Advances in Geo-Energy Research

Vol. 13, No. 2, p. 119-131, 2024

Original article

Novel Transformer-based deep neural network for the prediction of post-refracturing production from oil wells

Jing Jia[1,2,3], Diquan Li[1,3], Lichang Wang[1,3]*, Qinghu Fan[2]

[1]*School of Geosciences and Info-physics, Central South University, Changsha 410083, P. R. China*

[2]*Downhole Service Company, PetroChina Xibu Drilling Company Limited, Karamay 834000, P. R. China*

[3]*Deep Underground Resources and Energy Technology and Innovation Center, Changsha 410083, P. R. China*

Keywords:
Post-refracturing production prediction
deep neural network
Transformer architecture
Transformer-based time-series model

Cited as:
Jia, J., Li, D., Wang, L., Fan, Q. Novel Transformer-based deep neural network for the prediction of post-refracturing production from oil wells. Advances in Geo-Energy Research, 2024, 13(2): 119-131.
https://doi.org/10.46690/ager.2024.08.06

Abstract:
The accurate prediction of post-refracture production can be of great value in the selection of target wells for refracturing. Given that production from post-refracture wells yields time-series data, deep neural networks have been utilized for making these predictions. Conventional deep neural networks, including recurrent neural network and long short-term memory neural network, often fail to effectively capture long-range dependencies, which is particularly evident in tasks such as forecasting oil well production over periods extending up to 36 years. To overcome this limitation, this paper presents a novel deep neural network based on Transformer architecture, meticulously designed by fine-tuning the key components of the architecture, including its dimensions, the number of encoder layers, attention heads, and iteration cycles. This Transformer-based model is deployed on a dataset from oil wells in the Junggar Basin that spans the period of 1983 to 2020. The results demonstrate that the Transformer significantly outperforms traditional models such as recurrent neural networks and long short-term memory, underscoring its enhanced ability to manage long-term dependencies within time-series data. Moreover, the predictive accuracy of Transformer was further validated with data from six newly refractured wells in the Junggar Basin, which underscored its effectiveness over both 90 and 180 days post-refracture. The effective application of the proposed Transformer-based time-series model affirms the feasibility of capturing long-term dependencies using Transformer-based encoders, which also allows for more accurate predictions compared to conventional deep learning techniques.

1. Introduction

Refracturing involves the reapplication of fracture treatments to previously fractured wells to rejuvenate fracture conductivity that has diminished due to phenomena such as proppant embedment, fines plugging, or rock creep. Refracturing can also create new fractures, activate existing natural fractures, or connect a more extensive area of the reservoir (Lu et al., 2020; Li et al., 2022; Abdelaziz et al., 2023; Liao et al., 2024). Consequently, this technique serves as a vital enhancement to traditional reservoir stimulation techniques (He et al., 2021).

The success of refracturing critically depends on targeting the right wells, with the accurate prediction of post-fracture production being the cornerstone of this process. Despite its conceptual simplicity, however, the well selection process is still complex in practice because:

1) The geological conditions among wells are predominantly discontinuous (Davies et al., 2023; Kakemem et al., 2023; Shabani et al., 2023), which complicates the inference of geological characteristics based solely on core analysis.
2) Due to reservoir heterogeneity and well interference (Farhoodi et al., 2019; Faramarzi and Sadeghnejad, 2020; Esfandi et al., 2024; Jamshidi Gohari et al., 2024), wells take a long time to stabilize and pressure tests often fail to yield conclusive results.

Yandy Scientific Press

*Corresponding author.
E-mail address: 225001036@csu.edu.cn (J. Jia); lidiquan@csu.edu.cn (D. Li); wlccsu@csu.edu.cn (L. Wang); fanqh001@cnpc.com.cn (Q. Fan).

Received June 2, 2024; revised June 25, 2024; accepted July 16, 2024; available online July 21, 2024.

3) Utilizing the detailed parameters of rock mechanics and reservoir-specific data like porosity, permeability, thickness, and geostress distribution, a geomechanical model can be constructed using numerical simulation tools (Malki et al., 2023; Cheng et al., 2024; Wang et al., 2024). However, in older fields that have been exploited for decades, acquiring accurate reservoir characteristic parameters is challenging and costly. Meanwhile, new geological exploration or in-situ coring are impractical.
4) Particularly gravel sandstone reservoirs, which are in the experimental focus of this study, have more complex subsurface fluid dynamics due to numerous mobile grains and varied grain sizes (Yu et al., 2020). This complexity makes it nearly impossible to find highly accurate analytical solutions for predicting production from re-fractured wells in such reservoirs.

Given the above limitations, it is essential to explore alternative methods for productivity prediction without the knowledge of the reservoir characteristic parameters, such as utilizing production statistics and type-curves (Reeves et al., 2000). Production statistics evaluate individual well performance within a region by comparing each well to its neighbors to identify underperforming wells as potential refracturing candidates. This approach accounts for the full lifecycle of wells; however, it does not effectively identify high-performing wells that could benefit from refracturing. Additionally, in highly heterogeneous reservoirs, production curves cannot differentiate reservoir heterogeneity effects from completion practices. Type-curves, on the other hand, estimate permeability and skin factor using minimal data and facilitate selecting wells with favorable characteristics for refracturing. However, type-curve analysis is typically suited for homogeneous, single-layer reservoirs and introduces uncertainties in multi-layered structures, and estimating parameters like effective thickness and porosity for each well can lead to inaccuracies.

Given this context, deep learning methods emerge as a superior alternative due to their ability to discern complex nonlinear relationships and their reliance on data-driven mechanisms. Production data from refractured wells inherently constitute time-series data. Traditional time-series forecasting methodologies, such as state-space models (McCausland et al., 2011) and auto-regressive models (Kaur et al., 2023), analyze each time series independently, requiring manual trend identification. This constraint limits their utility for larger-scale forecasting in the oil and gas sector. Deep neural networks, including recurrent neural networks (RNNs) (Huang et al., 2019) and long short-term memory (LSTM) networks, retain historical information, making them suitable for time-series tasks.

RNNs, initially introduced by Rumelhart et al. (1986), have been widely adopted for time-series prediction owing to their ability to retain past information. However, RNNs frequently encounter vanishing or exploding gradients during training. LSTMs, developed by Hochreiter and Schmidhuber (1997), address these long-term dependencies on sequential data using forget, input and output gates to manage information flow. Despite this advanced design, the effective contextual capacity

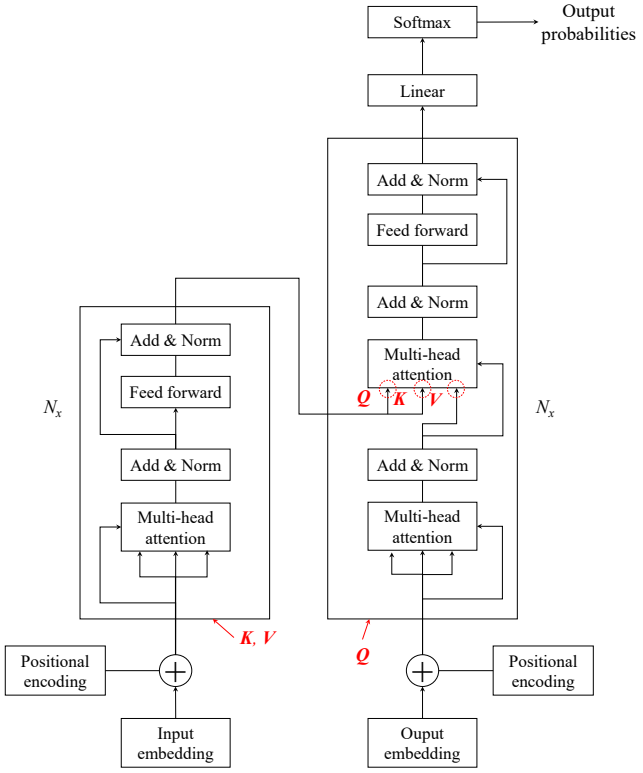

Fig. 1. Schematic of the Transformer architecture.

of LSTM-based language models remains limited to about 200 tokens, making it difficult to capture long-term dependencies effectively (Wu et al., 2021).

The need to model long-term dependencies is especially critical for large-scale datasets like those of oil well production, which exhibit both short-term and long-term repetitive patterns. As a novel deep neural network, the Transformer architecture (Vaswani et al., 2017) employs attention mechanisms to process sequential data and access any part of the historical data, making it suitable for capturing repetitive patterns with long-term dependencies. Although there has been an ongoing debate about the efficacy of Transformers in time series prediction (Zeng et al., 2023), the Transformer architecture (hereinafter referred to as Transformer) can indeed be utilized for time series forecasting (Nie et al., 2023). This paper introduces a deep neural network based on Transformer to manage long-term dependencies on time-series data to forecast production in post-refractured oil wells. The efficacy of this model is validated using real-world data from the Junggar Basin, and hyperparameter tuning is tailored to such data. This study marks the first application of a Transformer-based time-series model for predicting outputs in refractured wells, which also draws comparisons with traditional models like RNNs and LSTMs to highlight its enhanced performance.

This article is organized as follows: Section II provides a brief overview of the Transformer architecture, with a particular focus on the encoder module. Section III introduces the deep learning model proposed in this paper and outlines the construction of the research dataset. Section IV presents the fitting and forecasting results using a dataset from the W Block oil field in the Junggar Basin, covering the period of 1983

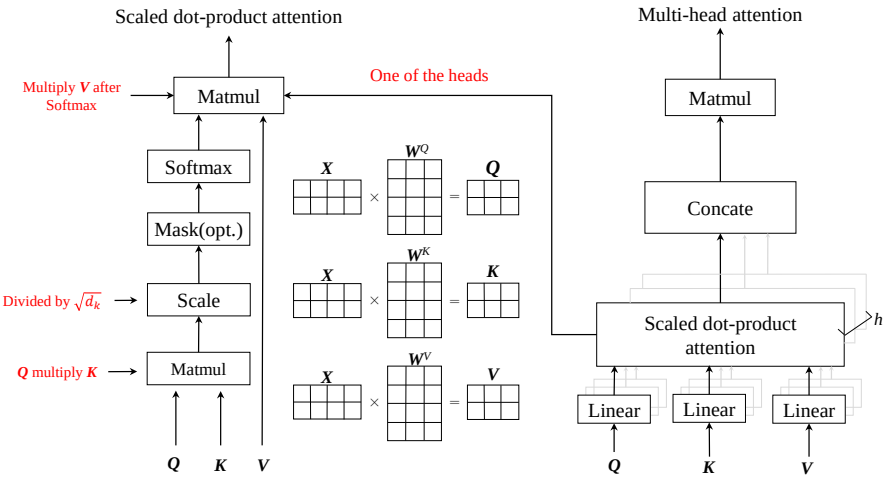

Fig. 2. Computation process of multi-head attention.

to 2020. This section also compares the fitting and forecasting accuracy of the time-series Transformer with RNNs and LSTMs. Section V describes a two-step short-term historical production fitting and prediction experiment conducted on 15 wells. Section VI summarizes the experimental results and suggests viable future research directions.

2. Transformer architecture

The Transformer is a deep neural network based on the self-attention mechanism (Vaswani et al., 2017), Its structure is shown in Fig. 1, where $\boldsymbol{K}$, $\boldsymbol{V}$ and $\boldsymbol{Q}$ represent key, value and query, respectively.

The first step in the Transformer architecture is input embedding, which transforms the input sequence into fixed-size vectors, converting each word into a vector in high-dimensional space (Vaswani et al., 2017):

$$\text{Embedding}(x) = \boldsymbol{W_e}\boldsymbol{X} + \boldsymbol{b_e} \tag{1}$$

where $\boldsymbol{W_e}$ represents the embedding matrix, $\boldsymbol{X}$ is the input matrix and $\boldsymbol{b_e}$ denotes the bias term.

Since the Transformer lacks the sequential awareness of RNNs, position encoding is used to provide information about word position in the sequence (Wang et al., 2022):

$$\text{PE}_{(\text{pos},2i)} = \sin\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}} \tag{2}$$

$$\text{PE}_{(\text{pos},2i+1)} = \cos\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}} \tag{3}$$

where $\text{PE} \in \mathbf{R}$ represents positional encoding information, pos represents the position of an element within the sequence; i indicates the index of the dimension; d_{model} denotes the dimensionality of the model.

The self-attention mechanism enables words in an input sequence to communicate and compute their relationships. For an input sequence $\boldsymbol{x} = (x_1, x_2, \cdots, x_i)$, each element x_i is transformed into $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ using three sets of weight matrices:

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}^Q \tag{4}$$

$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^K \tag{5}$$

$$\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}^V \tag{6}$$

where $\boldsymbol{Q}$ represents the query (or attention) of the current word (or position) towards other positions in the sequence; $\boldsymbol{K}$ represents each position in the sequence and is used to match with the query; $\boldsymbol{V}$ represents the content of each position in the sequence, where the corresponding value is used to compute the output once a key at a given position matches with a query. $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$ are learnable weight matrices.

Next, the dot product between the query and all keys is calculated to obtain the attention score matrix:

$$\text{Attention} = \frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}} \tag{7}$$

where d_k represents the scaling factor, which is equal to the dimension of $\boldsymbol{K}$.

The attention scores for each row are normalized using the softmax function, which ensures that each element is a positive value and that the sum of these elements equals 1:

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Softmax}(\boldsymbol{A})\boldsymbol{V} = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{8}$$

The self-attention mechanism surpasses RNNs by processing all positions in the sequence simultaneously through parallel computation, capturing long-distance dependencies and analyzing the attention weights to reveal which parts of the sequence the model prioritizes, thus enhancing interpretability.

In the attention layer of the Transformer, there are multiple attention heads. For each head i, distinct weight matrices $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$ and $\boldsymbol{W}_i^V$ are used. Each head computes the attention scores and outputs independently:

$$\boldsymbol{Q}_i = \boldsymbol{Q}\boldsymbol{W}_i^Q \tag{9}$$

$$\boldsymbol{K}_i = \boldsymbol{K}\boldsymbol{W}_i^K \tag{10}$$

$$\boldsymbol{V}_i = \boldsymbol{V}\boldsymbol{W}_i^V \tag{11}$$

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^T}{\sqrt{d_k}}\right)\boldsymbol{V}_i \tag{12}$$

Subsequently, the attention outputs from all heads are concatenated and mapped back to the original dimension via another linear transformation:

$$\text{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Contact}(head_1,\cdots,head_n)\boldsymbol{W}^O \tag{13}$$

where $head_i = \text{Attention}_i(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})$; $\boldsymbol{W}^O$ represents the output weight matrix.

The multi-head attention mechanism (Fig. 2) enhances the capacity of the model by increasing its depth or complexity. Different heads can learn various aspects of the data, such as structure and relationships between key factors. By distributing the attention across heads, the Transformer captures diverse relationships in different subspaces, leading to a more nuanced and comprehensive understanding, which is crucial for complex sequence processing tasks.

Each attention layer is followed by a Feed-Forward Network (FFN), applied identically to each position. The FNN consists of two linear transformations with a ReLU activation function between them:

$$\text{FFN}(x) = \max(0, x\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{14}$$

where $\boldsymbol{W}_1$ represents the weight matrix of the first layer, $\boldsymbol{W}_2$ denotes the weight matrix of the second layer, $\boldsymbol{b}_1$ is the bias vector of the first layer and $\boldsymbol{b}_2$ indicates the bias vector of the second layer.

Each sub-layer (attention layer and feed-forward network) includes a residual connection followed by layer normalization. This allows information to bypass the sub-layer and be directly added to its output, ensuring information flow to subsequent layers:

$$\text{Output} = \text{Sublayer}(\boldsymbol{x}) + \boldsymbol{x} \tag{15}$$

Layer normalization targets each sample independently to improve training stability and accelerate convergence. It accomplishes normalization by computing the mean and variance of all features for each sample:

$$\mu = \frac{1}{H}\sum_{i=1}^{H} x_i \tag{16}$$

$$\sigma^2 = \frac{1}{H}\sum_{i=1}^{H}(x_i - \mu)^2 \tag{17}$$

$$\text{LayerNorm}(\boldsymbol{x}) = \gamma\left(\frac{\boldsymbol{x}-\mu}{\sqrt{\sigma^2 - \varepsilon}}\right) + \beta \tag{18}$$

where $\boldsymbol{x}$ represents the input to the layer; H represents the number of features, and γ and β are learnable parameters used for rescaling and shifting, respectively. These parameters allow the network to learn to restore the original distribution that may be more useful for a specific task.

The output of each sub-layer is added to the input via residual connection, followed by layer normalization:

$$\text{Output} = \text{LayerNorm}(\text{Sublayer}(\boldsymbol{x}) + \boldsymbol{x}) \tag{19}$$

Combining residual connections and layer normalization maintains information flow, prevents the vanishing gradient problem, accelerates training, and enhances performance. These techniques enable the Transformer to effectively train deep networks and capture complex sequential relationships.

Finally, the decoder output passes through a linear layer followed by a softmax layer to produce the final output sequence:

$$\text{Output} = \text{Softmax}(\boldsymbol{W}_o\boldsymbol{x} + \boldsymbol{b}_o) \tag{20}$$

where $\boldsymbol{W}_o$ and $\boldsymbol{b}_o$ denote parameters of the linear layer.

3. Methodology

3.1 Problem definition

This paper assumes that there is a collection of N interrelated univariate time series $\{\boldsymbol{z}_{i,1:t_0}\}_{i=1}^N$, where $z_{i,t} \in \mathbf{R}$ is the value of time series i at time t and $\boldsymbol{z}_{i,1:t_0} = \lfloor \boldsymbol{z}_{i,1}, \boldsymbol{z}_{i,2}, \cdots, \boldsymbol{z}_{i,t_0} \rfloor$.

Let $\{X_{i,1:t_0+\tau}\}_{i=1}^N$ denote a set of time-dependent covariates of dimension d that are known throughout the entire time period (e.g., specific days of the year or particular hours of the day). To predict the time series $\{\boldsymbol{z}_{i,t_0+1:t_0+\tau}\}_{i=1}^N$ over τ time steps, the model described in Eq. (21) is executed to estimate the distribution of z_t, given $\boldsymbol{Y}_t$ (Eq. (22)):

$$z_t | Y_t \sim D(f(\boldsymbol{Y}_t;\theta)) \tag{21}$$

where D represents a distribution parameterized by the function f and parameter θ, and $\boldsymbol{Y}_t$ represents the known information at time t:

$$p\left(\boldsymbol{z}_{i,t_0+1:t_0+\tau} | \boldsymbol{z}_{i,i:t_0}, \boldsymbol{X}_{i,1:t_0+\tau};\Phi\right) = \prod_{t=t_0+1}^{t_0+\tau} p(\boldsymbol{z}_{i,t} | \boldsymbol{z}_{i,1:t-1}, \boldsymbol{x}_{i,1:t};\Phi) \tag{22}$$

where z_i denotes the value of time series i at time t, Φ represents the set of learnable parameters that are common across all time series within the dataset.

Next, let $y = \phi(x)$, where $\phi(\cdot)$ is an embedding function from $\mathbf{R}$ to $\mathbf{R}^d$, and let $\phi(x)$ represent the encoding of x in the d-dimensional space. The value d is referred to as the model dimension, typically set to 512 or 1,024; in this study, d is set to 64.

3.2 Deep neural network architecture

The deep neural network architecture used in this study is based on the Transformer encoder and is illustrated in Fig. 3. This architecture includes an embedding and positional encoding module for transforming raw data, a multi-head attention layer, and a point-wise feed-forward network layer. Between and after these layers, this paper applies dropout, residual connections and layer normalization, then a fully connected layer outputs the prediction results.

In Fig. 3, "×N" indicates that operations are repeated N times. In this study, N is set to 6 because the computation converges after 6 iterations.

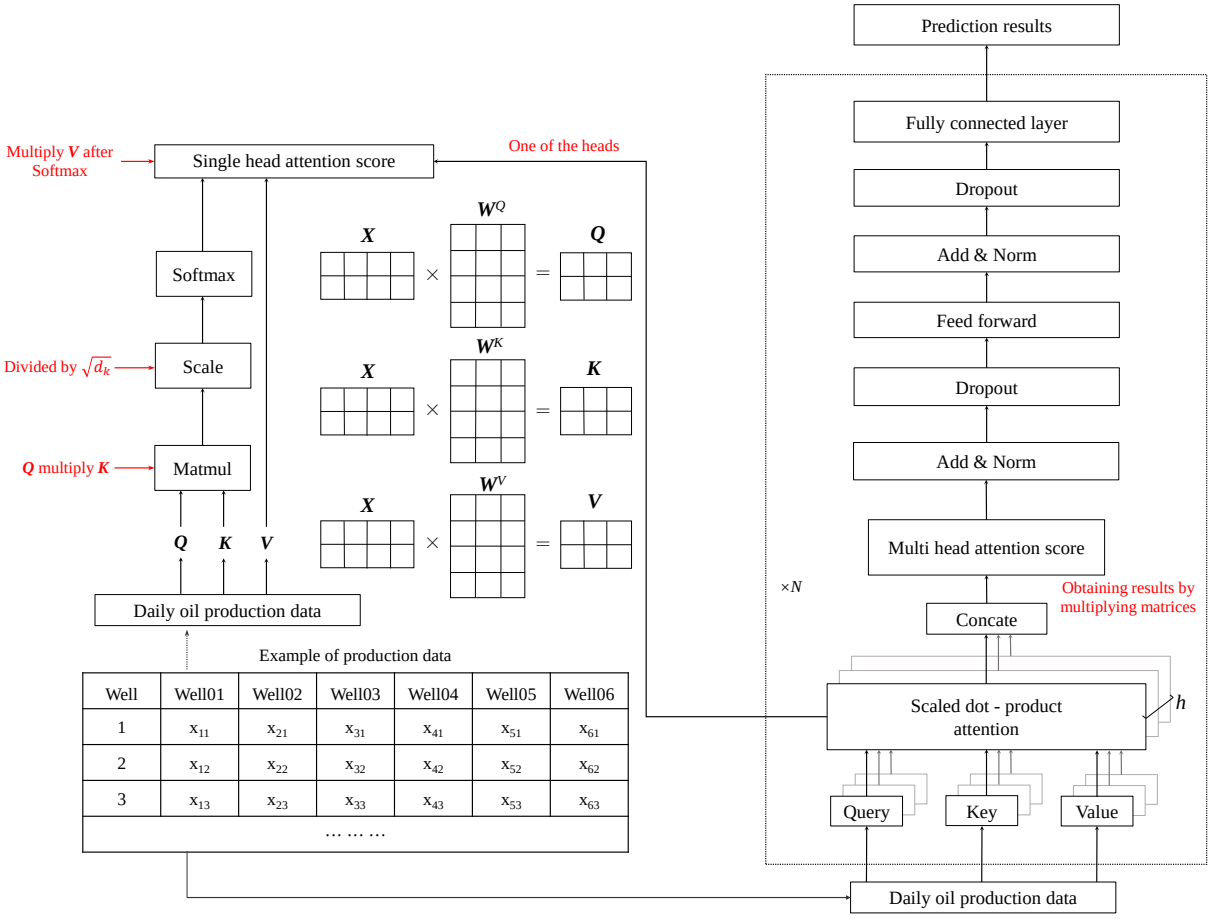

Fig. 3. Flowchart of the deep neural network of this study.

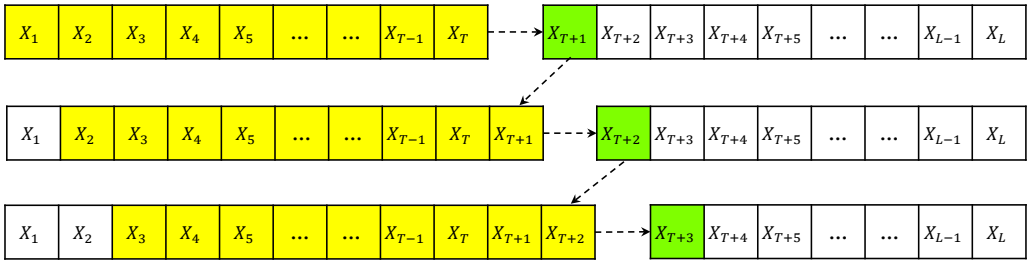

Fig. 4. Illustration of the sliding window technique.

In each prediction, the previous $T = 7$ observation points are utilized to forecast the production at the next observation point. The sliding window method is employed to construct features and labels from the observed time series, as illustrated in Fig. 4. During this process, the T data points highlighted in yellow serve as the input features for the model, while the subsequent data highlighted in green are designated as the output labels.

3.3 Dataset construction

The study area in this work is Block W, located on the downthrown side of the Ke-Wu Fault in the northwestern margin of the Junggar Basin. The reservoirs in this region are primarily distributed in the Lower Karamay Formation of the Middle Triassic.

In 2017, these reservoirs entered a secondary development stage. The entire area contains 191 oil and water wells, with 134 oil production wells, 111 of which are active, and 57 water injection wells, 29 of which are active. The daily oil production rate across the entire area is 171.4 tons per day, with a monthly oil production of 4,587 tons and a monthly water production of 8,181 cubic meters. The oil production rate is 0.49% and the liquid production rate is 1.38%. The cumulative oil production is 897,000 tons, with a recovery factor of only 8.06%. To improve crude oil recovery in this area, there is an urgent need to employ techniques such as the

re-fracturing of old wells.

Taking the production data from well J001 in Block W from July 1983 to April 2020 as an example, the time-series production data are transformed into multiple subsequences of length l, denoted as $[x]_i = (x_i, x_{i+1}, \cdots, x_{i+l-1})$. Using $[x]_i$, various types of sequence datasets can be constructed to effectively train the model.

1) A single non-overlapping sequence dataset $[x]_i$ to predict x_{l+1}, $[x]_{l+1}$ to predict $[x]_{2l+1}$, and so on.
2) A single overlapping sequence dataset $[x]_1, [x]_2, \cdot, [x]_{m-l}$, with corresponding values used to predict $x_{l+1}, x_{l+2}, \cdots, x_m$.
3) l non-overlapping sequence datasets $D_1, D_2, \cdots, D_l$, allowing the model to be trained on each dataset separately and then using an ensemble method.

From the above, method 2) is employed in this study, and the sequence dataset is divided into training sets $\{[x]_1, [x]_2, \cdots, [x]_{m_1}\}$ with prediction targets $\{x_{l+1}, x_{l+2}, , \cdots, x_{l+m_1}\}$ and testing sets $\{[x]_{m_1+1}, [x]_{m_1+2}, \cdots, [x]_{m-l},\}$ with prediction targets $\{x_{l+m_1+1}, x_{l+m_1+2}, \cdots, x_m\}$.

3.4 Evaluation metrics of the model

Evaluating a time-series model is typically based on a function comparing the predicted and actual values, much like all supervised machine learning problems. The commonly employed metrics include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), as shown:

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{23}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{24}$$

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \tag{25}$$

where y_i means accurate value, and $\hat{y}_i$ is predicted value.

4. Long-term historical production fitting and prediction experiment

Production in well J001, located in Block W, was initiated in July 1983 and operations there were concluded in April 2020, amounting to an operational duration of 36 years and 10 months. The monthly production data collected over the period covered 442 months. Using the monthly production data from well J001 from July 1983 to December 2012 as the training set and the production data from December 2012 to April 2020 as the testing set, the training set contains 354 months of production observations, and the test set contains 88 months of production observations.

4.1 Preliminary training of the model

The model parameters used in this study are shown in Table 1. The sequence length $l = 7$, which represents a step length of 7 days. The initial learning rate of the model is set to 0.01.

Eighty percent of the time-series data is allocated for training and twenty percent for testing, with the training extending over 5,000 epochs. The training loss curve before fine-tuning is shown in Fig. 5(a), and the production forecast results before fine-tuning are shown in Fig. 5(b).

The model accuracy and training loss of the base model after 500, 1,000 and 5,000 training epochs are shown in Table 2, where the number of observations for each experiment is 354.

It is observed that the training loss of the model does not converge within 1,000 epochs and there are instances where the loss increases instead of decreasing, with the loss curve showing oscillations. After 5,000 epochs of training, the loss begins to converge but at a very slow rate, indicating poor training performance. Therefore, it is necessary to optimize the model parameters to achieve optimal performance.

4.2 Fine-tuning

During training, machine learning models may experience stagnant or increasing training loss after several iterations, indicating slow or stalled convergence and potential overfitting. The key reasons for this phenomenon may include learning rate issues, excessive model complexity and improper parameter initialization.

Importantly, the learning rate determines the step size of weight updates. A high learning rate can cause overshooting, while a low one leads to slow convergence. Meanwhile, excessive model complexity can cause overfitting, where the model fits the training data perfectly but performs poorly on unseen data. Proper parameter initialization is crucial for convergence, whereas improper initialization can trap the model in local optima, hindering the gradient from reaching the global optimum.

Considering that the classic Transformer architecture is designed for natural language processing, which typically requires high model complexity (e.g., model dimension of 512, 6 encoder/decoder layers, and 8 attention heads), these settings may not suit time series processing, especially for small sample sizes. Therefore, adjusting the model parameters is necessary for optimizing a time-series Transformer model. The model dimensions are set to 64, 32, 16 and 8, the number of encoder layers are set to to 6 and 3, the attention heads are set to 8 and 4, and the epoch count is fixed at 500 for training. The loss curves of the model under different parameters are shown in Fig. 6.

The sensitivity of the model to different parameters is listed in Table 3. In Table 3, each set of parameter configurations is set to 500 epochs. In the first column of Table 3, the first number represents the dimension of model parameters, the second number represents the number of encoder layers, and the third number represents the number of attention heads. Based on Fig. 6 and Table 3, it is evident that the complexity of the model significantly affects its training performance. High complexity leads to oscillations in the loss curve and makes it difficult to converge, whereas low complexity, such as a model dimension of 8, 6 encoder layers, and 4 attention heads, results in a loss curve that does not converge at all. The

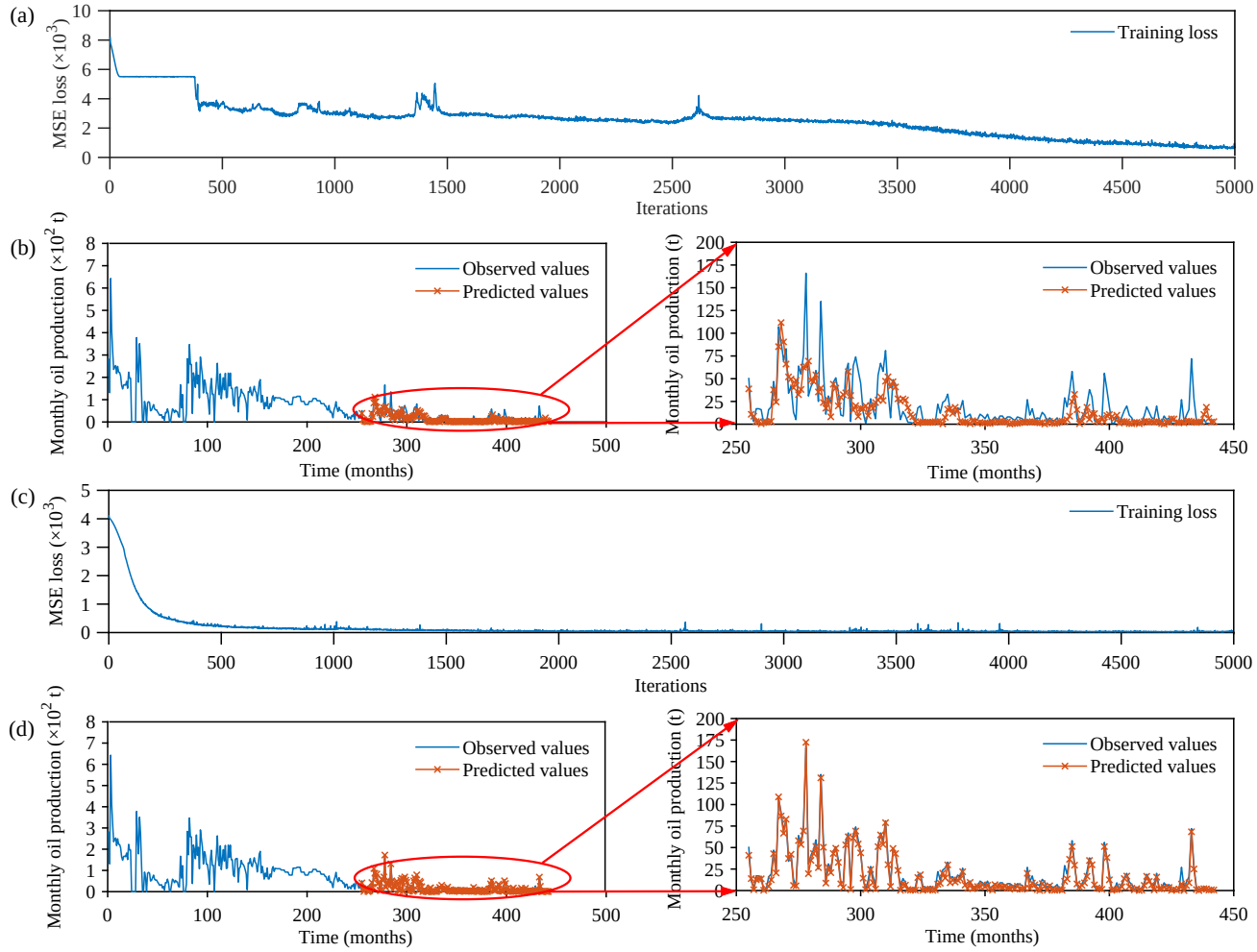

Fig. 5. Preliminary results of Well J001: (a) Training loss curve before fine-tuning, (b) production forecast results before fine-tuning, (c) training loss curve after fine-tuning and (d) production forecast results after fine-tuning.

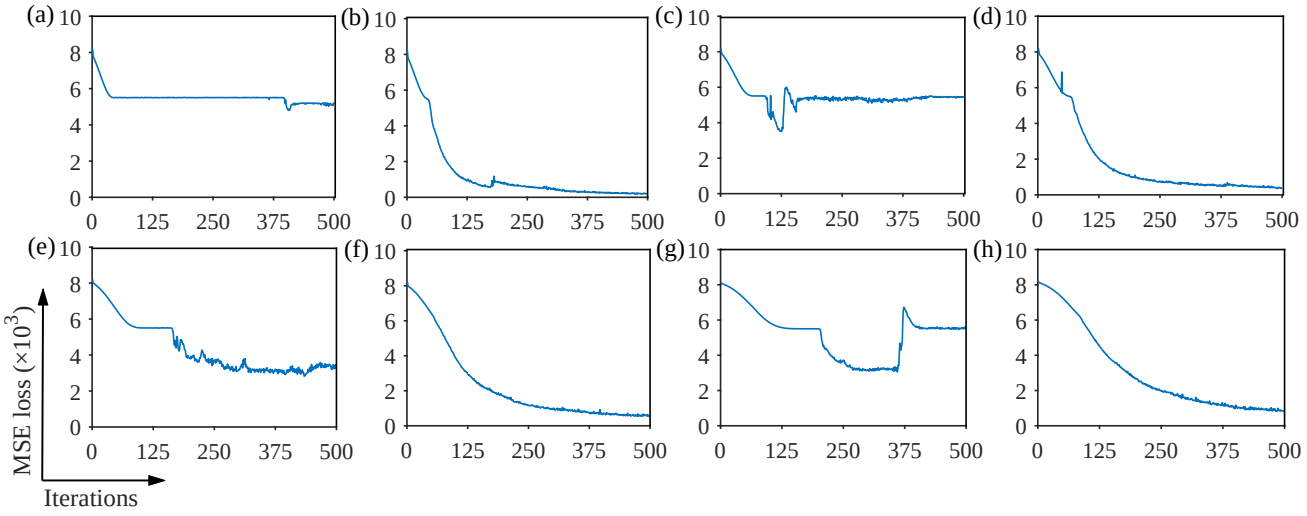

Fig. 6. Loss curves of the model under varying parameters: (a) 64-6-8, (b) 64-3-8, (c) 32-6-8, (d) 32-3-8, (e) 16-6-4, (f) 16-3-4, (g) 8-6-4 and (h) 8-3-4. The first, second, and third numbers represent the dimensions, the number of encoder layers, and the number of attention heads, respectively.

Table 1. Model parameters and specifications of this study.

Parameter	Encoder examples for natural language processing	Encoder examples for time series forecasting
Sequence length	$l \sim 1,024$	$l = 7$
Embedding dimension	$d = 512$	$d = 64$
Positional encoding method	sin and cos	sin and cos
Number of attention heads	$h = 8$	$h = 8$
Attention head size	$d_k = d/N_h = 64$	$d_k = 8$
Number of block iterations	$N = 6$	$N = 6$
Number of units in feed-forward network	$d_{ff} = 4d = 2,048$	$d = 768$

Table 2. Model accuracy comparison before and after fine-tuning.

Impact of fine-tuning	Epochs	MSE	RMSE	MAE	R^2	Loss
Before	500	5,508.3298	74.2181	55.8359	-0.0007	5,506.6142
	1,000	5508.4406	74.2188	55.8446	-0.0007	5499.4531
	5,000	724.7259	26.9207	13.5847	0.9090	735.2403
After	500	346.2329	18.6073	8.9990	0.9370	442.924
	1,000	154.1741	12.4166	7.3381	0.9719	362.6547
	5,000	27.5885	5.2524	4.2393	0.9949	83.4593

Table 3. Sensitivity analysis results for different parameter configurations.

Configuration	MSE	RMSE	MAE	R^2	Loss
64-6-8	5229.0928	72.3124	51.2264	0.0499	5307.1548
64-3-8	175.6045	13.2515	9.0640	0.9680	207.0067
32-6-8	5450.7927	73.8294	54.9118	0.0097	5441.74023
32-3-8	296.0290	17.2054	7.8183	0.9462	392.788
16-6-4	3143.3805	56.0658	34.4843	0.4289	3256.1179
16-3-4	403.7547	20.0936	8.2472	0.9266	550.7932
8-6-4	5513.3452	74.2519	56.1605	-0.0016	5497.7207
8-3-4	623.4303	24.9685	11.067	0.8867	816.0358

number of encoder layers has the most direct impact on the training results; reducing this number can most visibly improve training performance. While higher model dimensions can increase the convergence speed, this also raises the likelihood of greater training loss. When considering the experimental results comprehensively, the optimal model parameters are determined to be a model dimension of 16, 3 encoder layers, and 4 attention heads. This configuration is designated as Time-Series Transformer for Refracturing (TST-Refrac).

Based on the new optimal model parameters, the monthly oil production data for well J001 in Block W is retrained for 5,000 epochs. The loss curve of the optimal model training is shown in Fig. 5(c), and the production prediction results are illustrated in Fig. 5(d). The accuracy analysis of the fine-tuned model is presented in Table 2.

4.3 Comparison of accuracy with RNN and LSTM

The RNN and LSTM models with similar parameters are also constructed for comparative validation. The number of model layers (RNN layers, LSTM layers) is set to 3, the batch size is set to 20, and the sequence length is set to 7. After 5,000 epochs of training, the comparison results shown in Fig. 7 are obtained.

The accuracies of different models are detailed in Table 4. It can be observed that, under similar model parameters, the Transformer exhibits superior performance compared to traditional time-series forecasting network models. The main advantage of the Transformer-based model lies in its ability to capture long-range dependencies without the need for recursion or convolution, as is the case in traditional sequence

Table 4. Accuracy comparison of different predictive models.

Experiment	Model	MSE	RMSE	MAE	R^2
Preliminary experiment on well J001	Transformer	27.5885	5.2524	4.2393	0.9949
	RNN	751.5615	27.4146	12.3361	0.8634
	LSTM	436.4481	20.8913	10.4695	0.9207
Further validation on 9 wells	Transformer	0.2002	0.4475	0.2846	0.9481
	RNN	1.2219	1.1053	0.8324	0.7938
	LSTM	0.7756	0.8806	0.5356	0.8504

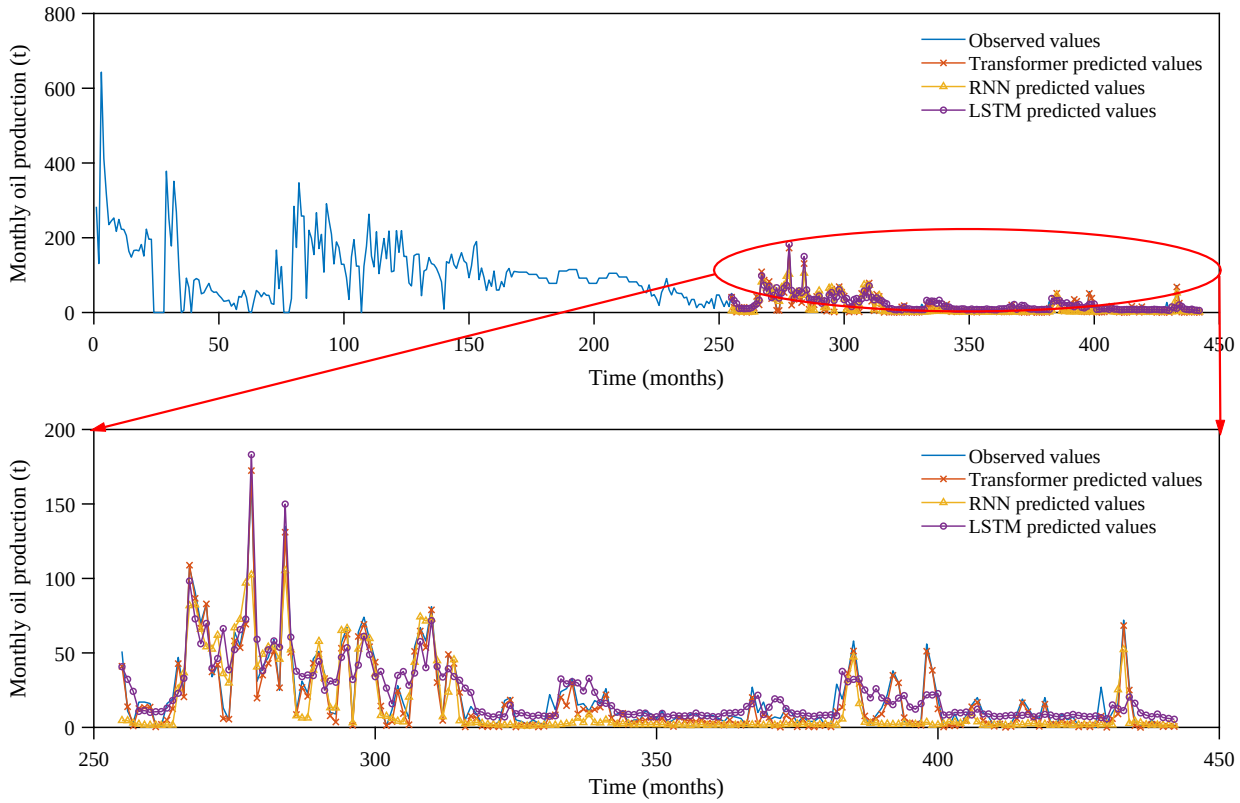

Fig. 7. Prediction comparison for Well J001 using different models.

models. This model introduces a self-attention mechanism, allowing it to simultaneously consider all positions in the input sequence rather than processing each position sequentially, resulting in higher efficiency in handling the data.

5. Short-term historical production fitting and prediction experiment

Using the production dataset from an oilfield in Block W of the Junggar Basin, nine wells are randomly selected for fitting and prediction. Taking the most recent one-year production data with daily time steps, short-term production fitting and prediction are performed. Due to varying frequencies of well interventions such as well washes, pump shutdowns and maintenance closures throughout the year, the actual volume of production data varies. The number of data points collected ranges from 200 to 300. The selected wells include J002, J003, J004, J005, J006, J007, J008, J009, and J010. The fitting and prediction results for these nine wells are presented in Fig. 8, and the error validation results for the nine wells are shown in Fig. 9.

The R^2 values representing the fitting accuracy of RNN, LSTM, and the time-series Transformer for the nine wells are summarized in Table 4. It can be seen that the average RMSE is 1.1053 for RNN, 0.8806 for LSTM, while the average RMSE for the production prediction model based on the time-series Transformer is only 0.4475. The average R^2 for 0.7938 for RNN, 0.8504 for LSTM, and it reaches 0.9481 for the time-series Transformer-based production prediction model. These data demonstrate that the proposed method has stronger generalization ability compared to RNN and LSTM.

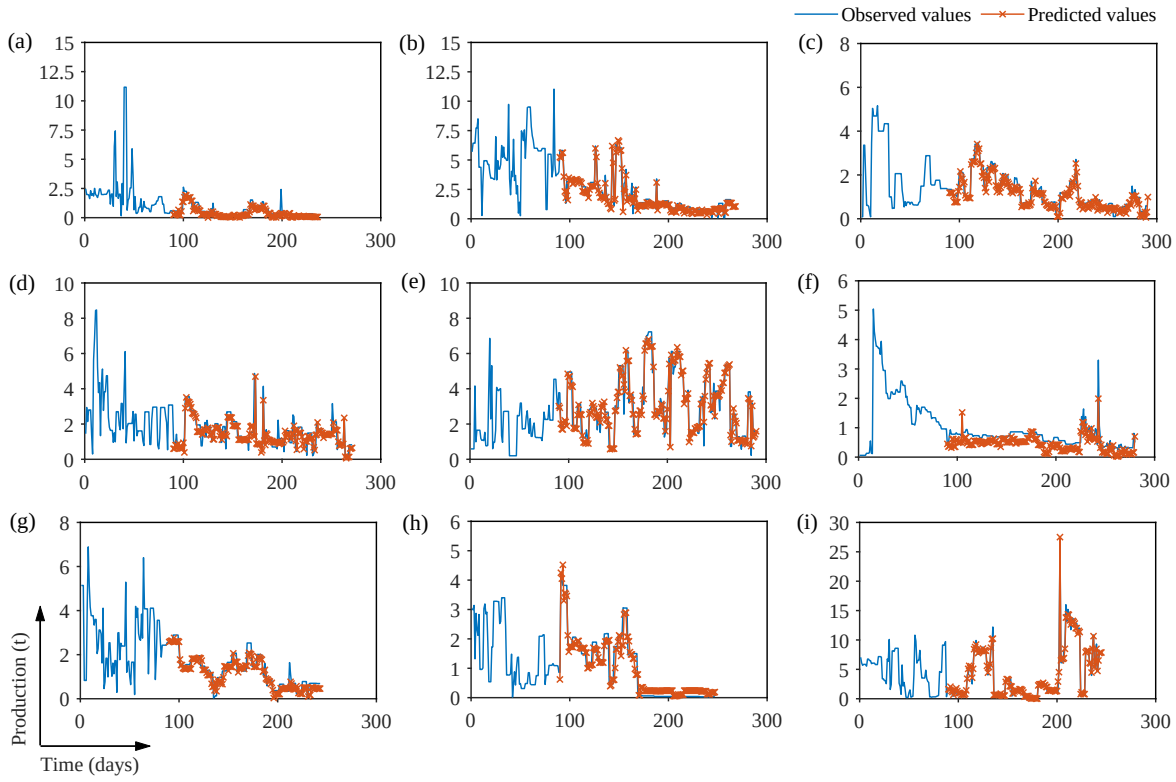

Fig. 8. Prediction outcomes for nine selected wells: (a) J002, (b) J003, (c) J004, (d) J005, (e) J006, (f) J007, (g) J008, (h) J009 and (i) J010.

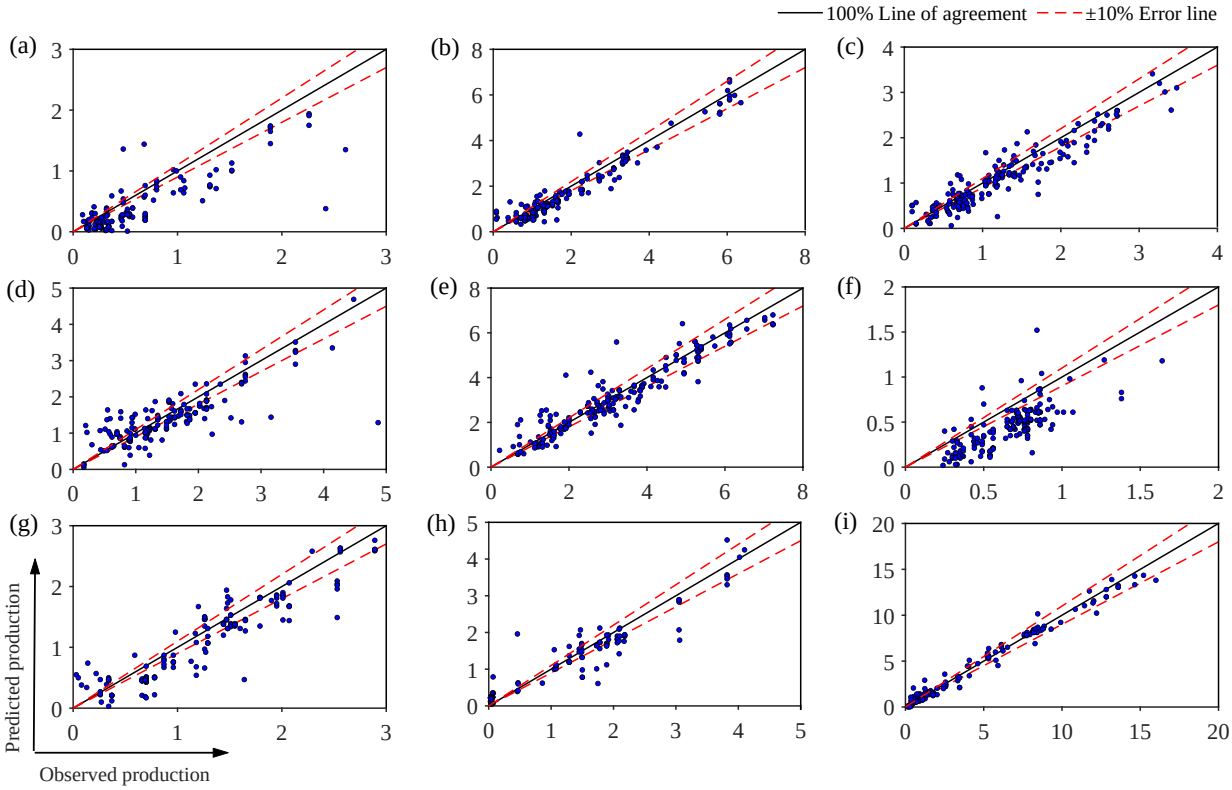

Fig. 9. Error validation across nine selected wells: (a) J002, (b) J003, (c) J004, (d) J005, (e) J006, (f) J007, (g) J008, (h) J009 and (i) J010.

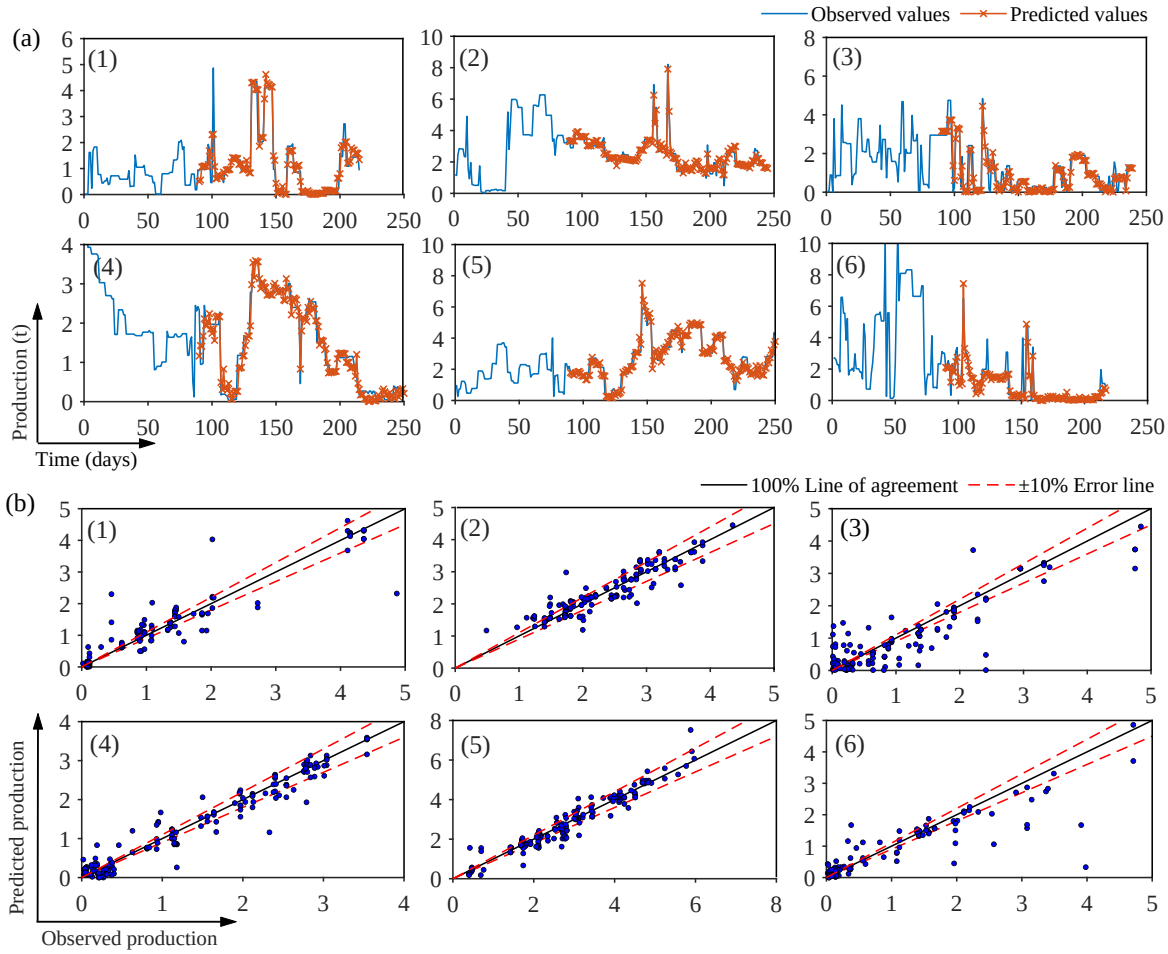

Fig. 10. Production prediction for six newly refractured wells: (a) fitting and prediction results for each well, (b) error validation for each well. In (a) and (b): (1) 5D001, (2) 5D002, (3) 5D003, (4) 5D004, (5) 5D005 and (6) 5D006.

5.1 Experiment on the measured production data of newly refractured wells

Between June and July 2023, six wells in Block W of an oilfield in the Junggar Basin underwent re-fracturing. The construction dates of these wells were not entirely consistent, and additional operational variances such as pump shutdowns and maintenance closures were present. Consequently, the volume of actual production data varies, with the number of collected data points ranging from 200 to 300. The six wells are numbered as 5D001, 5D002, 5D003, 5D004, 5D005, 5D006. The fitting and prediction results for each well are shown in Fig. 10(a), and the error validation for each well is shown in Fig. 10(b).

Subsequently, the cumulative production predicted by the time-series Transformer is calculated and compared with the actual cumulative production at the time points of 90 days and 180 days. The prediction performance of TST-Refrac for the six wells is shown in Table 5. As shown in the table, the TST-Refrac achieved a prediction accuracy of 95.61% for 90 days post-fracturing and 96.86% for 180 days post-fracturing. The errors for both key indicators are less than 5%, hence they meet the accuracy standards required for large-scale industrial application.

6. Conclusions and future directions

6.1 Conclusions

1) This paper introduces and scrutinizes the efficacy of a novel Transformer-based model, TST-Refrac, designed specifically to predict the production of refractured oil wells over an extensive timeframe of up to 36 years. Rigorous evaluations across a suite of wells in the Junggar Basin affirm the superior predictive power of TST-Refrac and its robustness in capturing long-term dependencies.
2) The coefficient of determination (R^2) of TST-Refrac is close to 1, which validates the utility of the model in improving decision-making for oil well refracturing. This is crucial for optimizing yields, particularly for gravel sandstone reservoirs.
3) Integrating the proposed model into operational frameworks can transform predictive insights into actionable strategies, enhancing the resource efficiency and economic viability. Furthermore, exploring the adaptability

Table 5. Model performance on six newly refractured wells.

No.	Production after 90 days			Production after 180 days		
	Predicted	Actual	Accuracy	Predicted	Actual	Accuracy
5D001	65.56	70.89	0.9248	191.19	194.46	0.9831
5D002	272.11	267.08	0.9811	531.68	526.4	0.9899
5D003	168.76	180.05	0.9372	254.4	276.92	0.9186
5D004	182.49	185.78	0.9822	364.58	369.03	0.9879
5D005	149.51	161.39	0.9263	386.33	400.31	0.9650
5D006	353.81	359.09	0.9852	461.42	476.96	0.9674

of the model to diverse geological datasets and real-time predictive applications could set new forecasting standards in the oil and gas industry.

6.2 Future directions

1) While the original Transformer architecture omits convolutional layers, incorporating them can be beneficial, particularly for time-series data. Many Transformers enhance the performance by adding convolutional layers or integrating them into the attention mechanisms. The next step involves experimenting with convolutional self-attention modules to process large-scale oil and gas data.
2) The original Transformer architecture can be referred to as post-layer normalization (post-LN), where layer normalization is located outside the residual block. Post-LN converges more slowly and requires a learning rate warm-up strategy. The next step could be to try using pre-layer normalization (pre-LN) Transformers to accelerate convergence without the need for warm-up. Pre-LN Transformers achieve this by controlling gradient magnitude and balancing residual dependencies.
3) To enhance the performance of time-series Transformer models on smaller datasets, better initialization techniques are needed. These techniques can regulate model updates, eliminate the need for learning rate warm-up and layer normalization, and facilitate training deeper Transformer models on small datasets.
4) Hyperparameters such as embedding dimensions and the number of heads/layers significantly impact the Transformer's performance. However, manually configuring these is time-consuming and can reduce performance. For industry-scale time-series data with high dimensions and long sequences, automated techniques like Neural Architecture Search are crucial for discovering memory-efficient and computationally effective Transformer architectures, marking an important future direction for time-series Transformers.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 52104112), the Research Foundation of the Department of Natural Resources of Hunan Province (No. 20230101DZ), the Science and Technology Program of Changsha of China (No. kh2401026), and the Science and Technology Innovation Program of Hunan Province of China (No. 2023RC3051). All research materials for this paper originate from the research project of PetroChina Xibu Drilling Co., Ltd.

Conflict of interest

The authors declare no competing interest.

References

Abdelaziz, A., Ha, J., Li, M., et al. Understanding hydraulic fracture mechanisms: From the laboratory to numerical modelling. Advances in Geo-Energy Research, 2023, 7(1): 66-68.

Cheng, L., Xie, Y., Luo, Z., et al. Numerical analysis of 3D nonplanar hydraulic fracture propagation in fractured-vuggy formations using a hydromechanical coupled XFEM approach. Computers and Geotechnics, 2024, 170: 106267.

Davies, A., Cowliff, L., Simmons, M. D. A method for fine-scale vertical heterogeneity quantification from well data and its application to siliciclastic reservoirs of the UKCS. Marine and Petroleum Geology, 2023, 149: 106077.

Esfandi, T., Sadeghnejad, S., Jafari, A. Effect of reservoir heterogeneity on well placement prediction in CO_2-EOR projects using machine learning surrogate models: Benchmarking of boosting-based algorithms. Geoenergy Science and Engineering, 2024, 233: 212564.

Faramarzi, N., Sadeghnejad, S. Fluid and rock heterogeneity assessment of gas condensate reservoirs by wavelet transform of pressure-transient responses. Journal of Natural Gas Science and Engineering, 2020, 81: 103469.

Farhoodi, S., Sadeghnejad, S., Dehaghani, A. H. S. Simultaneous effect of geological heterogeneity and condensate blockage on well test response of gas condensate reservoirs. Journal of Natural Gas Science and Engineering, 2019, 66: 192-206.

He, J., Okere, C. J., Su, G., et al. Formation damage mitigation mechanism for coalbed methane wells via refracturing

with fuzzy-ball fluid as temporary blocking agents. Journal of Natural Gas Science and Engineering, 2021, 90: 103956.

Hochreiter, S., Schmidhuber, J. Long short-term memory. Neural Computation, 1997, 9(8): 1735-1780.

Huang, Y., Shen, L., Liu, H. Grey relational analysis, principal component analysis and forecasting of carbon emissions based on long short-term memory in China. Journal of Cleaner Production, 2019, 209: 415-423.

Jamshidi Gohari, M. S., Niri, M. E., Sadeghnejad, S., et al. Synthetic graphic well log generation using an enhanced deep learning workflow: Imbalanced multiclass data, sample size, and scalability challenges. SPE Journal, 2024, 29(1): 1-20.

Kakemem, U., Ghasemi, M., Adabi, M. H., et al. Sedimentology and sequence stratigraphy of automated hydraulic flow units-The Permian Upper Dalan Formation, Persian Gulf. Marine and Petroleum Geology, 2023, 147: 105965.

Kaur, J., Parmar, K. S., Singh, S. Autoregressive models in environmental forecasting time series: A theoretical and application review. Environmental Science and Pollution Research, 2023, 30(8): 19617-19641.

Liao, Q., Wang, B., Chen, X., et al. Reservoir stimulation for unconventional oil and gas resources: Recent advances and future perspectives. Advances in Geo-Energy Research, 2024, 13(1): 7-9.

Li, Y., Zhao, Q., Lyu, Q., et al. Evaluation technology and practice of continental shale oil development in China. Petroleum Exploration and Development, 2022, 49(5): 1098-1109.

Lu, M., Su, Y., Zhan, S., et al. Modeling for reorientation and potential of enhanced oil recovery in refracturing. Advances in Geo-Energy Research, 2020, 4(1): 20-28.

Malki, M. L., Saberi, M. R., Kolawole, O., et al. Underlying mechanisms and controlling factors of carbonate reservoir characterization from rock physics perspective: A comprehensive review. Geoenergy Science and Engineering, 2023, 226: 211793.

McCausland, W. J., Miller, S., Pelletier, D. Simulation smoothing for state-space models: A computational efficiency analysis. Computational Statistics & Data Analysis, 2011, 55(1): 199-212.

Nie, Y., Nguyen, N. H., Sinthong, P., et al. A time series is worth 64 words: Longterm forecasting with Transformers. ArXiv preprint Arxiv: 2211.14730, 2023.

Reeves, S. R., Bastian, P. A., Spivey, J. P., et al. Benchmarking of restimulation candidate selection techniques in layered, tight gas sand formations using reservoir simulation. Paper SPE 63096 Presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, 1-4 October, 2000.

Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning representations by back-propagating errors. Nature, 1986, 323(6088): 533-536.

Shabani, F., Amini, A., Tavakoli, V., et al. 3D basin and petroleum system modelling of the early cretaceous play in the NW Persian Gulf. Geoenergy Science and Engineering, 2023, 226: 211768.

Vaswani, A., Shazeer, N., Parmar, N., et al. Attention is all you need. Paper 3058 Presented at the Thirty-first Annual Conference on Neural Information Processing Systems, Long Beach, California, 4-9 December, 2017.

Wang, C., Chen, Y., Zhang, S., et al. Stock market index prediction using deep Transformer model. Expert Systems with Applications, 2022, 208: 118128.

Wang, Z., Liang, W., Lian, H., et al. Numerical study of multiple hydraulic fractures propagation in poroelastic media based on energy decomposition phase field methods. Computers and Geotechnics, 2024, 170: 106259.

Wu, D., Xu, H., Jiang, Z., et al. EdgeLSTM: Towards deep and sequential edge computing for IoT applications. IEEE/ACM Transactions on Networking, 2021, 29(4): 1895-1908.

Yu, Y., Zhu, W., Li, L., et al. Multi-fracture interactions during two-phase flow of oil and water in deformable tight sandstone oil reservoirs. Journal of Rock Mechanics and Geotechnical Engineering, 2020, 12(4): 821-849.

Zeng, A., Chen, M., Zhang, L., et al. Are Transformers effective for time series forecasting? Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(9): 11121-11128.