mean Squared Error (MSE) between real samples and the generated samples is incorporated into the generator's loss function. This addition aims to improve the model stability by adjusting the generator based on the MSE when the discriminator makes incorrect judgments. The generator's loss function for enhancing WGAN-GP is given by:

$$-E_{x\sim P_g}[D(x)] - E[(x_{P_r} - x_{P_g})^2] \tag{19}$$

Table I compares the loss functions of generators and discriminators in the basic GAN, basic WGAN-GP, and proposed IWGAN-GP.

III. Experiments

A. Evaluation index

Similar to the classical evaluating metrics of regression algorithms, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) are utilized to evaluate the performance of the proposed IWGAN-GP.

In Section 2.2, the output of the generator

$\widetilde{X}^{'} = (\tilde{x}_4^{'}, \tilde{x}_5^{'}, ..., \tilde{x}_n^{'})$ is denormalized to obtain the predicted values $Y = (y_4, y_5, ..., y_n)$; given the original data $X = (x_4, x_5, ..., x_n)$. These metrics are calculated as follows

1) Mean Squared Error (MSE): A commonly used metric in regression analysis, defined as the average of the squared differences between predicted data and original data.

$$MSE = \frac{1}{n-3}\sum_{t=4}^{n}(y_t - x_t)^2 \tag{20}$$

2) Root Mean Squared Error (RMSE): The square root of the average of the squares of prediction errors, which considers the magnitude of the prediction errors and penalizes larger errors more.

$$RMSE = \sqrt{\frac{1}{n-3}\sum_{t=4}^{n}(y_t - x_t)^2} \tag{21}$$

3) Mean Absolute Error (MAE): The average of the absolute differences between predicted data and original data. The smaller MAE represents the better predictive performance.

$$MAE = \frac{1}{n-3}\sum_{t=4}^{n}\left|y_t - x_t\right| \tag{22}$$

4) Mean Absolute Percentage Error (MAPE): The average of the percentage differences between predicted data and original data, often used to measure relative error size.

$$MAPE = \frac{1}{n-3}\sum_{t=4}^{n}\frac{|y_t - x_t|}{x_t} \tag{23}$$

B. Settings

In evaluating the proposed IWAGN-GP, the BiLSTM, the basic GAN and the basic WGAN-GP are also be implemented for comparison, of which Both the GAN and the WGAN-GP adopt the BiLSTM as the generator and CNN as the discriminator. All models utilize Adam's optimization algorithm with learning rates selected from 0.0001, 0.0003, 0.001, to 0.003. Through experiments, it is found that lower learning rates lead to smoother loss function changes and improved prediction results. Therefore, the optimal learning rate is set to 0.0001, the number of cycles is determined by the stability of the loss function curve, and the random number seed for IWGAN-GP is set to 5.

C. Results

The prediction results of BiLSTM, GAN, WGAN-GP, and IWGAN-GP are depicted in Figure 6. Figure 6 illustrates that it can be seen that the predicted data trend aligns with the actual data trend. However, the proposed IWGAN-GP (Figure 6 (d)) demonstrates a better fit to actual data compared to the other three models. In Figure 6 (a), the BiLSTM prediction results show a nearly straight line, which contradicts the non-stationary and non-linear characteristics depicted in Figure 2. Comparing Figure 6 (b) and Figure 6 (c), the prediction results of GAN and WGAN-GP are similar before 2009, but WGAN-GP performs worse than GAN after 2009. The proposed IWGAN-GP shows better fitting to the actual data compared to the other models.

To objectively evaluate the predictive performance of IWGAN-GP, four evaluation metrics-MSE, RMSE, MAE, MAPE are utilized to compare three models. Table II is the predictive performance of the proposed IWGN-GP for training datasets and Table III for test datasets, with the best results highlighted in bold. The analysis reveals that IWGAN-GP shows the best predictive performance, followed by WGAN-GP, while GAN performs the weakest. A comparison of the four metrics shows that IWGAN-GP demonstrates lower predictive errors, reducing MSE, RMSE, MAE, and MAPE on the test set by 72.33%, 70.87%, 72.72%, and 46.62% respectively compared to GAN; and by 66.33%, 41.12%, 47.17%, and 41.34% respectively compared to WGAN-GP.

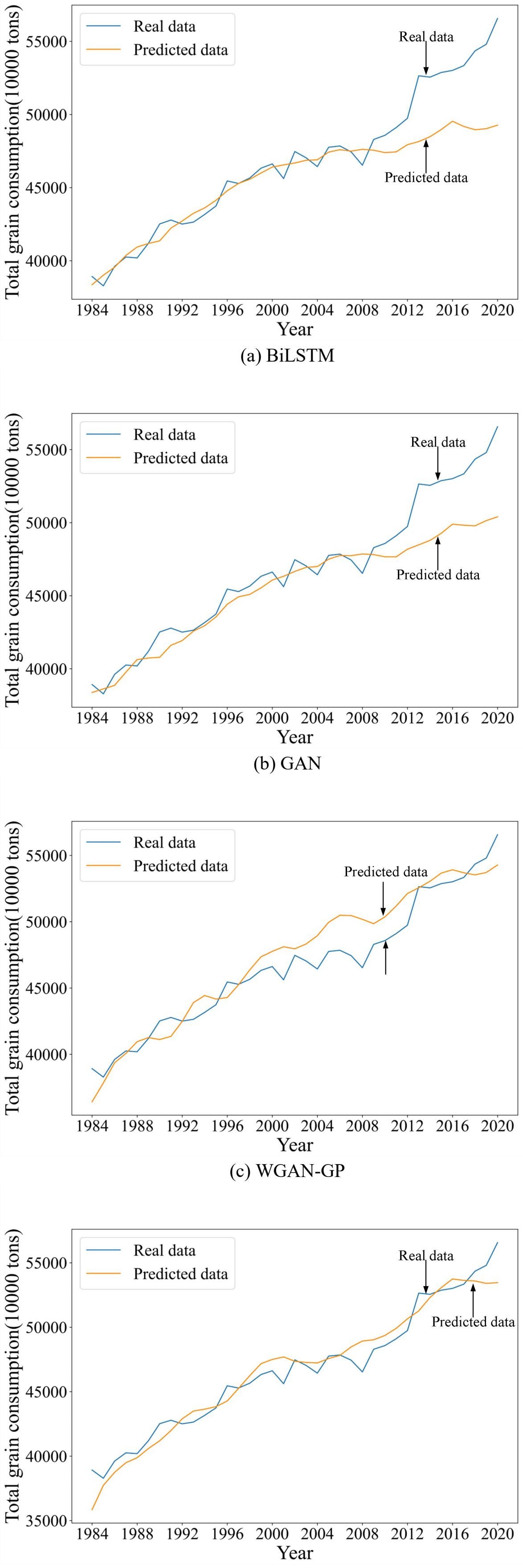
TABLE I
LOSS FUNCTIONS OF THE DISCRIMINATOR AND GENERATOR OF DIFFERENT GAN MODELS

Model	Discriminator	Generator
basic GAN	$-E_{x\sim P_r}[\lg D(x)]$ $-E_{x\sim P_g}[\lg(1-D(x))]$	$-E_{x\sim P_g}[\lg(D(x))]$
basic WGAN-GP	$E_{x\sim P_g}[D(x)]-E_{x\sim P_r}[D(x)]$ $+\lambda E_{\hat{x}\sim P_{\hat{x}}}[(\Vert\nabla_x D(x)\Vert_2-1)^2]$	$-E_{x\sim P_g}[D(x)]$
Proposed IWGAN-GP	$E_{x\sim P_g}[D(x)]-E_{x\sim P_r}[D(x)]$ $+\lambda E_{\hat{x}\sim P_{\hat{x}}}[(\Vert\nabla_x D(x)\Vert_1-1)^2]$	$-E_{x\sim P_g}[D(x)]$ $-E[(x_{P_r}-x_{P_g})^2]$

Table IV displays the grain consumption forecasts obtained by different methods from 2016 to 2020. The bolded results indicate the values closest to the actual values. From Table IV, the proposed IWGAN-GP, which includes BiLSTM and CNN, effectively captures both long-term and short-term characteristics of grain consumption trends, resulting in improved fitting performance. Table V presents the relative errors calculated from the actual and predicted grain consumption from 2016 to 2020 using different methods, with the smallest errors highlighted in bold. The average errors for the basic GAN, basic WGAN-GP, and IWGAN-GP over these five years are 7.40%, 7.95%, and 3.70%, respectively. IWGAN-GP has increased the prediction accuracy by 50% and 53.46% compared to the GAN and WGAN-GP, respectively.

IV. Conclusion

A new WGAN-GP model is proposed for grain consumption prediction, involving the following key aspects: 1) construction of a new WGAN-GP using BiLSTM as the generator and CNN as the discriminator to enhance the extraction of time series features; 2) introduction of prediction MSE in the generator loss function to minimize the difference between generated and real data; 3) adoption of the L_1 norm instead of the L_2 norm in the gradient penalty term of the discriminator loss function to enhance model sparsity and robustness. It is verified by experiments that the proposed IWGAN-GP enhances prediction performance compared to existing models, and the concept of the proposed WGAN-GP can be applied to other time series forecasting tasks.



(a) BiLSTM

(b) GAN

(c) WGAN-GP

(d) Proposed IWGAN-GP

Fig. 6 Prediction results of the proposed IWGAN-GP

TABLE II
PREDICTIVE PERFORMANCE OF THE PROPOSED IWGAN-GP ON TRAINING SET

Evaluation metrics	GAN	WGAN−GP	IWGAN−GP
MSE	0. 0031	0. 0080	**0. 0006**
RMSE	0. 5522	0. 0896	**0. 0253**
MAE	0. 0426	0. 0833	**0. 0221**
MAPE (%)	12. 7532	19. 5332	**5. 3200**

TABLE III
PREDICTIVE PERFORMANCE OF THE PROPOSED IWGAN-GP ON TEST SET

Evaluation metrics	GAN	WGAN−GP	IWGAN−GP
MSE	0. 0365	0. 0300	**0. 0101**
RMSE	0. 1910	0. 1733	**0. 1003**
MAE	0. 1785	0. 1634	**0. 0943**
MAPE (%)	21. 4223	19. 4265	**11. 4357**

TABLE IV
PREDICTIVE VALUES OF THE PROPOSED IWGAN-GP FOR THE YEARS 2016-2020 (THE THOUSAND TONS)

Year	Actual values	Predicted Values		
		GAN	WGAN-GP	IWGAN-GP
2016	53012.2906	49929.9152	49232.3567	**51440.8984**
2017	53340.04034	49794.9574	49835.3661	**51967.0522**
2018	54348.09278	50640.7640	50236.8975	**52425.5705**
2019	54801.2475	50760.6806	50484.1072	**52812.3368**
2020	56566.0091	50499.2780	50552.3718	**53296.7498**

TABLE V
RELATIVE ERROR OF THE PROPOSED IWGAN-GP FOR THE YEARS 2016-2020 (THE THOUSAND TONS)

Year	Relative Errors		
	GAN	WGAN-GP	IWGAN-GP
2016	5.81%	7.13%	**2.96%**
2017	6.65%	6.57%	**2.57%**
2018	6.82%	7.56%	**3.54%**
2019	7.37%	7.88%	**3.63%**
2020	10.73%	10.63%	**5.78%**

REFERENCES

[1]. S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp1735-178, 1997.

[2]. K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares , H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arxiv preprint arxiv:1406.1078, 2014.

[3]. H. Sen, "Time Series Prediction based on Improved Deep Learning," IAENG International Journal of Computer Science, vol. 49, no.4, pp1133-1138, 2022.

[4]. J. Chen, H. Dai, S. Wang, and C. Liu, "Improving Accuracy and Efficiency in Time Series Forecasting with an Optimized Transformer Model," Engineering Letters, vol. 32, no. 1, pp1-11, 2024.

[5]. Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult," IEEE Transactions on Neural Networks, vol. 5, no. 2, pp157-166, 1994.

[6]. Y. Zhang, J. Li, H. Wang and S. C. T. Choi. "Sentiment-guided adversarial learning for stock price prediction," Frontiers in Applied Mathematics and Statistics, vol. 7, 601105, 2021.

[7]. I. Goodfellow I, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial networks," in Communications of the ACM, vol. 63, no.11, pp 139-144, 2020.

[8]. H. Lin, C. Chen, G. Huang and A. Jafari, "Stock price prediction using generative adversarial networks," Journal of Computer Science, vol 17, no.3, pp188-196, 2021.

[9]. J. Wang, H. Zou, D. Qu, and L. Bai. "Financial time series prediction based on empirical mode decomposition generative adversarial networks," Computer Applications and Software, vol. 37, pp293-297, 2020.

[10]. S. B. Jiang, C. F. Bai, and C. F. Du, "Traffic forecast based on empirical mode decomposition and RBF neural network," Advanced Materials Research, vol. 846, pp1270-1273, 2014.

[11]. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. C. Courville, "Improved training of Wasserstein gans," Advances in neural information processing systems, vol. 30, pp1-11, 2017.

[12]. S. Takahashi, Y. Chen, and K. Tanaka-Ishii, "Modeling financial time-series with generative adversarial networks," Physica A: Statistical Mechanics and its Applications, vol.527, 121261, 2019.

[13]. B. Gong, and F. Y. Yin, "Measurement of the impact of urbanization on China’s grain consumption," Theory Pract. Financ. Econ, vol. 39, pp134-140, 2018.

[14]. C. J. Ogbonna, C. J. Nweke, E. C. Nwogu, and I. S. Iwueze, "Wavelet Transform as an Alternative to Power Transformation in Time Series Analysis," Bulletin of Mathematical Sciences and Applications, vol. 17, pp57-74, 2016.

[15]. W. Wang, K. Chau, D. Xu, and X. Y. Chen, "Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition," Water Resources Management, vol. 29, pp2655-2675, 2015.

[16]. D. Song, H. Chen, G. Jiang, and Y. Qin, "Dual stage attention based recurrent neural network for time series prediction," U.S. Patent No. 10, 929, 674. 23 Feb. 2021.

[17]. Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," Knowledge-Based Systems, vol. 181, pp 1-9, 2019.

[18]. S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," 2019 IEEE International Conference on Big Data (Big Data), pp3285-3292, 2019.

[19]. O. S. Bérot, H. Canot, P. Durand, B. Hassoune-Rhabbour, H. Acheritobehere, C. Laforet, and V. Nassiet, "Choice of Parameters of an LSTM Network, based on a Small Experimental Dataset," Engineering Letters, vol. 32, no. 1, pp59-71, 2024.

[20]. D. Hussain, T. Hussain, A. A. Khan, S. A. A. Naqvi and A. Jamil, "A deep learning approach for hydrological time-series prediction: A case study of Gilgit river basin," Earth Science Informatics, vol.13, pp915-927, 2020.

[21]. F. Liu, X. Zhou, J. Cao, Z. Wang, T. Wang, H. Wang, and Y. Zhang. " Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional LSTM-CNN," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 6, pp2626-2640, 2020.

[22]. K. F. Wang, C. Gou, Y. J. Duan, Y. L. Lin, and X. H. Zheng, "Generative adversarial networks: The state of the art and beyond, " Acta Automatica Sinica, vol. 43, no. 3, pp321-332, 2017.

[23]. X. Zhou, Z. Pan, G. Hu, S. Tang, and C. Zhao, " Stock Market Prediction on High-Frequency Data Using Generative Adversarial Nets," Mathematical Problems in Engineering, pp1-11, 2018.

Pei Li (1985.12-), female, acts as an Associate Professor in Zhengzhou Tourism College. Her interest is computer related research.

Chunhua Zhu (1976.08-), female, acts as a Professor in Henan University of Technology. Her interest is intelligent signal and information processing related research.