processes

MDPI

Article

A Novel Ensemble Machine Learning Model for Oil Production Prediction with Two-Stage Data Preprocessing

Zhe Fan [1], Xiusen Liu [2], Zuoqian Wang [3], Pengcheng Liu [1,*] and Yanwei Wang [1]

[1] School of Energy Resources, China University of Geosciences, Beijing 100083, China; 3006200045@email.cugb.edu.cn (Z.F.); wangyw@cugb.edu.cn (Y.W.)
[2] Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong SAR, China; xiusen24@connect.hku.hk
[3] Research Institute of Petroleum Exploration and Development, PetroChina, Beijing 100083, China; wangzuoqian@petrochina.com.cn
* Correspondence: lpc@cugb.edu.cn

Abstract: Petroleum production forecasting involves the anticipation of fluid production from wells based on historical data. Compared to traditional empirical, statistical, or reservoir simulation-based models, machine learning techniques leverage inherent relationships among historical dynamic data to predict future production. These methods are characterized by readily available parameters, fast computational speeds, high precision, and time–cost advantages, making them widely applicable in oilfield production. In this study, time series forecast models utilizing robust and efficient machine learning techniques are formulated for the prediction of production. We have fused the two-stage data preprocessing methods and the attention mechanism into the temporal convolutional network-gated recurrent unit (TCN-GRU) model. Firstly, the random forest (RF) algorithm is employed to extract key dynamic production features that influence output, serving to reduce data dimensionality and mitigate overfitting. Next, the mode decomposition algorithm, complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN), is introduced. It employs a decomposition–reconstruction approach to segment production data into high-frequency noise components, low-frequency regular components and trend components. These segments are then individually subjected to prediction tasks, facilitating the model's ability to capture more accurate intrinsic relationships among the data. Finally, the TCN-GRU-MA model, which integrates a multi-head attention (MA) mechanism, is utilized for production forecasting. In this model, the TCN module is employed to capture temporal data features, while the attention mechanism assigns varying weights to highlight the most critical influencing factors. The experimental results indicate that the proposed model achieves outstanding predictive performance. Compared to the best-performing comparative model, it exhibits a reduction in RMSE by 3%, MAE by 1.6%, MAPE by 12.7%, and an increase in R^2 by 2.6% in Case 1. Similarly, in Case 2, there is a 7.7% decrease in RMSE, 7.7% in MAE, 11.6% in MAPE, and a 4.7% improvement in R^2.

Keywords: machine learning; oil production prediction; TCN-GRU-MA model; random forest algorithm; CEEMDAN algorithm

Citation: Fan, Z.; Liu, X.; Wang, Z.; Liu, P.; Wang, Y. A Novel Ensemble Machine Learning Model for Oil Production Prediction with Two-Stage Data Preprocessing. *Processes* **2024**, *12*, 587. https://doi.org/10.3390/pr12030587

Academic Editor: Xiong Luo

Received: 16 February 2024
Revised: 6 March 2024
Accepted: 12 March 2024
Published: 14 March 2024

1. Introduction

Production prediction has always been an essential task in the oil and gas industry. By employing models for petroleum production forecasting, the oil and gas industry can adapt the available time and technology in response to evolving operational and maintenance scenarios [1,2]. Accurate prediction results can assist enterprises in making reasonable development strategies, resource allocation plans, and investment schemes. It serves as a comprehensive and strategic tool in the development process of oil and gas fields, contributing to the enhancement of field development efficiency and the reduction of economic

risks. This is significant for the sustainable development of decision-makers, investors, and oil and gas companies. Traditional production forecasting methods primarily rely on empirical engineering formulas and numerical simulation methods [3]. Currently, the most commonly used method in oil fields is the Arps decline model [4]. It is a common method for predicting oil well production, summarized by scholars such as Arps based on a large amount of historical production data, suitable for describing the decay characteristics of oil well production. However, its drawback lies in its insufficient prediction accuracy. Numerical simulation involves constructing structural models of oil reservoirs and simulating production processes using computer technology. It can predict future dynamics on the premise of accurate historical fitting. Chinese oil companies have vast oil and gas assets overseas, but due to the unstable operating environment in resource-rich countries, it is difficult to conduct large-scale field-testing activities, leading to insufficient or unavailable critical reservoir parameters. The limited reservoir parameters have to some extent restricted the technique's applicability and generalization, affecting the prediction accuracy of numerical simulation [5,6]. In addition to the aforementioned ways, some statistical methods have also been introduced into the production forecasting domain, including statistical methods like autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), and autoregressive integrated moving average (ARIMA) [7]. These models are all based on the assumption of linear relationships, which may fail to capture accurate relationships between past and future production data when facing nonlinear or complex time series relationships. Therefore, it is challenging to establish effective nonlinear models. To overcome the limitations of traditional methods, many researchers delve into traditional methods, while others leverage artificial intelligence technologies to develop and train models for predicting.

Artificial intelligence methods have garnered significant attention across various domains such as weather forecast, economics, and engineering, driven by advancements in sophisticated artificial intelligence techniques and the availability of robust high-performance computing resources. Deep learning is a prominent approach in artificial intelligence. LeCun et al. introduced the convolutional neural network (CNN) model, which has multilayer learning architecture [8]. CNN models leverage spatial relationships to effectively reduce parameters and enhance training efficacy. Due to the limitations of convolutional layers, CNN models have traditionally been applied to short-term forecasting tasks, as they struggle to effectively capture temporal dependencies in time series data, thus making them less suitable for modeling long-term time series. To address this issue, Hochreiter and Schmidhuber introduced Long Short-Term Memory (LSTM) network [9]. LSTM utilizes gate mechanisms within its memory cells, enabling the network to determine which hidden states to retain and update. This architecture enables LSTM to capture long-term temporal dependencies in sequential data. Integrating LSTM with features extracted from raw data by other deep learning methods could enhance forecasting efficiency. Subsequently, a more efficient model, called gated recurrent unit (GRU) [10], was proposed. GRU optimizes LSTM's three gate functions by consolidating the forgetting and input gates into a unified update gate, reducing the number of required training parameters and effectively mitigating the vanishing gradient problem. TCN is an innovative architecture derived from CNN, distinguished by its utilization of expanded causal convolution and residual blocks [11,12]. This architecture endows TCNs with the capacity to extract features and conduct predictions on extensive time series samples. Moreover, TCN effectively mitigates the performance degradation commonly encountered in deep network training. The oil and gas industry has also greatly benefited from these advancements. They are applied in diverse petroleum engineering tasks, encompassing well log analysis [13], hydrocarbon production forecasting [14], bottom hole pressure prediction [15], reservoir heterogeneity characterization [16], and ultimate recovery estimation [17]. Utilizing deep learning algorithms for production forecasting is also a significant application in the oil and gas domain. By learning patterns and correlations from extensive historical dynamic data, novel data-driven methods for production forecasting have been achieved. Various models

have demonstrated satisfactory results in production forecasting [18–23]. However, there remains ample room for further research to enhance its accuracy.

In this paper, we introduce a novel ensemble machine learning model and validate its performance using monthly production data from individual wells in the Middle East oil fields and a carbonate reservoir within the territory of Central Asia. Specifically, preliminary screening of production feature parameters is conducted based on engineering expertise, followed by the utilization of the RF algorithm to select the most highly correlated dynamic parameters with production. Given the complexity of production data, where the model may struggle to fully capture its variability, the CEEMDAN algorithm is utilized to decompose the production into multiple components. Subsequently, these components are recombined based on the proposed recombination principles to obtain high-frequency, low-frequency, and trend components. Finally, the TCN-GRU-MA model is employed for modeling and prediction, with the final prediction results obtained through linear combination. A comparative analysis with the common ensemble models demonstrates the superior predictive performance of the proposed model.

The main contributions of this paper are as follows: (1) a two-stage data preprocessing method based on RF-CEEMDAN has been proposed; (2) a principle for recombining components is proposed, and by combining components reasonably, it reduces the accuracy degradation and the high time cost caused by too many components to be predicted; and (3) in both Case 1 and Case 2, comparative analyses were conducted with the other 12 machine learning models. The results indicate that the RF-CEEMDAN-TGMA model exhibits outstanding predictive performance.

2. Methodology

2.1. Random Forest Algorithm

The random forest algorithm is an iterative improvement of the decision tree algorithm. In the conventional decision tree model, the sample data are partitioned based on features, and each partition determines the optimal splitting property. With an increase in the number of features and corresponding branch nodes, the model constructed in this way is the decision tree [24–26].

The process of constructing a decision tree involves the following steps:

(a) Creation of root nodes: Initially, all data samples are placed in the root node. Subsequently, each feature is examined, and the optimal feature is identified to split the data samples, resulting in the creation of multiple subsets. Feature evaluation methods, such as information gain, information gain rate, and Gini index, are employed for this purpose.

(b) Creation of leaf nodes: The datasets divided by the optimal feature are placed in the leaf nodes.

(c) Segmentation of leaf nodes: For each sub-dataset, the feature set at that point consists of the remaining features after removing the optimal feature. The process continues by traversing all features, and the best feature is selected to further split the sub-dataset, forming a new subset.

(d) The construction of the decision tree model: Steps (a) to (c) are repeated until the predefined conditions for stopping the split are met. Typically, these stopping conditions include criteria such as the attainment of a specified number of leaf nodes that satisfy certain conditions, the utilization of all features for data division, and so on.

The construction of the decision tree involves a relatively straightforward process, and the computational complexity of the decision tree construction is relatively modest. However, since the data are locally optimized at each split, following a principle similar to the greedy algorithm for data sample partitioning, it is prone to overfitting. To mitigate this issue, the random forest adopts a random sampling method of putting back. Multiple sub-datasets are derived from the original data, and each is used to train a decision tree (weak classifier) model. The final model is then constructed through processes such as voting or taking the mean value, integrating the outputs of these individual models.

Figure 1 shows the structure of the random forest. The model construction process involves several steps. Firstly, utilizing the Bagging method, N training datasets are generated by selecting sample datasets from the original training datasets. Subsequently, N decision tree models are individually trained based on these N training datasets. Lastly, the random forest is formed by combining these N decision trees. In classification problems, the ultimate classification outcome is determined by the ensemble of N decision tree classifiers. In regression problems, the final prediction result is determined by averaging the predicted values from the N decision trees [27,28].

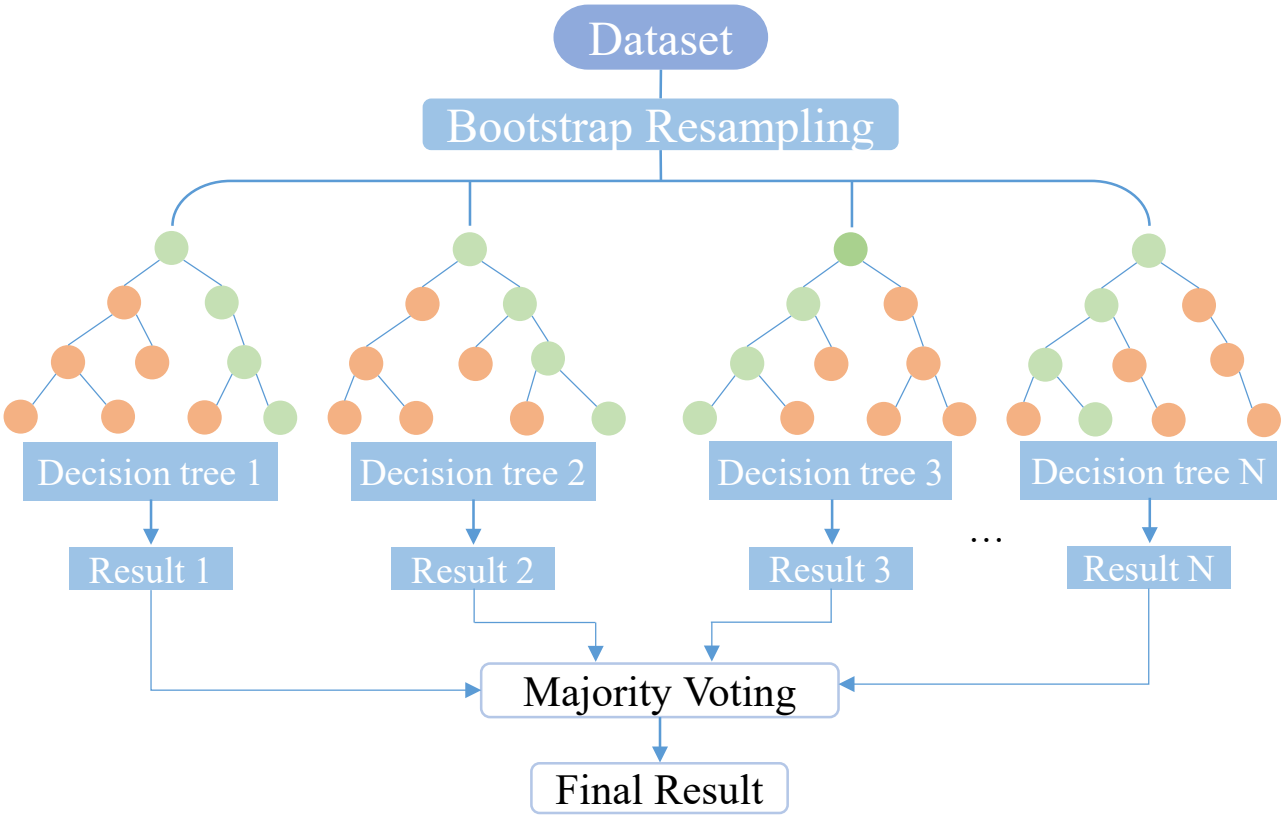


Figure 1. Schematic diagram of the random forest algorithm.

2.2. CEEMDAN Algorithm and Principle of Component Recombination

(1) CEEMDAN algorithm

The Empirical Mode Decomposition (EMD) method, introduced by Huang et al., is an adaptive data analysis approach utilized for the examination of non-linear and non-stationary data [29]. The Ensemble Empirical Mode Decomposition (EEMD) algorithm [30] introduces normally distributed white noise to the original signal based on the EMD algorithm. This addition ensures a uniform distribution of the signal across the extrema throughout the frequency band, thereby mitigating the mode mixing effect. However, the EEMD has defects associated with high computational cost and the presence of residual noise. The CEEMDAN algorithm [31] improves upon EEMD by introducing finite adaptive white noise, addressing issues related to EEMD's incompleteness and reconstruction errors after adding white noise. The decomposition steps of the CEEMDAN algorithm are as follows:

Step 1: Add $v^i(t)$, which is white noise, to the original signal $S(t)$. The signal of the ith is represented as $S^i(t) = S(t) + v^i(t)$, $i = 1,2,\ldots,I$. The experimental signal $S^i(t)$ is decomposed by EMD to obtain IMF_1^i, so $IMF_1 = \frac{1}{I}\sum_{i=1}^{I} IMF_1^i$, residual $r_1(t) = S(t) - IMF_1$.

Step 2: Add $v^i(t)$ to residual $r_1(t)$, and conduct the experiment i times (i = 1,2,. . . ,I). In each experiment, the EMD was applied to decompose $r_1^i(t) = x(t) + v^i(t)$ to obtain its first-order component IMF_1^i. $IMF_2 = \frac{1}{I}\sum_{i=1}^{I} IMF_1^i$, and residual $r_2(t) = S(t) - IMF_2$.

Step 3: Iterate the aforementioned decomposition process to acquire the IMF components satisfying the conditions along with their corresponding residuals. The program concludes when the residual exhibits monotony functions and cannot be further decomposed using EMD. The original signal can be expressed as $S(t) = \sum_{i=1}^{n} IMF_i + r_n(t)$.

(2) Principle of component recombination

The results obtained from CEEMDAN decomposition are sequentially arranged from top to bottom as high-frequency components, low-frequency components, and residual terms. Observing the morphological features of the components, it can be noted that IMF components exhibit a local symmetric characteristic. Specifically, high-frequency components have a short period, with relatively uniform data distribution and a mean approaching zero. In contrast, low-frequency IMF components exhibit a larger signal period, and their envelope is obtained by interpolating a small number of peak values, leading to significant deviations from the original signal trend, and for the low-frequency component, it is hard to ensure a mean of zero. Considering the large value of production data, a threshold of 1% of the mean production is set as a boundary. A statistical t-test is employed to examine whether the mean of each component significantly differs from this threshold, serving as a standard for delineation between high- and low-frequency components.

2.3. Temporal Convolutional Network

The TCN model can be conceptualized as a combination of 1-D fully convolutional networks and causal convolutions. This neural network architecture preserves the robust feature extraction capabilities inherent to traditional convolutional neural networks while demonstrating high efficacy in the processing and analysis of time series data. At present, many studies have shown that TCN has achieved good results in many fields such as traffic flow estimation, wind power forecasting, weather forecasting, etc. [32–35]. However, the application of TCN in oil production prediction is relatively limited.

TCN exhibits two distinctive characteristics:

(1) Causal convolution ensures that the output at a given time is solely dependent on the current time and historical inputs, devoid of any influence from future inputs.
(2) The architecture possesses the capability to map time series data of arbitrary length to output data of the same length, akin to RNN.

To fulfill the first characteristic, the initial layer of TCN is a one-dimensional fully convolutional network. In this design, each intermediate layer matches the size of the input layer, followed by zero-padding to ensure that the output dimension of the subsequent layer aligns with the dimension of the input in the previous layer. To adhere to the second property, TCN performs convolutions exclusively on the inputs from the current moment and historical time. Achieving an effective historical data span for extended periods requires large filters or extremely deep structures, resulting in substantial computational requirements. Hence, TCN employs dilated convolutions. TCN comprises three main components: causal convolution, dilated convolution, and residual connections.

(1) Causal convolution

Figure 2 illustrates the architecture of the causal convolutional network. The convolutional output at time t is solely influenced by the input at time t and preceding times, with no reliance on input beyond time t. The one-way nature of causal convolution, strictly adhering to the temporal sequence, makes this specialized structure particularly well suited for processing time series data.

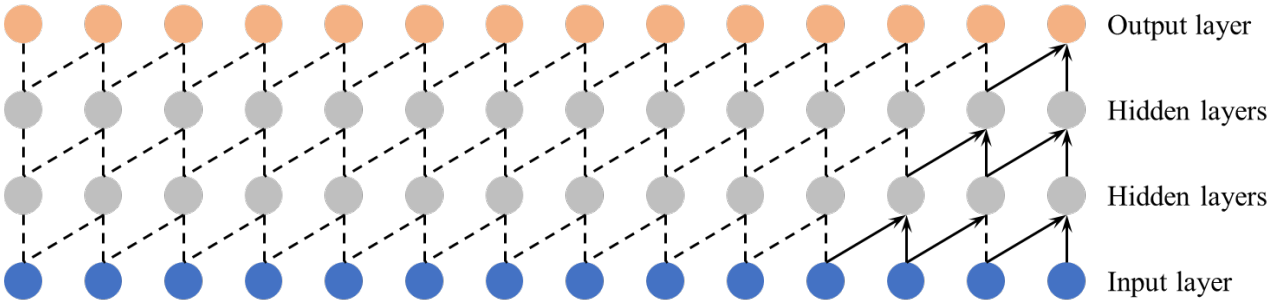


Figure 2. Schematic diagram of causal convolution structure.

(2) Dilated convolution

As the convolutional neural network's complexity increases to meet the demand for higher accuracy, the network structure becomes deeper, leading to larger computational complexity. The deep models face challenges such as vanishing gradients due to a large number of layers, resulting in decreased network performance and potential saturation [36]. To tackle the challenge of information loss related to historical data, the TCN model integrates dilated convolution with causal convolution. Figure 3 depicts the schematic architecture of the dilated convolutional network.

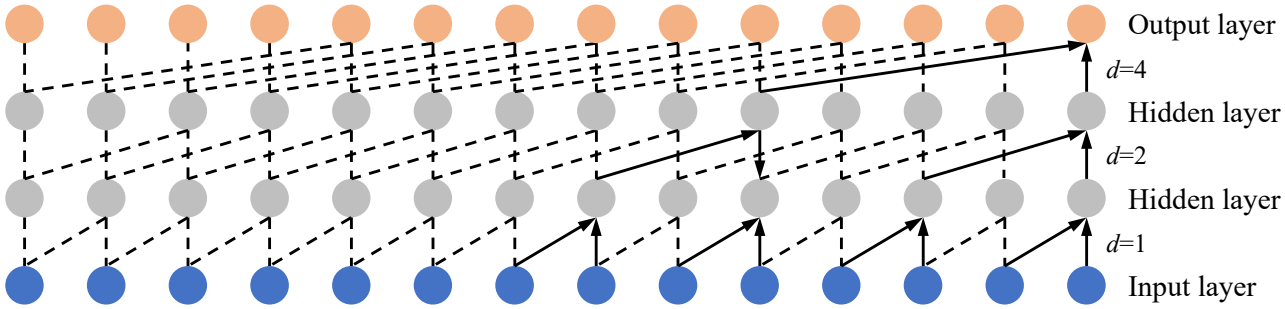


Figure 3. Schematic diagram of dilated convolution structure.

This amalgamation extends the field of view, enabling the model to capture long-range dependencies within the input sequence. This involves augmenting the convolution kernel with additional weights, ensuring the input data remain unchanged. Consequently, the network expands its capacity to observe the time series, enhancing its capability to capture intricate patterns, all while maintaining computational efficiency. The definition of dilated convolution is shown in Equation (1):

$$F(s) = (X *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot X_{s-d \cdot i} \quad (1)$$

where $*_d$ is the convolution operator, k is the size of the filter, and $s - d \cdot i$ is the past direction. $f(i)$ is the convolution kernel, $i = 0,1,2,\ldots,k-1$. Specifically, when d is set to 1, dilated convolution is equivalent to the standard convolution.

(3) Residual connection

The TCN model incorporates the use of residual blocks to mitigate issues related to information loss and instability arising from excessive network depth [37]. Its core ideas involves the incorporation of a 'jump connection' operation, allowing for the bypassing of one or more layers [38]. The complete residual module, comprising multiple dilated causal convolutions, is applied during the model construction process. Figure 4 shows the TCN residual module structure.

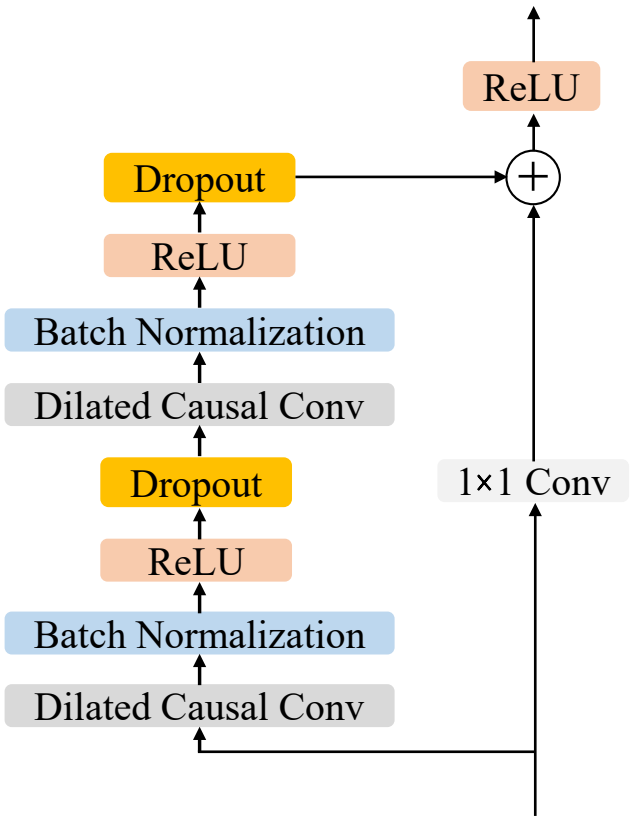


Figure 4. Residual block structure diagram.

The residual module encompasses a sequence of transformations, with one part involving the transformation of the input x through a series of overlays. The other part consists of a shortcut connection utilizing a 1×1 convolution. This shortcut connection is added to the original input through the branch to ensure that the dimensions of the original input and the output of the residual block remain consistent. The formula is shown in Equation (2).

$$O = Activation(X + F(X)) \tag{2}$$

2.4. Gated Recurrent Unit

GRU, an improved method of RNN and LSTM, incorporates improvements in the gate control structure. In contrast to LSTM, GRU consolidates the input gate and forget gate of LSTM into a single update gate (Z_t) and transforms the output gate into a reset gate (r_t) [39]. The update gate regulates the quantity of data transferred from the memory information of the previous moment to the current moment, while the reset gate controls the amount of historical information to be discarded. Figure 5 depicts the structural diagram of the GRU.

In comparison to LSTM, GRU boasts a simpler architectural design, reduced redundancy in internal units, fewer parameters, and faster computational speed. The operational expression of GRU is as provided in Equations (3)–(6).

$$Z_t = \sigma(W_Z \cdot [h_{t-1}, x_t]) \tag{3}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{4}$$

$$\widetilde{h_t} = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{5}$$

$$h_t = (1 - Z_t) * h_{t-1} + Z_t * \widetilde{h_t} \tag{6}$$

where x_t is the input at time t, h_{t-1} is the output at time $t-1$, r_t is the reset gate, σ is the activation function, W is the weight matrix, and $\widetilde{h_t}$ is the candidate hidden state.

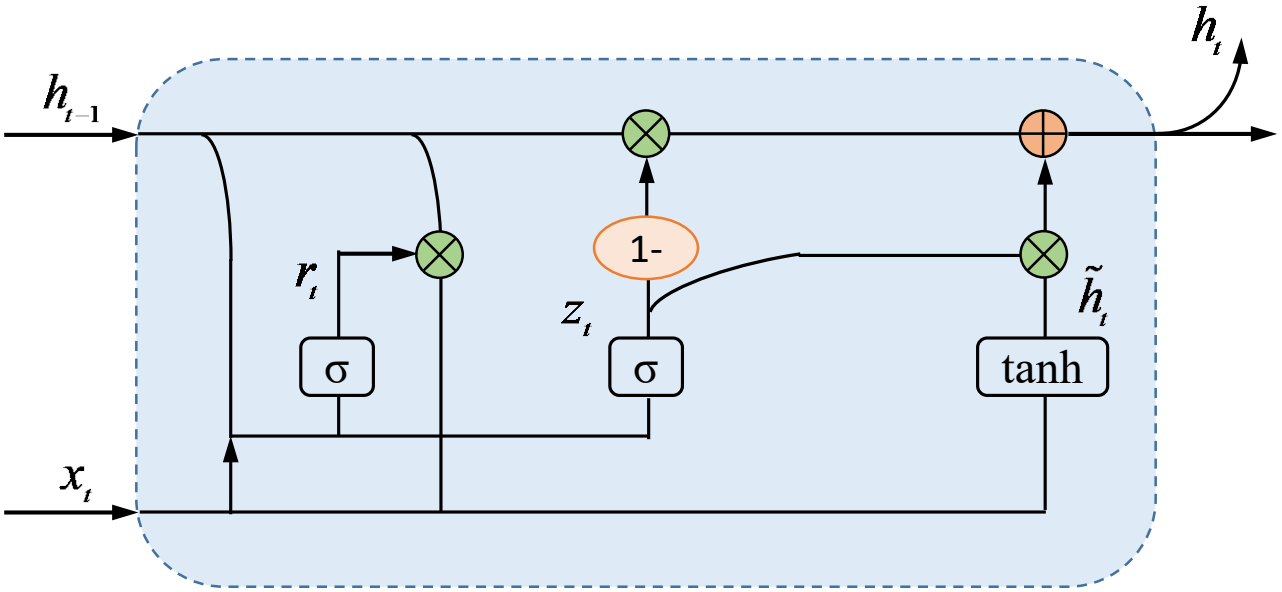


Figure 5. GRU structure diagram.

2.5. Multi-Head Attention

The attention mechanism, a crucial part of neural network architecture, assigns varying weights to input data during feature extraction. This allows the model to focus on pertinent information while reducing the impact of irrelevant elements. The input of the module includes d_Q, d_K, and d_V, dimensional queries (**Q**), keys (**K**), and values (**V**), respectively. The formula of attention mechanism is presented in Equation (7):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V} \tag{7}$$

Multi-head self-attention is an enhanced version of the attention mechanism that employs multiple parallel attention heads. These heads operate simultaneously, conducting h times separate linear projections for queries, keys, and values [40]. The attention is computed in parallel, resulting in (d_V/h) dimensional outputs for each head. These outputs are then concatenated to create the final values. It can be written as Equations (8) and (9):

$$\text{Multihead}(\mathbf{Q}, \mathbf{V}, \mathbf{K}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathrm{W}^O \tag{8}$$

$$\text{head}_\text{i} = \text{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right) \tag{9}$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d_Q/h}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_K/h}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_V/h}$ are the weight matrices for the projections, and W^O is the weight matrix.

2.6. Model Architecture and Modeling Steps

Based on the above-mentioned methods, this section will introduce the application steps and the architecture of the proposed model.

Step 1: Preprocessing of Two-Stage Production Sequence Data

Firstly, the random forest algorithm is applied to extract the most strongly correlated feature data from all possible related dynamic production data. It also serves to reduce dimensionality and mitigate overfitting. In the second stage, CEEMDAN is used to identify the noise components, production regularity components, and trend components in the production data.

Step 2: Reconstruction of Component Data

The components obtained through the CEEMDAN method consist of multiple high-frequency components, multiple low-frequency components, and a residual term. Following the principles of component recombination, multiple high-frequency components are combined to form a single high-frequency component, and multiple low-frequency components are combined to form a single low-frequency component. In this way, a total of three time series component data are obtained.

Step 3: Component data prediction and integration

Applying the TCN-GRU-MA model to each component data obtained in the second step, the production data are learnt and predicted, and then, the final production forecast result can be obtained by linear combination.

Step 4: Numerical Experiments and Model Evaluations

To validate the effectiveness and accuracy of the proposed method, various ensemble machine learning models are introduced for result prediction and comparative analysis.

In the proposed model, each module serves a unique purpose: the RF algorithm is employed to extract the primary production factors influencing crude oil yield; the CEEMDAN decomposition algorithm is used to gain a clearer understanding of the variation patterns in production data at different time scales; the TCN network is utilized to extract features along the time dimension, capturing temporal dependencies; and the MA mechanism assists the model in concentrating on the most crucial aspects of the task, disregarding relatively insignificant information. Ultimately, leveraging the time series fitting capability of the GRU neural network, we construct an integrated production forecasting model. Figure 6 shows the architecture of the RF-CEEMDAN-TGMA model.

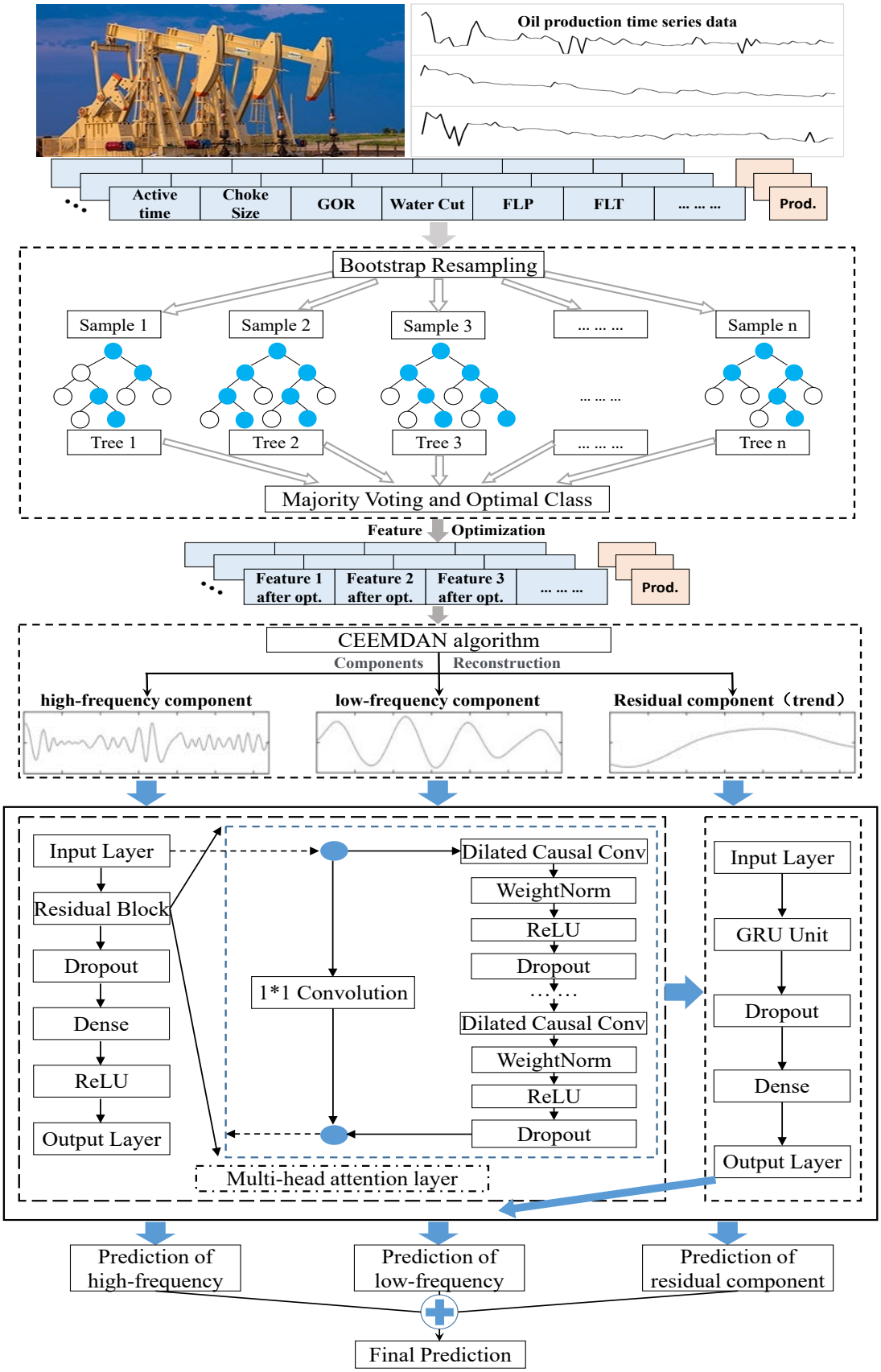


Figure 6. The architecture of the RF-CEEMDAN-TGMA model.

3. Model Evaluation Indicators

In assessing forecasting techniques, we introduce four commonly used evaluation metrics. Mean absolute error (*MAE*) quantifies the mean absolute error between predicted and observed values and is less sensitive to outliers. Root mean square error (*RMSE*) employs a quadratic loss function, making it more sensitive to larger deviations. Mean absolute percentage error (*MAPE*) excels in trend forecasting. The lower predicting error signifies a more accurate forecasting model. Meanwhile, R^2 scores gauge the fitness of the forecasting approach to the available data, with higher scores indicating a superior predictive performance. These four approaches are implemented as presented in Equation (10) through (13).

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{11}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{12}$$

$$R^2 = 1 - \sum_{i=1}^{m}\frac{(y_i - \hat{y}_i)^2}{(y_i - \overline{y})^2} \tag{13}$$

where y_i and $\hat{y}_i$ represent the actual output and the predicted output at the ith moment, respectively. In the context of this paper, y_i and $\hat{y}_i$ represent the total monthly oil production, measured in ten thousand barrels (10^4 bbl). $\overline{y}$ is the mean value of the actual data, and m denotes the data size.

4. Results and Discussions

In this section, we will validate the predictive capabilities of the proposed model using two cases study. Selecting two oil fields operated by a Chinese petroleum company in overseas, modeling and prediction studies will be conducted using dynamic historical data from monthly reports. Subsequently, a comparison of predictive accuracy will be made with other ensemble models. A PC with an Intel Core i5-12490F CPU@3.00 GHz and 16 GB of RAM is employed to run all the models.

4.1. Application in Case 1

4.1.1. Data Description

To evaluate the prediction performance of RF-CEEMDAN-TGMA model, the research will be conducted using a development well from the H oil field in the Middle East region. The production data are recorded on a monthly basis, comprising dynamic variables such as the operation months, active days, choke size, gas–oil ratio (GOR), water cut, flowline pressure (FLP), flowline temperature (FLT), tubing head pressure (THP), and monthly oil production, totaling nine variables. Table 1 shows the descriptive statistics of historical dynamic data.

4.1.2. Data Preprocessing

Table 2 lists the top five main features selected by the RF algorithm under different settings of tree depth and numbers, which are most correlated with the production data. From the results, Operation Months, Choke Size, GOR, and Active Days are the stable top four features. Therefore, these four features are selected as input data for the model.

Table 1. Statistical value of variables used in Case 1.

		Maximum Value	Minimum Value	Average Value	Standard Deviation	Coefficient of Variation	Kurtosis	Skewness
The target variable	Monthly oil production (10^4 bbl)	13.2	1.6	6.0	2.1	0.3	0.3	0.3
The input data	Operation months (m)	118.0	1.0	59.5	34.2	0.6	−1.2	0.0
	Active days (d)	31.0	12.8	29.0	3.7	0.1	8.2	−2.8
	Choke Size (in)	64.0	32.0	52.3	6.9	0.1	0.0	0.0
	GOR (m^3/m^3)	829.1	438.5	628.2	84.5	0.1	−0.1	0.2
	Water Cut (%)	4.7	0.0	0.7	1.1	1.6	4.1	2.1
	FLP (Psi)	220.0	145.0	181.9	22.3	0.1	−1.0	−0.5
	FLT (°C)	66.0	40.0	57.0	6.2	0.1	0.0	−0.9
	THP (Psi)	1051.0	280.0	508.2	163.7	0.3	3.4	1.9

Table 2. The top five features selected through RF algorithm in Case 1.

Depth of the Tree	Number of Trees	The Top Five Features (Importance)				
5	50	Operation Months (3.10)	Choke Size (1.16)	GOR (0.47)	FLP (0.36)	Active Days (0.25)
	100	Operation Months (3.21)	Choke Size (1.12)	GOR (0.40)	THP (0.25)	Active Days (0.23)
	150	Operation Months (2.97)	Choke Size (1.04)	GOR (0.43)	FLP (0.31)	Active Days (0.26)
10	50	Operation Months (3.73)	Choke Size (1.26)	GOR (0.53)	Active Days (0.44)	FLP (0.38)
	100	Operation Months (3.45)	Choke Size (1.15)	GOR (0.49)	Active Days (0.48)	FLT (0.36)
	150	Operation Months (3.19)	Choke Size (1.07)	GOR (0.50)	Active Days (0.41)	THP (0.39)
15	50	Operation Months (3.90)	Choke Size (1.22)	GOR (0.61)	Active Days (0.57)	FLT (0.43)
	100	Operation Months (3.38)	Choke Size (1.13)	GOR (0.64)	Active Days (0.55)	FLT (0.45)
	150	Operation Months (3.15)	Choke Size (1.04)	GOR (0.62)	Active Days (0.57)	THP (0.47)

Next, we use the CEEMDAN algorithm to preprocess the original data. The default parameters for CEEMDAN are set as follows: Gaussian white noise amplitude, ensemble number, and maximum iteration times are set to 0.2, 500, and 5000 [41], respectively. Figure 7 shows the decomposition results.

According to the principle of component recombination, a statistical *t*-test is conducted for each component, and the results are listed in Table 3. The *p*-value for IMF_1 is greater than 0.05, indicating acceptance of the null hypothesis. It means there is no significant difference between IMF_1 and the test values, thus identifying it as a high-frequency component. However, starting from IMF_2, the *p*-value is less than 0.05, indicating that all components except for the residue are low-frequency components.

Table 3. The *t*-test values for partial components of Case 1.

T-Test	Test Value = 602				
	Size	t	Prob.	Mean Value of IMF_x	Standard Deviation
IMF_1	118	−0.10	0.918	537.99	6730.64
IMF_2	118	−4.83	0.000	−47.37	1461.56
IMF_3	118	−2.01	0.046	−93.57	3750.78
...	...	...	...	...	...

By linearly summing all low-frequency components (with a coefficient of one), the original production data were effectively partitioned into three parts based on high-frequency, low-frequency, and residual components. Figure 8 illustrates the combined curves of the three components after recombination.

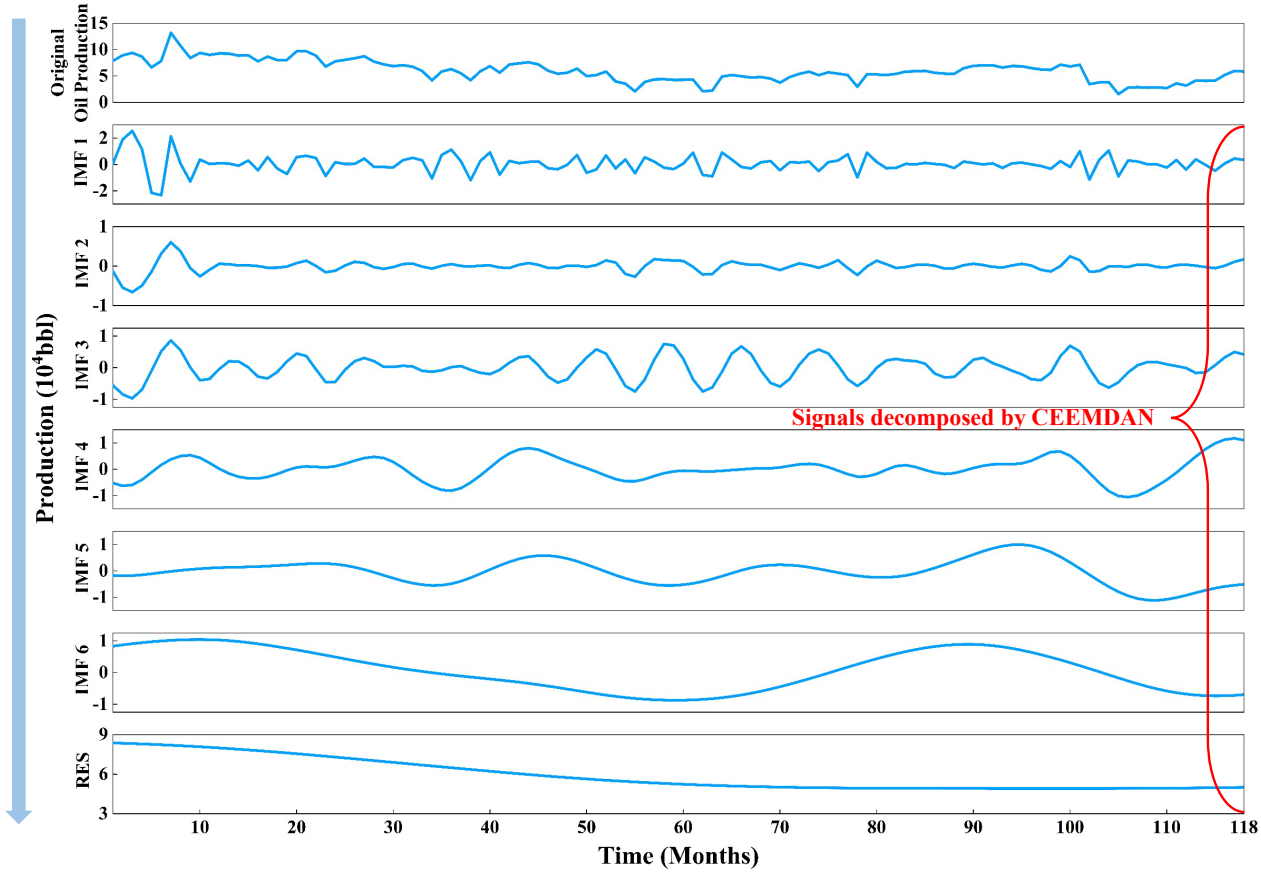


Figure 7. The IMFs, from high frequency to low frequency, of the CEEMDAN method in Case 1, distinguished by the recombination standards of the components mentioned in Section 2.2. 'RES' represents the residual term, reflecting the trend of the production.

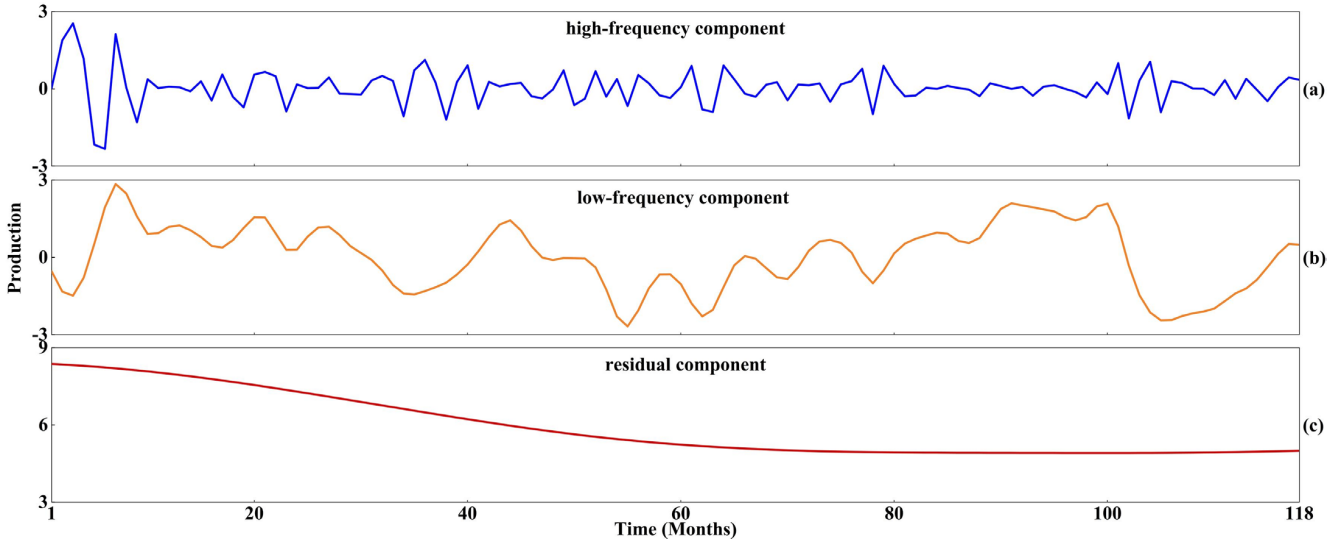


Figure 8. Production component based on the principle of the component recombination. According to Table 3, the high-frequency (**a**) component is IMF1, the low-frequency component (**b**) is the sum of IMF2 to IMF6, and the residual component (**c**) is represented by RES.

4.1.3. Parameters of the Models

In accordance with extensive experimentation and prior research [42–44], the parameters of each component are set as below:

(1) High-frequency component: For the TCN module, the number of filters is 100, the kernel size is set to 2, and dilations = [1, 2, 4]. For the GRU module, the number of hidden units is 100. The dropout is set to 0.4. Adam is chosen as the optimizer. The batch size is 64, the maxepochs is 600, and the original learning rate is 0.001.
(2) Low-frequency component: For the TCN module, the number of filters is 90, the kernel size is set to 3, and dilations = [1, 2, 4]. For the GRU module, the number of hidden units is 80. The dropout is set to 0.5. Adam is chosen as the optimizer. The batch size is 32, the maxepochs is 600, and the original learning rate is 0.001.
(3) Residual component: For the TCN module, the number of filters is 96, the kernel size is set to 3, and dilations = [1, 2, 4]. For the GRU module, the number of hidden units is 16. The dropout is set to 0.5. Adam is chosen as the optimizer. The batch size is 32, the maxepochs is 600, and the original learning rate is 0.001.

4.1.4. Experimental Results and Discussions

In this section, the proposed model is utilized along with the processed data in Section 4.1.2 and the parameters from Section 4.1.3 to model and forecast the production. Due to the relatively small dataset, the first 90% of the data are allocated as the training set, while the remaining 10% is designated as the testing set to ensure an ample amount of training data for the model. Figure 9 illustrates the predicted values of production components and the final summation result.

To assess the learning capability of the proposed model, it was compared with other twelve existing machine learning models for production forecasting accuracy, with each group of models undergoing multiple experiments. Among these, CNN-GRU [45], CNN-LSTM [46], and CNN-BILSTM [47] were previously proposed by researchers and have demonstrated good predictive performance. Additionally, we introduced RF and attention mechanism into the model for a fair comparison with the proposed model in this paper. Table 4 lists the best-performing results of each model during the experimental period. We recorded the runtime of different models to compare their operational efficiency. Figure 10 visually illustrates the prediction curves of the proposed model compared to other models.

Table 4. Error metrics and model execution time of RF-CEEMDAN-TGMA and other comparative models in Case 1.

Model	Error Metrics of Training Set				Error Metrics of Testing Set				Times (s)
	RMSE (10^4 bbl)	MAE (10^4 bbl)	MAPE	R^2	RMSE (10^4 bbl)	MAE (10^4 bbl)	MAPE	R^2	
CNN-GRU	0.447	0.313	0.052	0.949	0.738	0.639	0.157	0.573	6.98
CNN-LSTM	0.575	0.385	0.063	0.917	0.728	0.608	0.147	0.584	7.18
CNN-BILSTM	0.597	0.410	0.069	0.911	0.748	0.577	0.156	0.562	7.32
RF-CNN-GRU	0.630	0.397	0.064	0.900	0.667	0.570	0.174	0.651	6.81
RF-CNN-LSTM	0.637	0.442	0.073	0.898	0.669	0.576	0.142	0.650	6.92
RF-CNN-BILSTM	0.617	0.389	0.064	0.905	0.664	0.547	0.172	0.654	7.30
CNN-GRU-MA	0.538	0.368	0.061	0.927	0.693	0.565	0.136	0.623	6.83
CNN-LSTM-MA	0.654	0.427	0.069	0.893	0.697	0.607	0.183	0.620	7.38
CNN-BILSTM-MA	0.537	0.358	0.058	0.928	0.698	0.589	0.146	0.618	7.47
RF-CNN-GRU-MA	0.623	0.393	0.063	0.903	0.638	0.550	0.166	0.681	6.87
RF-CNN-LSTM-MA	0.653	0.429	0.070	0.893	0.646	0.565	0.170	0.673	6.98
RF-CNN-BILSTM-MA	0.621	0.387	0.063	0.903	0.645	0.545	0.168	0.674	7.30
Proposed Approach	0.319	0.256	0.049	0.974	0.619	0.541	0.145	0.699	99.2

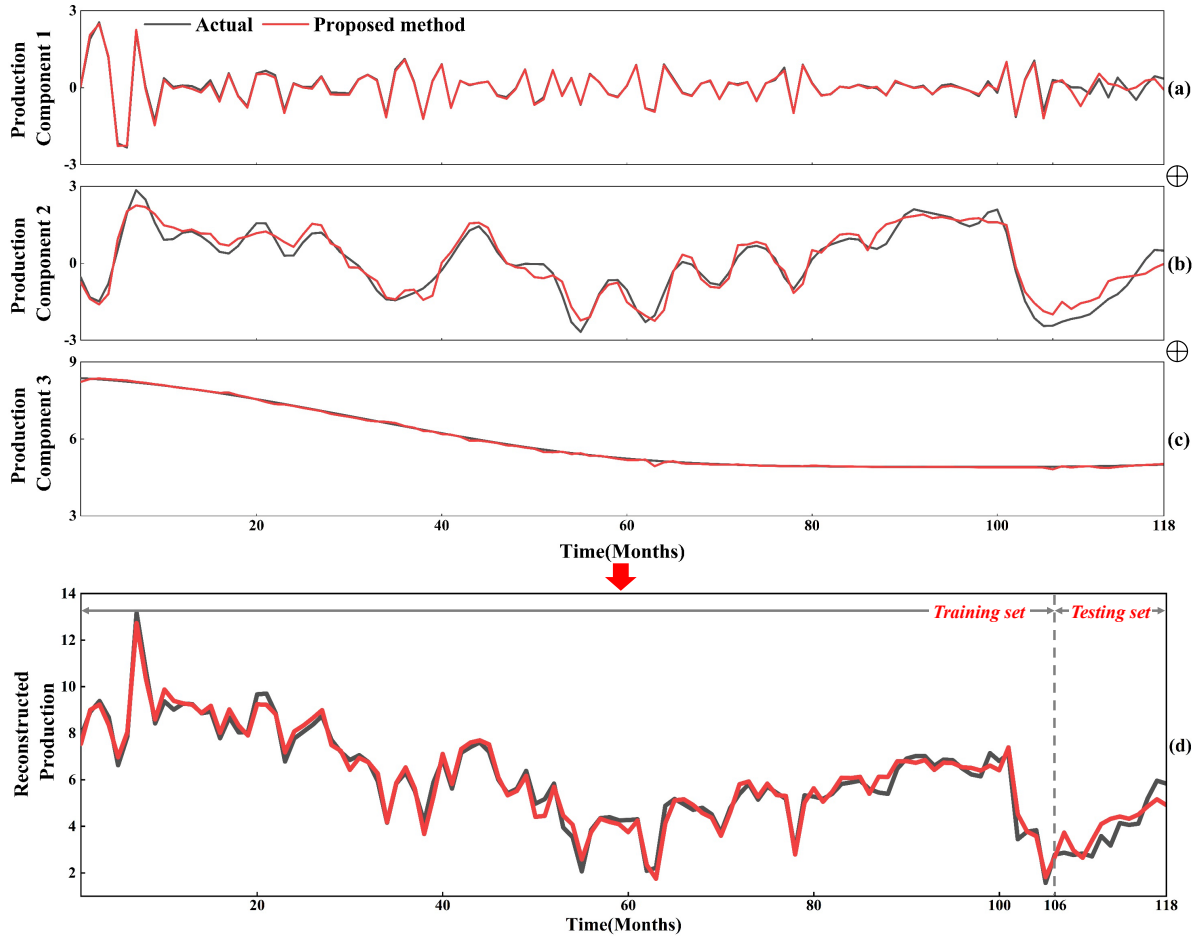


Figure 9. The forecasted results for Case 1 based on the RF-CEEMDAN-TGMA model. The actual production, represented by the black curve, and the predicted production, indicated by the red curve, are obtained by aggregating the yields from (**a**–**c**), and (**d**) represents the summation of the production from (**a**) to (**c**).

It is worth noting that the proposed RF-CEEMDAN-TGMA model outperforms other comparative models. While there is little difference in the fitting performance of all comparative models on the training set, the proposed model demonstrates a significant improvement in fitting performance. For instance, considering the R^2 metric, the CNN-GRU model achieves the highest score among the comparative models at 0.949, while the proposed model achieves a score of 0.974, resulting in a 2.6% improvement. In the testing set, due to the CNN-GRU, CNN-LSTM, and CNN-BILSTM models solely employing simple ensemble strategies without preprocessing the input data or incorporating advanced attention mechanisms, their performance across all four metrics is relatively poor, with average RMSE, MAE, MAPE, and R^2 values of 0.738, 0.608, 0.153, and 0.573, respectively. We utilize the mean values of these three models as the baseline for subsequent comparisons. Considering the prediction models derived by integrating the RF algorithm for feature selection, namely, RF-CNN-GRU, RF-CNN-LSTM, and RF-CNN-BILSTM models, aside from a slight variation in the MAPE metric, the remaining three metrics on the testing set exhibit significant improvements compared to the baseline. The mean RMSE decreases by 9.6%, the mean MAE decreases by 7.2%, and the mean R^2 increases by 13.8%. The models incorporating only the MA mechanism, namely, the CNN-GRU-MA, CNN-LSTM-MA, and CNN-BILSTM-MA models, show only marginal improvement in predictive performance, with a mean RMSE reduction of 5.7%, a mean MAE reduction of 3.5%, and a mean R^2 increase of 8.2%. The models that combine the RF algorithm and the MA mechanism, namely, the RF-CNN-GRU-MA, RF-CNN-LSTM-MA, and RF-CNN-BILSTM-MA models,

achieve the best performance among all comparative models. The mean RMSE decreases by 12.9%, the mean MAE decreases by 9%, and the mean R^2 increases by 18%. Among them, the RF-CNN-GRU-MA model achieves the highest accuracy, with an R^2 of 0.681. The model proposed in this study, namely, RF-CEEMDAN-TGMA, not only combines the RF algorithm and attention mechanism but also utilizes the CEEMDAN algorithm to decompose and reassemble production data into three sets with different frequencies, which clearly reflect production characteristics. Additionally, it enhances the model's ability to learn the inherent relationships within the data. Therefore, it exhibits better predictive performance and achieves superior metrics compared to the best-performing comparative model (RF-CNN-GRU-MA), with a reduction in RMSE by 3% to 0.619, a decrease in MAE by 1.6% to 0.541, a decrease in MAPE by 12.7% to 0.145, and an increase in R^2 by 2.6% to 0.699. Furthermore, numerical experiments also indicate that a good fitting performance on the training set does not guarantee an ideal predictive performance. For instance, the CNN-GRU model exhibits good performance on the training set but performs the worst on the testing set. This suggests that the model is susceptible to interference from noise, weakly correlated features, and other factors during the learning process, thereby impacting its generalization performance. In contrast, the proposed model achieves the highest scores on both the training set and testing set, indicating the effectiveness of feature selection, data preprocessing, and other modules. These factors assist the model in truly learning the intrinsic relationships between the data during training, resulting in good generalization performance and ensuring ideal predictive outcomes. From an efficiency standpoint, given that the proposed model is applied to three sets of data, the recorded running time is three times higher. However, it remains highly efficient in comparison to traditional dynamic prediction methods, such as numerical simulation. For oilfield production technicians, it remains an efficient technological approach.

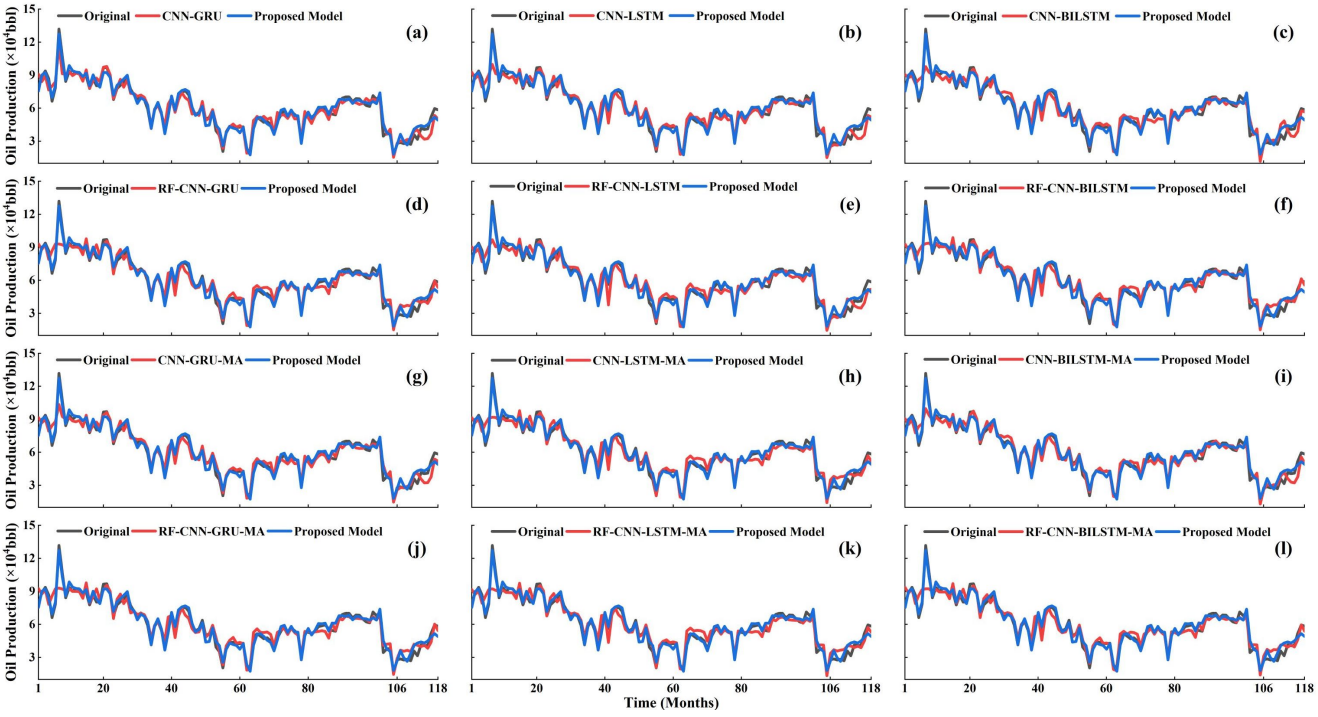


Figure 10. Performance comparison between the proposed model and other twelve models in Case 1. Each row represents a comparative analysis of prediction experiments, where the base model is fused with different functional modules. Each column (**a**,**d**,**g**,**j**) & (**b**,**e**,**h**,**k**) & (**c**,**f**,**i**,**l**) illustrates a comparison of prediction experiment results among different base models.

4.2. Application in Case 2

In case 1, we validated the performance of the theoretical model using historical data. In this section, we will simulate real-world production by predicting future production with unknown input data, testing the model's performance in practical applications.

4.2.1. Data Description

In this case, production forecasting will be conducted for the carbonate reservoirs of Block A in the onshore Caspian Sea region, originating from the Central Asian country. The production data are also recorded on a monthly basis, comprising dynamic variables such as the operation months, average active days, water injection, water cut, oil wells, water injection wells, total wells, injection–production ratio, GOR, and monthly oil production, totaling ten variables. Table 5 shows the descriptive statistics of historical dynamic data.

Table 5. Statistical value of variables used in Case 2.

		Maximum Value	Minimum Value	Average Value	Standard Deviation	Coefficient of Variation	Kurtosis	Skewness
The target variable	Monthly oil production (10^4 bbl)	25.9	4.31	15.09	5.53	0.37	−1.02	−0.05
The input data	Operation months (m)	279.00	1.00	140.00	80.68	0.58	−1.20	0.00
	Average active days (d)	31.00	23.00	30.20	1.36	0.05	7.72	−2.61
	Water injection (10^4 bbl)	37.32	9.63	23.29	6.19	0.27	−0.38	0.15
	Water cut (%)	52.10	0.19	16.59	14.24	0.86	−0.85	0.54
	Oil wells	61.00	19.00	42.89	13.15	0.31	−1.03	−0.70
	Water injection wells	20.00	2.00	9.24	4.63	0.50	−0.14	0.80
	Total wells	71.00	23.00	52.13	16.61	0.32	−1.15	−0.68
	Injection–production radio	1.92	0.40	0.73	0.32	0.44	2.38	1.69
	GOR (m^3/m^3)	1.00	0.22	0.66	0.28	0.42	−1.52	−0.13

4.2.2. Data Preprocessing

Table 6 lists the top five main features selected by the RF algorithm under different settings of tree depth and numbers, which are most correlated with the production data. From the results, GOR, operation months, total wells, and water injection are the stable top four features. Therefore, these four features are selected as input data for the model.

Table 6. The top five features selected through RF algorithm in Case 2.

Depth of the Tree	Number of Trees	The Top Five Features (Importance)				
5	50	GOR (1.27)	Operation Months (1.15)	Total wells (0.80)	Water Injection (0.45)	Water Injection Wells (0.43)
	100	GOR (1.33)	Operation Months (1.08)	Total wells (0.84)	Water Injection (0.45)	Water Injection Wells (0.39)
	150	GOR (1.39)	Operation Months (1.13)	Total wells (0.78)	Water Injection (0.43)	Water Injection Wells (0.40)
10	50	GOR (1.54)	Operation Months (1.25)	Total wells (0.80)	Injection–Production Ratio (0.55)	Water Injection (0.53)
	100	GOR (1.52)	Operation Months (1.15)	Total wells (0.81)	Water Injection (0.57)	Injection–Production Ratio (0.44)
	150	GOR (1.57)	Operation Months (1.19)	Total wells (0.77)	Water Injection (0.54)	Injection–Production Ratio (0.41)
15	50	GOR (1.58)	Operation Months (1.26)	Total wells (0.81)	Injection–Production Ratio (0.58)	Water Injection (0.56)
	100	GOR (1.59)	Operation Months (1.16)	Total wells (0.83)	Water Injection (0.59)	Injection–Production Ratio (0.48)
	150	GOR (1.60)	Operation Months (1.20)	Total wells (0.77)	Water Injection (0.57)	Injection–Production Ratio (0.46)

Next, we use the CEEMDAN algorithm to preprocess the original data. The parameter settings for the CEEMDAN algorithm are consistent with Case 1. Figure 11 shows the decomposition results.

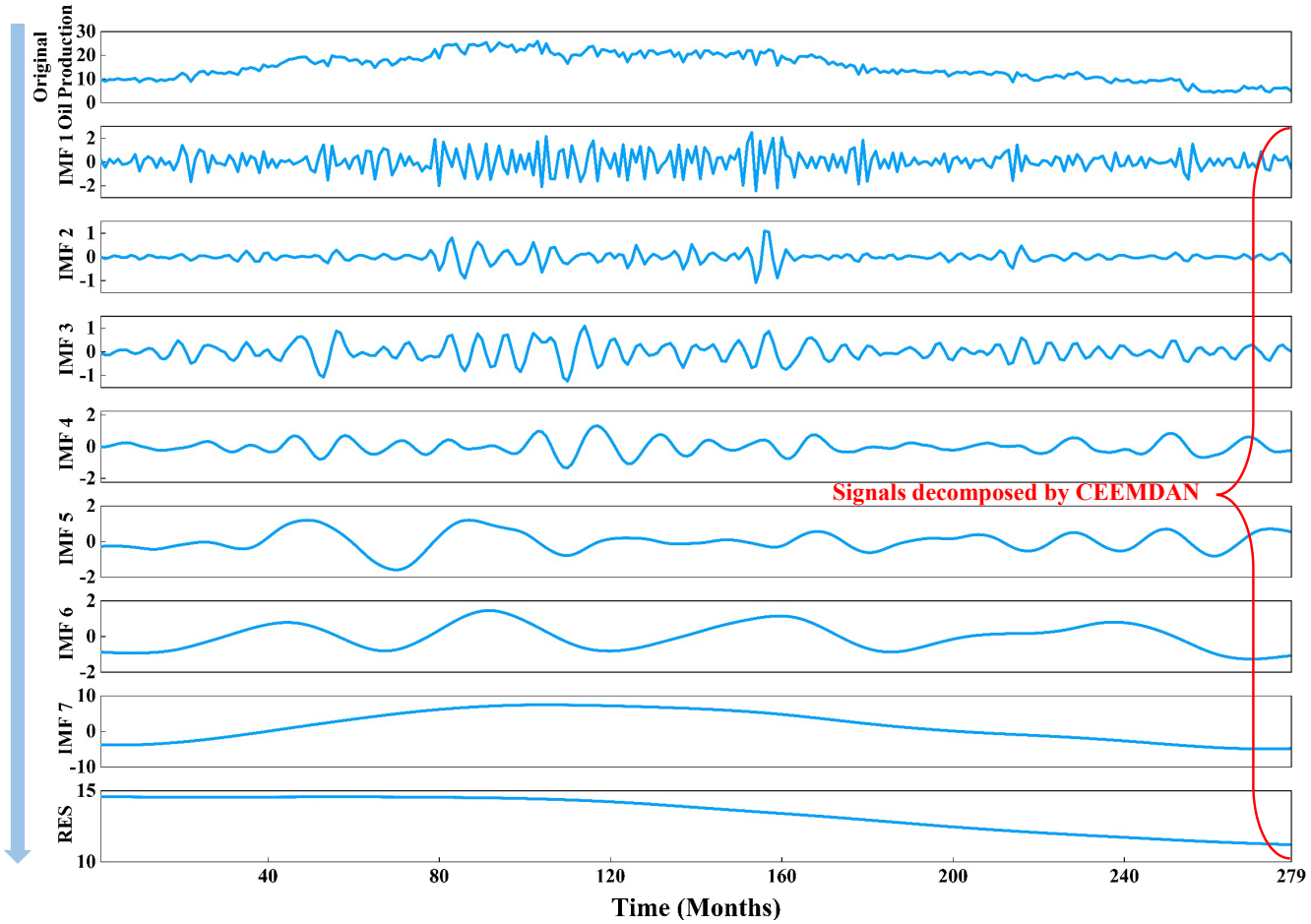


Figure 11. The IMFs, from high frequency to low frequency, of the CEEMDAN method in Case 2, distinguished by the recombination standards of the components mentioned in Section 2.2. 'RES' represents the residual term, reflecting the trend of the production.

To distinguish between high-frequency and low-frequency components, a statistical *t*-test is conducted for each component as in Case 1, and the results are listed in Table 7. The *p*-value for IMF1 is greater than 0.05 and indicates that it is a high-frequency component. The *p*-values for IMF2 to IMF7 are less than 0.05 and indicate that they are all low-frequency components.

Table 7. The *T*-test values for partial components of Case 2.

***T*-Test**	**Test Value = 1509**				
	Size	**t**	**Prob.**	**Mean Value of IMF_x**	**Standard Deviation**
IMF_1	279	−1.96	0.051	470.19	8758.74
IMF_2	279	−11.19	<0.001	−10.26	2253.51
IMF_3	279	−6.66	<0.001	−52.43	3631.38
...	...	...	...	...	...

By linearly summing all low-frequency components (with a coefficient of one), the original production data were effectively partitioned into three parts based on high-frequency, low-frequency, and residual components. Figure 12 illustrates the combined curves of the three components after recombination.

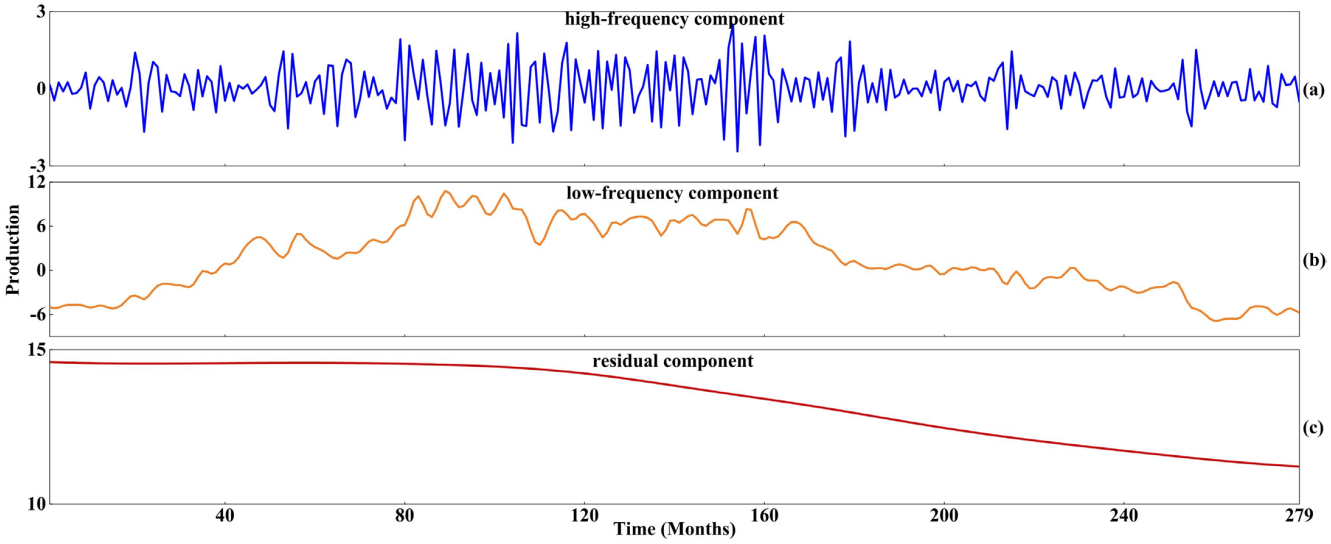


Figure 12. Production component based on the principle of the component recombination. According to Table 3, the high-frequency (**a**) component is IMF1, the low-frequency component (**b**) is the sum of IMF2 to IMF7, and the residual component (**c**) is represented by RES.

4.2.3. Parameters of the Models

The parameters of each component in Case 2 are set as below:

(1) High-frequency component: For the TCN module, the number of filters is 32, the kernel size is set to 3, and dilations = [1, 2, 4]. For the GRU module, the number of hidden units is 60. The dropout is set to 0.5. Adam is chosen as the optimizer. The batch size is 32, the maxepochs is 600, and the original learning rate is 0.001.

(2) Low-frequency component: For the TCN module, the number of filters is 32, the kernel size is set to 3, and dilations = [1, 2, 4]. For the GRU module, the number of hidden units is 32. The dropout is set to 0.2. Adam is chosen as the optimizer. The batch size is 32, the maxepochs is 600, and the original learning rate is 0.001.

(3) Residual component: For the TCN module, the number of filters is 32, the kernel size is set to 3, and dilations = [1, 2, 4]. For the GRU module, the number of hidden units is 64. The dropout is set to 0.5. Adam is chosen as the optimizer. The batch size is 32, the maxepochs is 600, and the original learning rate is 0.001.

4.2.4. Experimental Results and Discussions

This section will utilize the proposed model for production forecasting. First and foremost, our task is to address the determination of future input parameters. In theory, accurately obtaining future data are nearly impossible. However, in oil field production, a significant amount of dynamic data often exhibits certain regularities. Therefore, based on fitting historical production data, we will extrapolate the curve to obtain future production data for a certain period of time to validate the training effectiveness of the proposed model. The specific approach is as follows: use the training set's input data for curve fitting, and then extend the curve to the endpoint of the testing set to obtain corresponding 'future input data.' After replacing the corresponding data in the testing set with this batch of data, the input data for the testing set becomes completely new for the model, making the predictive experiment more valuable.

Due to the larger dataset in Case 2, the first 80% of the data are allocated to the training set, while the remaining 20% is designated as the testing set. The fitting and prediction results of the four sets of most relevant features selected by the RF algorithm are shown in Figure 13. The blue solid line represents the predicted results.

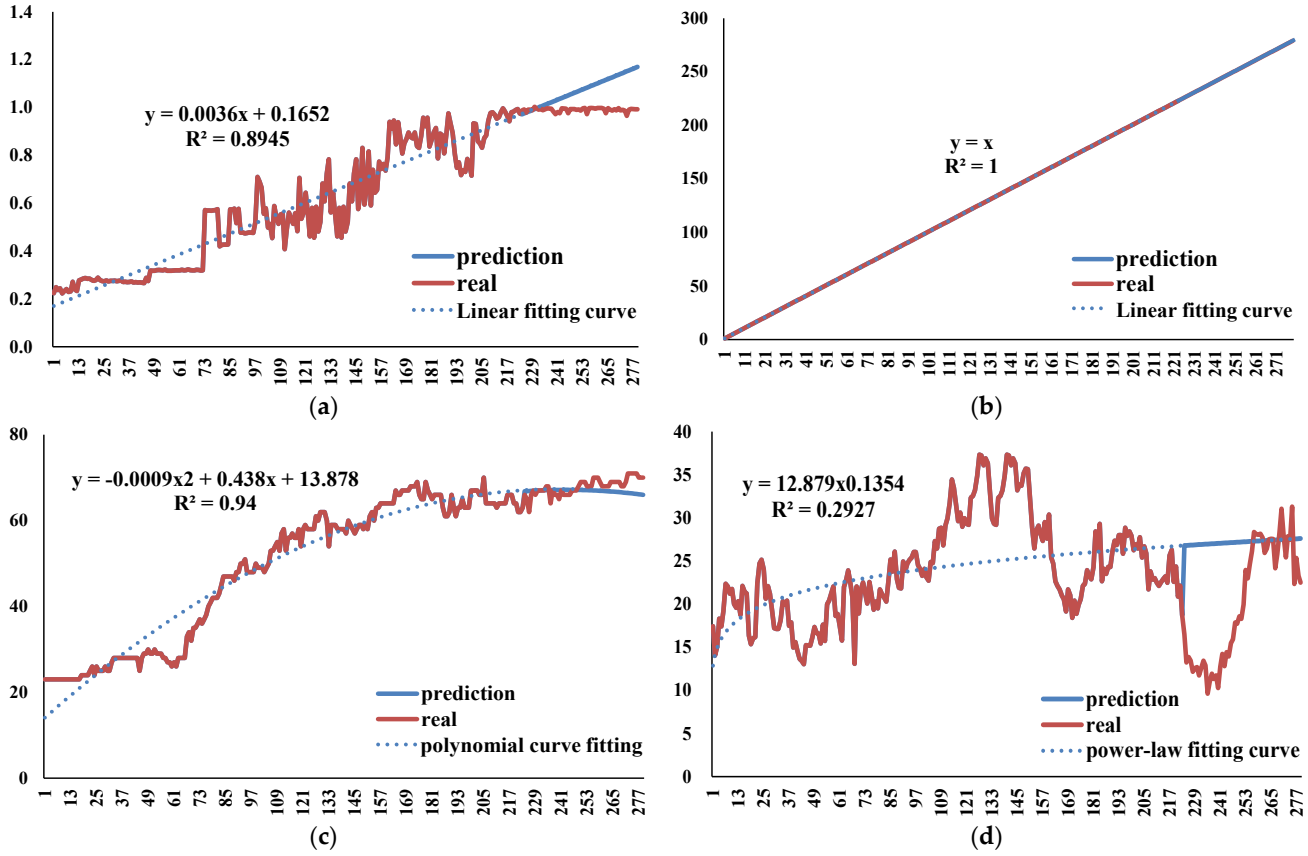


Figure 13. Fitting graph of historical input data in the training set for obtaining future input data in the testing set. Figure (**a**–**d**): fitting and prediction plots for GOR data, operation month data, total wells data, and water injection data, respectively.

Building upon Sections 4.2.2 and 4.2.3, and the predictive work on the testing set inputs in this section, we ultimately apply the proposed model to Case 2. Figure 14 presents the predicted values for each production component, along with their linear summation to obtain the total production (where each production component has a coefficient of one for the summation). Since the predictions are entirely based on unknown data, the results are more valuable for validation purposes. Additionally, it can be clearly observed that, unlike in Case 1, the predicted results no longer exhibit pronounced nonlinear variations but rather display smooth curves. This is attributed to the utilization of smoothed input data, aligning the predicted results with the features of the inputs.

To enhance the interpretability of the model, we employed the SHAP toolkit to calculate the contributions of each feature to the predicted results of each production component. Figure 15 presents the feature contribution ranking plots for the three production components. The results indicate that, for the first component, the feature contribution ranking is water injection, GOR, operation time, and total wells, suggesting that water injection is the primary influencing factor for the high-frequency fluctuations in production. However, the contribution of other factors is not negligible, indicating that high-frequency fluctuations are the result of the combined effects of multiple factors. For the second and third components, operation time consistently ranks first with a significant contribution, indicating a strong correlation between oilfield production and time, and showing a clear decreasing trend over time.

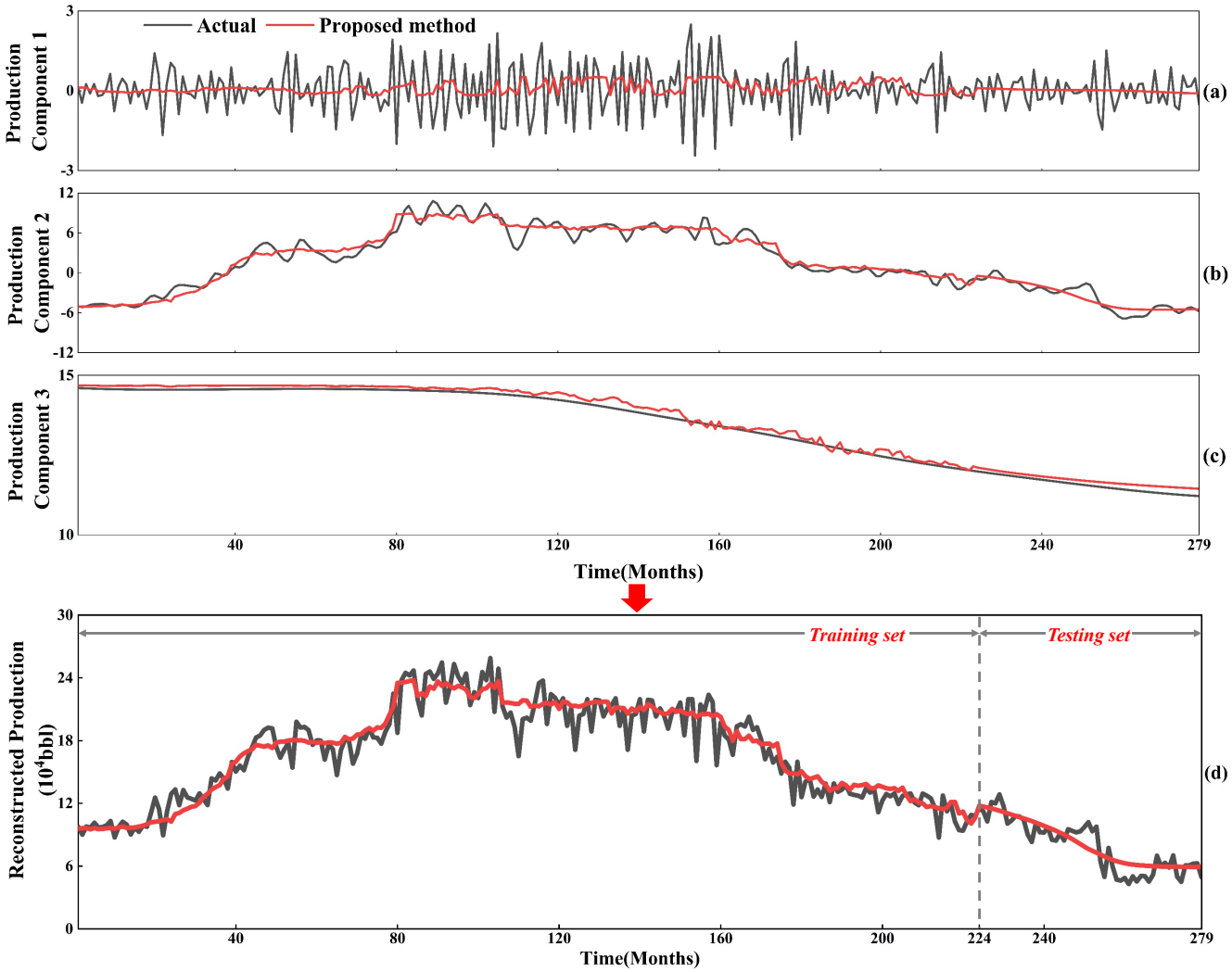
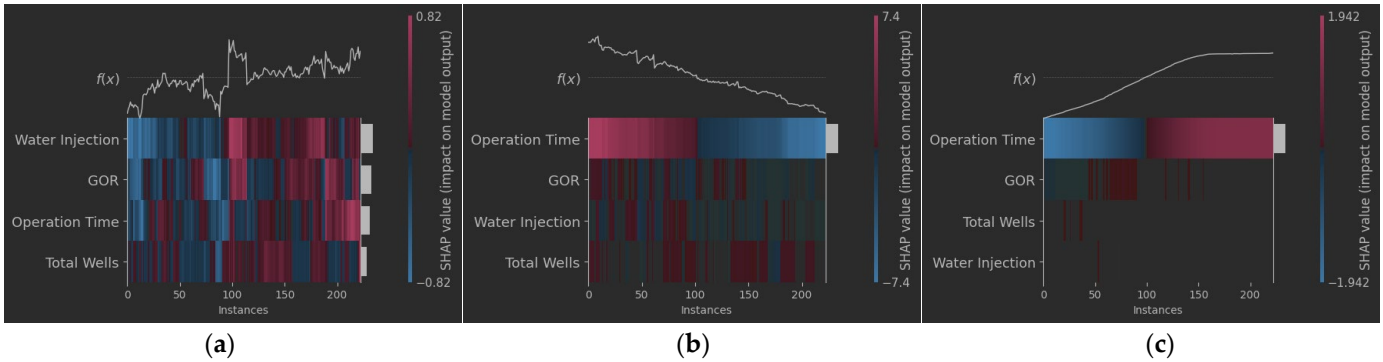


Figure 14. The forecasted results for Case 2 based on the RF-CEEMDAN-TGMA model. The actual production, represented by the black curve, and the predicted production, indicated by the red curve, are obtained by aggregating the yields from (**a**–**c**), and (**d**) represents the summation of the production from (**a**) to (**c**).



(**a**) (**b**) (**c**)

Figure 15. Ranking plot of the importance contributions of input features to the predicted results based on the SHAP toolkit. (**a**–**c**) depict the ranking results of feature contributions for production component 1, production component 2, and production component 3, respectively.

Similarly, we compare the results of this model with those of the other 12 models to assess the performance of the model. These models align with those used in Case 1. For the sake of fairness, we performed fitting on the remaining five features and obtained the input data for the testing set, serving as a comparative model without using RF algorithm. Table 8 lists the best-performing results of each model during the experimental period. At the same time, we also recorded the runtime of different models to compare their operational efficiency.

Based on the results, the proposed model continues to demonstrate stable and excellent performance in the second set of experiments. Except for the RF-CNN-BILSTM-MA model, which achieved an R^2 value of 0.753, the R^2 values for other models were all below 0.75. From the results on the testing set, comparing with the best reference model, the proposed model shows a reduction of approximately 7.7% in RMSE, a decrease of about 7.7% in MAE, a decrease of approximately 11.6% in MAPE, and an improvement of around 4.7% in R^2.

Figure 16 visually illustrates the prediction curves of the proposed model compared to other models. It can be observed that the nonlinearity in the predicted results of most comparative models is not significant, with only a few models showing some variations in the predicted curves. In contrast, the proposed model predicts yields at different time scales and, when aggregated, the predicted results reflect a trend closer to the actual production changes, further confirming the effectiveness of the proposed model. Additionally, it is important to note that the inputs for this set of experiments are predicted based on the historical trends of the data. In oil field production, there is often a clear plan for future production systems, such as plans for new well and water injection. This allows for obtaining more accurate 'future data'. We believe that utilizing such data in practical applications will lead to even better predictive results.

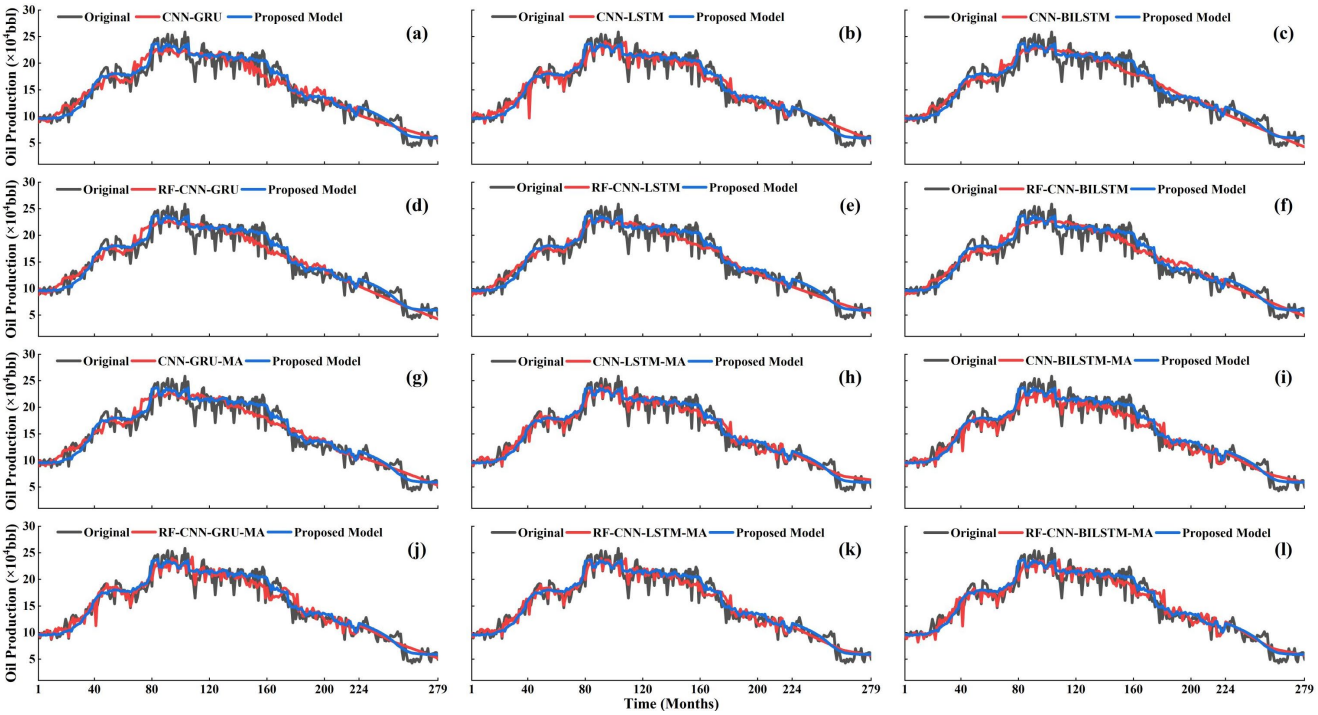


Figure 16. Performance comparison between the proposed model and other twelve models in Case 2. Each row represents a comparative analysis of prediction experiments, where the base model is fused with different functional modules. Each column (**a**,**d**,**g**,**j**) & (**b**,**e**,**h**,**k**) & (**c**,**f**,**i**,**l**) illustrates a comparison of prediction experiment results among different base models.

Table 8. Error metrics and model execution time of RF-CEEMDAN-TGMA and other comparative models in Case 2.

Model	Error Metrics of Training Set				Error Metrics of Testing Set				Times (s)
	RMSE (10^4 bbl)	MAE (10^4 bbl)	MAPE	R^2	RMSE (10^4 bbl)	MAE (10^4 bbl)	MAPE	R^2	
CNN-GRU	1.441	1.785	0.088	0.846	1.232	1.491	0.183	0.635	7.11
CNN-LSTM	1.358	1.021	0.062	0.911	1.431	1.101	0.182	0.664	8.03
CNN-BILSTM	1.642	1.309	0.080	0.869	1.427	1.237	0.168	0.666	7.90
RF-CNN-GRU	1.728	1.390	0.084	0.855	1.450	1.258	0.169	0.654	6.78
RF-CNN-LSTM	1.554	1.229	0.074	0.883	1.422	1.120	0.173	0.668	7.31
RF-CNN-BILSTM	1.733	1.400	0.086	0.855	1.298	1.089	0.162	0.723	7.51
CNN-GRU-MA	1.743	1.405	0.085	0.853	1.378	1.108	0.173	0.688	7.50
CNN-LSTM-MA	1.321	1.028	0.062	0.916	1.342	1.088	0.174	0.704	8.19
CNN-BILSTM-MA	1.499	1.154	0.067	0.891	1.312	1.052	0.165	0.717	8.33
RF-CNN-GRU-MA	1.422	1.080	0.065	0.902	1.294	1.051	0.164	0.725	7.23
RF-CNN-LSTM-MA	1.318	1.027	0.062	0.916	1.244	1.028	0.154	0.746	7.47
RF-CNN-BILSTM-MA	1.453	1.106	0.066	0.898	1.226	1.004	0.155	0.753	7.80
Proposed Approach	1.387	1.055	0.066	0.907	1.131	0.927	0.137	0.788	103.5

5. Conclusions

In this study, we propose an integrated production forecasting model, RF-CEEMDAN-TGMA, which combines data preprocessing techniques with attention mechanism. The model integrates the RF algorithm, CEEMDAN algorithm, TCN model, GRU model, and multi-head attention mechanism. Specifically, the RF algorithm is utilized for selecting the top features most relevant to production data, aiding in reducing computational complexity, and enhancing operational efficiency and robustness. The CEEMDAN algorithm is used to decompose production data into high-frequency noise data, low-frequency production regularity data, and residual terms reflecting trends, through appropriate parameter settings and component recombination principles. The reconstructed production data, characterized by distinct morphological features, better reflects the intrinsic relationship between production and dynamic features, thereby enhancing model learning effectiveness and predictive performance. The convolutional operations in the TCN model extract input features along the temporal dimension. The GRU model is used to handle the temporal dependencies within sequential data, enabling dynamic learning of information within the sequence and prediction. The attention mechanism dynamically adjusts the model's focus on different time points, aiding in concentrating attention on crucial time points or features to enhance predictive performance. The RF-CEEMDAN-TGMA model, constructed with a rational architecture, exhibits superior predictive performance compared to the best-performing models in the comparative analysis, resulting in a significant improvement. Specifically, compared to the best-performing reference model, it achieves a 3% reduction in RMSE, a 1.6% reduction in MAE, a 12.7% reduction in MAPE, and a 2.6% increase in R^2 in Case 1. In Case 2, there is a 7.7% decrease in RMSE, 7.7% decrease in MAE, 11.6% decrease in MAPE, and a 4.7% improvement in R^2. The proposed model contributes novel methods to the field of production forecasting.

Author Contributions: Conceptualization, Z.F.; Methodology, Z.F.; Software, Z.F.; Formal analysis, X.L.; Investigation, X.L.; Data curation, Z.W.; Writing—original draft, Z.F.; Writing—review & editing, P.L.; Supervision, P.L. and Y.W.; Funding acquisition, P.L. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the financial support of the National Natural Science Foundation of China (42202292 and 51774256) and Fundamental Research Funds for the Central Universities (No. 2-9-2023-052).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Zuoqian Wang was employed by the company PetroChina. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Gharbi, R.B.; Mansoori, G.A. An introduction to artificial intelligence applications in petroleum exploration and production. *J. Pet. Sci. Eng.* **2005**, *49*, 93–96. [CrossRef]
2. Mehrotra, R.; Gopalan, R. Factors influencing strategic decision-making processes for the oil/gas industries of UAE-A study. *Int. J. Mark. Financ. Manag.* **2017**, *5*, 62–69.
3. Doublet, L.E.; Pande, P.K.; McCollum, T.J.; Blasingame, T.A. Decline curve analysis using type curves–analysis of oil well production data using material balance time: Application to field cases. In Proceedings of the SPE International Oil Conference and Exhibition in Mexico, Veracruz, Mexico, 10–13 October 1994; p. SPE-28688.
4. Arps, J.J. Analysis of decline curves. *Trans. AIME* **1945**, *160*, 228–247. [CrossRef]
5. Geng, L.; Li, G.; Wang, M.; Li, Y.; Tian, S.; Pang, W.; Lyu, Z. A fractal production prediction model for shale gas reservoirs. *J. Nat. Gas Sci. Eng.* **2018**, *55*, 354–367. [CrossRef]
6. Chen, Y.; Ma, G.; Jin, Y.; Wang, H.; Wang, Y. Productivity evaluation of unconventional reservoir development with three-dimensional fracture networks. *Fuel* **2019**, *244*, 304–313. [CrossRef]
7. Box, G.E.P.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*, Rev. ed.; Holden-Day Series in Time Series Analysis and Digital Processing; Holden-Day: San Francisco, CA, USA, 1976; ISBN 978-0-8162-1104-3.
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
9. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
10. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014.
11. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271.
12. Wu, P.; Sun, J.; Chang, X.; Zhang, W.; Arcucci, R.; Guo, Y.; Pain, C.C. Data-driven reduced order model with temporal convolutional neural network. *Comput. Methods Appl. Mech. Eng.* **2020**, *360*, 112766. [CrossRef]
13. Zhou, C.D.; Wu, X.-L.; Cheng, J.-A. Determining reservoir properties in reservoir studies using a fuzzy neural network. In Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, TX, USA, 3–6 October 1993.
14. Pan, S.; Yang, B.; Wang, S.; Guo, Z.; Wang, L.; Liu, J.; Wu, S. Oil well production prediction based on CNN-LSTM model with self-attention mechanism. *Energy* **2023**, *284*, 128701. [CrossRef]
15. Ahmadi, M.A.; Chen, Z. Machine learning models to predict bottom hole pressure in multi-phase flow in vertical oil production wells. *Can. J. Chem. Eng.* **2019**, *97*, 2928–2940. [CrossRef]
16. Ali, M.; Zhu, P.; Jiang, R.; Huolin, M.; Ehsan, M.; Hussain, W.; Zhang, H.; Ashraf, U.; Ullaah, J. Reservoir characterization through comprehensive modeling of elastic logs prediction in heterogeneous rocks using unsupervised clustering and class-based ensemble machine learning. *Appl. Soft Comput.* **2023**, *148*, 110843. [CrossRef]
17. Liu, Y.-Y.; Ma, X.-H.; Zhang, X.-W.; Guo, W.; Kang, L.-X.; Yu, R.-Z.; Sun, Y.-P. A deep-learning-based prediction method of the estimated ultimate recovery (EUR) of shale gas wells. *Pet. Sci.* **2021**, *18*, 1450–1464. [CrossRef]
18. Maucec, M.; Garni, S. Application of automated machine learning for multi-variate prediction of well production. In Proceedings of the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain, 18–21 March 2019; p. D032S069R003.
19. Khan, M.R.; Alnuaim, S.; Tariq, Z.; Abdulraheem, A. Machine learning application for oil rate prediction in artificial gas lift wells. In Proceedings of the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain, 18–21 March 2019; p. D032S085R002.
20. Davtyan, A.; Rodin, A.; Muchnik, I.; Romashkin, A. Oil production forecast models based on sliding window regression. *J. Pet. Sci. Eng.* **2020**, *195*, 107916. [CrossRef]
21. Huang, Z.; Chen, Z. Comparison of different machine learning algorithms for predicting the SAGD production performance. *J. Pet. Sci. Eng.* **2021**, *202*, 108559. [CrossRef]
22. Al-Shabandar, R.; Jaddoa, A.; Liatsis, P.; Hussain, A.J. A deep gated recurrent neural network for petroleum production forecasting. *Mach. Learn. Appl.* **2021**, *3*, 100013. [CrossRef]
23. Ng, C.S.W.; Jahanbani Ghahfarokhi, A.; Nait Amar, M. Well production forecast in Volve field: Application of rigorous machine learning techniques and metaheuristic algorithm. *J. Pet. Sci. Eng.* **2022**, *208*, 109468. [CrossRef]
24. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man Cybern.* **1991**, *21*, 660–674. [CrossRef]
25. Cho, Y.H.; Kim, J.K.; Kim, S.H. A personalized recommender system based on web usage mining and decision tree induction. *Expert Syst. Appl.* **2002**, *23*, 329–342. [CrossRef]
26. Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J.C.; Sheridan, R.P.; Feuston, B.P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947–1958. [CrossRef]
27. Larivière, B.; Van den Poel, D. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Syst. Appl.* **2005**, *29*, 472–484. [CrossRef]

28. Prinzie, A.; Van den Poel, D. Random forests for multiclass classification: Random multinomial logit. *Expert Syst. Appl.* **2008**, *34*, 1721–1732. [CrossRef]
29. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.-C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. A* **1998**, *454*, 903–995. [CrossRef]
30. Wu, Z.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41. [CrossRef]
31. Torres, M.E.; Colominas, M.A.; Schlotthauer, G.; Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 4144–4147.
32. Zhao, W.; Gao, Y.; Ji, T.; Wan, X.; Ye, F.; Bai, G. Deep temporal convolutional networks for short-term traffic flow forecasting. *IEEE Access* **2019**, *7*, 114496–114507. [CrossRef]
33. Hewage, P.; Behera, A.; Trovati, M.; Pereira, E.; Ghahremani, M.; Palmieri, F.; Liu, Y. Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. *Soft Comput.* **2020**, *24*, 16453–16482. [CrossRef]
34. Zhu, J.; Su, L.; Li, Y. Wind power forecasting based on new hybrid model with TCN residual modification. *Energy AI* **2022**, *10*, 100199. [CrossRef]
35. Tang, Y.; Yang, K.; Zhang, S.; Zhang, Z. Wind power forecasting: A temporal domain generalization approach incorporating hybrid model and adversarial relationship-based training. *Appl. Energy* **2024**, *355*, 122266. [CrossRef]
36. Kumar, P.; Hati, A.S. Dilated convolutional neural network based model for bearing faults and broken rotor bar detection in squirrel cage induction motors. *Expert Syst. Appl.* **2022**, *191*, 116290. [CrossRef]
37. Zhu, R.; Liao, W.; Wang, Y. Short-term prediction for wind power based on temporal convolutional network. *Energy Rep.* **2020**, *6*, 424–429. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 July 2016; pp. 770–778.
39. Yin, L.; Wu, Y. Traffic flow combination prediction model based on improved VMD-GAT-GRU. *J. Electron. Meas. Instrum.* **2022**, *36*, 62–72.
40. Mercat, J.; Gilles, T.; El Zoghby, N.; Sandou, G.; Beauvois, D.; Gil, G.P. Multi-head attention for multi-modal joint vehicle motion forecasting. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 9638–9644.
41. Wang, J.; Wang, Y.; Li, Z.; Li, H.; Yang, H. A combined framework based on data preprocessing, neural networks and multi-tracker optimizer for wind speed prediction. *Sustain. Energy Technol. Assess.* **2020**, *40*, 100757. [CrossRef]
42. Yazici, I.; Beyca, O.F.; Delen, D. Deep-learning-based short-term electricity load forecasting: A real case application. *Eng. Appl. Artif. Intell.* **2022**, *109*, 104645. [CrossRef]
43. Kosana, V.; Teeparthi, K.; Madasthu, S. A novel and hybrid framework based on generative adversarial network and temporal convolutional approach for wind speed prediction. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102467. [CrossRef]
44. Li, Y.; Zuo, Z.; Pan, J. Sensor-based fall detection using a combination model of a temporal convolutional network and a gated recurrent unit. *Future Gener. Comput. Syst.* **2023**, *139*, 53–63. [CrossRef]
45. Time Series Forecasting of Oil Production in Enhanced Oil Recovery System Based on a Novel CNN-GRU Neural Network—ScienceDirect. Available online: https://www.sciencedirect.com/science/article/pii/S2949891023011156 (accessed on 6 March 2024).
46. Zha, W.; Liu, Y.; Wan, Y.; Luo, R.; Li, D.; Yang, S.; Xu, Y. Forecasting monthly gas field production based on the CNN-LSTM model. *Energy* **2022**, *260*, 124889. [CrossRef]
47. Raj, N.; Prakash, R. Assessment and prediction of significant wave height using hybrid CNN-BiLSTM deep learning model for sustainable wave energy in Australia. *Sustain. Horiz.* **2024**, *11*, 100098. [CrossRef]