Optimization of Well Control during Gas Flooding Using the Deep-LSTM-Based Proxy Model: A Case Study in the Baoshaceng Reservoir, Tarim, China

Qihong Feng 1, Kuankuan Wu 1, Jiyuan Zhang 1,*, Sen Wang 1, Xianmin Zhang 1, Daiyu Zhou 2 and An Zhao 2

1 School of Petroleum Engineering, China University of Petroleum (East China), Qingdao 266580, China; fqlihong@upc.edu.cn (Q.F.); z19020086@upc.edu.cn (K.W.); wangsen@upc.edu.cn (S.W.); happym@upc.edu.cn (X.Z.)
2 Tarim Oilfield Company, China National Petroleum Corporation, Korla 841000, China; zdy-tlm@petrochina.com.cn (D.Z.); zhaoan-tlm@petrochina.com.cn (A.Z.)
* Correspondence: 20180075@upc.edu.cn

Abstract: Gas flooding has proven to be a promising method of enhanced oil recovery (EOR) for mature water-flooding reservoirs. The determination of optimal well control parameters is an essential step for proper and economic development of underground hydrocarbon resources using gas injection. Generally, the optimization of well control parameters in gas flooding requires the use of compositional numerical simulation for forecasting the production dynamics, which is computationally expensive and time-consuming. This paper proposes the use of a deep long-short-term memory neural network (Deep-LSTM) as a proxy model for a compositional numerical simulator in order to accelerate the optimization speed. The Deep-LSTM model was integrated with the classical covariance matrix adaptive evolutionary (CMA-ES) algorithm to conduct well injection and production optimization in gas flooding. The proposed method was applied in the Baoshaceng reservoir of the Tarim oilfield, and shows comparable accuracy (with an error of less than 3%) but significantly improved efficiency (reduced computational duration of ~90%) against the conventional numerical simulation method.

Keywords: gas flooding; well control optimization; deep long short-term memory neural network; proxy model

1. Introduction

Gas flooding has proven to be a viable enhanced oil recovery (EOR) strategy for mature water-flooding reservoirs [1,2]. Previous laboratory and numerical simulation studies suggest that injected gas is favorable to oil swelling, oil viscosity reduction, and phase behavior, and thus tends to improve the ultimate oil recovery beyond the primary and secondary oil recovery stages. During the gas flooding process, gas channeling is a critical issue due to reservoir heterogeneity, inherent high oil/gas viscosity ratio and improper well control schedules (e.g., injection and production rates). The occurrence of gas channeling results in significant reductions in the sweep efficiency, ultimate recovery and economics of the gas injection EOR project. Therefore, to optimize the well control parameters prior to deploying a gas injection project is a vital step in order to increase gas flooding performance [3].

The optimal gas flooding control parameters can be determined using either core flooding experiments [4] or field-scale numerical simulations. Core flooding experiments are generally conducted on core samples, which are associated with significantly higher recovery factors than real reservoirs. Besides, the core flooding tests are incapable of simulating multiple injectors or multiple producers. The field-scale numerical simulation method integrates the numerical simulator with the optimization algorithm, and automatically finds the optimal well control parameters. For example, Chen et al. [5] combined the...
GA with numerical simulations to optimize water-alternating gas well control parameters. Liu et al. [6] used the genetic algorithm (GA) to optimize the injection and production of CO2 flooding. A most significant issue associated with the conventional numerical simulation method is that hundreds or thousands of numerical simulation scenarios are required in order to derive reasonable near-optimal results. The exercise of such a number of numerical simulations is computationally expensive and time-consuming and thus results in relatively low efficiency of optimization runs.

In recent years, with the development of artificial intelligence technology, many scholars have introduced machine-learning methods to predict the production dynamics of oilfields. Chakra et al. [7] used a high-order neural network (HONN) to predict the cumulative oil production of the reservoir. Wang et al. [8] established a prediction model for shale oil reservoirs using a deep learning network (DNN) to replace the numerical model process. Gu et al. [9] proposed the use of an extreme gradient boosting decision tree model to construct a surrogate model for predicting water cut based on real production data. Wang et al. [10] established a proxy model for the production performance prediction of multi-stage fracturing horizontal wells of shale oil reservoirs based on the Deep Belief Network (DBN) model, and also obtained outstanding results using the trained DBN model as a proxy to optimize the fracturing design. Gang et al. [11] employed the Extra Trees algorithm to build up the final prediction model, which is used to optimize total fluid volume and proppant quality for high volume shale gas production. Huang et al. [12] collected the actual development data in the field, and established a water-flooding reservoir development prediction model based on a long short-term memory artificial neural network (LSTM). Javadi et al. [13] combined the artificial neural network with GA to optimize well control parameters for gas flooding.

In view of the above research problems and experiences, this paper takes as its research background the optimization of injection and production parameters in the process of continuous gas flooding development in Tarim Baoshaceng reservoir. It proposes a gas flooding prediction model based on deep learning theory to replace the numerical simulation of gas injection development reservoirs, and to combine with the adaptive covariance matrix intelligent optimization algorithm (CMA-ES) to solve the built injection-production parameter optimization mathematical model, thus establishing rapid optimization method for a set of Baoshaceng reservoir gas flooding injection and production parameters. The development and adjustment of reservoir gas flooding injection–production parameters have important guiding significance.

2. Materials and Methods

2.1. Background of the Baoshaceng Reservoir

The Baoshaceng reservoir of the Tarim Oilfield is located in Shaya County, Aksu Prefecture, Xinjiang Province, China. The structure of the Baoshaceng reservoir includes nose-shaped uplifts that subside in the northwest direction and uplift in the southeast direction. The reservoir is characterized by a relatively large burial depth (5000–5023 m), high pressures (51.7–53.87 MPa) and high temperatures (111.5–115.9 °C). The average porosity and permeability are 13.67% and 98.68 × 10^{-3} \mu m^2, respectively. The porosity, permeability and pressure distributions are shown in Figure 1. The underground crude oil has an average density of 0.79 g/cm^3, and average viscosity of 4.71 mPa-s.

The development of the Baoshaceng reservoir started with primary depletion in September 1998, which was followed by subsequent water flooding initiated in October 2002. Currently, the target reservoir has 12 water injectors, 4 gas injectors and 21 producers. The comprehensive water cut, formation pressure and oil recovery are 77.6%, 48.5 MPa and 33.24%, respectively. The injected gas type is the natural gas associated with a minimum miscible pressure of 45.2 MPa with the crude oil. Figure 1 shows the field map distribution of the Baoshaceng reservoir model, and Figure 2 the historical production dynamics of the Baoshaceng reservoir. The accuracy of history matching in the whole area is over 90%, indicating that this model can be used to predict reservoir development performance.
of the Baoshaceng reservoir model, and Figure 2 the historical production dynamics of the Baoshaceng reservoir. The accuracy of history matching in the whole area is over 90%, indicating that this model can be used to predict reservoir development performance.

Figure 1. Basic properties of Baoshaceng reservoir: (a) Permeability distribution, (b) Porosity distribution, (c) Pressure distribution, (d) Oil saturation distribution.

Figure 2. Production history of the Baoshaceng reservoir. (a) Daily oil production history fitting. (b) Water cut history fitting.

2.2. Sample Construction

To construct the data samples used for training the Deep-LSTM model, numerical simulations were conducted for 800 scenarios of injection and production parameter combinations. In this paper, the simulation scenarios were generated using the Latin hypercube sampling method [14]. Data was divided into the training set and test set, according to the distribution 8 to 2. The composite simulator ECLIPSE E300 was used for simulation. The
number of effective grids in the H reservoir was 104,073, and the average running time was 30 min. Each scenario considered the well control parameters including (i) the initiating time of gas injection, gas injection rate and total gas injection volume of each gas injector, (ii) the water injection rate of each water injector and (iii) the liquid production rate of each producer. The well control parameter boundaries are given in Table 1. The upper and lower limits of the gas injection rate depend on the power of the on-site compressor, the water injection rate limit depends on the power of the on-site pump, and the upper and lower limits of the liquid production rate refer to the actual on-site data. All the data are based on the actual situation on site. However, in order to achieve a better optimization effect, the upper limit of the optimization variable was appropriately enlarged. The economic parameters involved in calculating the objective function are shown in Table 2. Gas injection costs, water injection costs, and water treatment costs are derived from the actual field data from the oil field. We referred to the literature for the value of the annual discount rate, and we assumed a reasonable value of 10 \(10,15–17\). Numerical simulations were run on a production duration of 10 years and the net present value (NPV) was calculated based on the simulated injection and production data, which can be written as:

\[
NPV = \sum_{i=1}^{Nt} \frac{1}{(1 + b)^k} \left[ \sum_{i=1}^{Np} r_{gp} q_{gp}^{ij}\k + \sum_{i=1}^{Np} r_{op} q_{op}^{ij}\k - \sum_{i=1}^{Nw} c_{wi} q_{wi}^{ij}\k - \sum_{i=1}^{Nc} c_{gi} q_{gi}^{ij}\k - \sum_{i=1}^{Np} c_{wp} q_{wp}^{ij}\k \right]
\]  

(1)

where \(q_{gp}^{ij}\k\) is the cumulative oil production of the \(i\)th well in the \(k\)th year; \(q_{gp}^{ij}\k\) is the cumulative gas production of the \(i\)th well in the \(k\)th year; \(q_{wp}^{ij}\k\) is the cumulative water production of the \(i\)th well in the \(k\)th year; \(q_{wp}^{ij}\k\) is the accumulated water injection of the \(i\)th well in the \(k\)th year; \(r_{op}\) is oil price; \(r_{gp}\) is gas price; \(c_{wp}\) is water treatment price; \(c_{wi}\) is water injection cost; \(c_{gi}\) is gas injection cost; \(N_t\) is evaluation time, years; \(b\) is the annual discount rate; \(N_{IW}\) is number of water injection wells; \(N_{IG}\) is number of gas injection wells; \(N_P\) is number of producer wells.

Table 1. Latin hypercube sampling upper and lower limits.

| Parameters                    | Lower Sampling Limit | Upper Sampling Limit |
|-------------------------------|----------------------|----------------------|
| Gas injection timing (days)   | 0                    | 120                  |
| Gas injection rate (m³/d)     | 0                    | 120,000              |
| Total gas injection (10⁸ m³)  | 0                    | 4                    |
| Water injection rate (m³/d)   | 0                    | 150                  |
| Liquid production rate (m³/d) | 0                    | 150                  |

Table 2. Economic parameter value.

| Parameters                        | Value  |
|-----------------------------------|--------|
| Gas injection cost (RMB/m³)       | 1.3    |
| Water injection cost (RMB/m³)     | 30     |
| Water treatment cost (RMB/m³)     | 10     |
| Oil sale price (RMB/m³)           | 2800   |
| Gas sale price (RMB/m³)           | 0.9    |
| Annual discount rate              | 10     |

Since the orders of magnitude of the well control parameters are different, each well control parameter was normalized so that all the parameters are comparable at an equivalent order of magnitude. The normalization was achieved using the following formula [18]:

\[
X_i = \frac{x_i - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}
\]  

(2)
where \( X_i \) is the \( i \)th normalized data corresponding to the \( i \)th original data \( x_i \); \( x_{\text{min}} \) and \( x_{\text{max}} \) are the upper and lower sampling boundaries, respectively.

It is worth noting that all calculations in this work were run on the following computer system configuration, as shown in Table 3:

**Table 3.** Computer system configuration and simulation environment.

| Parameter               | Type               |
|-------------------------|--------------------|
| RAM                     | 16G                |
| GPU                     | NVIDIA GTX 1060    |
| CPU                     | Intel i7 10875H@2.30 GHz |
| Computer system         | Windows10          |
| Deep learning environment | TensorFlow         |

### 2.3. Development of the Deep-LSTM Proxy Model

#### 2.3.1. Basics of the Deep-LSTM

Long Short-Term Memory (LSTM) networks are popular neural models that are used for learning and predicting long-term dependencies. An LSTM network typically consists of an input layer, a hidden layer and an output layer. The hidden layer, which is also referred to as the memory unit, controls the long-distance temporal information transmission. The memory unit consists of three gate calculation structures, namely the "input gate", "forget gate" and "output gate" (Figure 3). The "input gate" is used for inputting and filtering new information. The "forget gate" is the most important controlling mechanism of an LSTM, discarding the useless information and retaining the useful information. The "output gate" enables the memory cell to output only information relevant to the current time step. These three gate structures perform operations such as matrix multiplication and nonlinear summation in the memory unit so that the memory information will not disappear in continuous iterations (i.e., the issue of gradient disappearance can be effectively resolved) [19–21].

![LSTM neural network memory unit structure](image)

**Figure 3.** LSTM neural network memory unit structure.

Mathematically, the information input at the \( t \)-th time step passes through the input gate, and the input value in the memory unit and the candidate value of the hidden layer \((i_t \text{ and } C_{th})\) are computed respectively as:

\[
i_t = \sigma(W_i \times [H_{t-1}, X_t] + h_t) \\
C_{th} = \tanh(W_c \times [H_{t-1}, X_t] + h_c)
\]
Then the input information passes through the forget gate, and the forgetting layer of the memory unit at the $t$ time step is assigned as $f_t$:

$$f_t = \sigma(W_f \times [H_{t-1}, X_t] + b_f)$$  \hspace{1cm} (5)

At this time, the information of the hidden layer in the memory unit is updated to $C_t$:

$$C_t = f_t \times C_{t-1} + i_t \times C_{th}$$  \hspace{1cm} (6)

Finally, the information passes through the output gate. The value $O_t$ of the output layer in the memory unit at the $t$-th time step and the final output value $h_t$ are:

$$O_t = \sigma(W_o \times [H_{t-1}, X_t] + b_o)$$  \hspace{1cm} (7)

$$h_t = O_t \times \tan(h(C_t))$$  \hspace{1cm} (8)

Previous studies have shown that Deep-LSTM has stronger stability and higher accuracy than conventional LSTM [22,23]. Deep-LSTM can be considered as multiple LSTM network neurons stacked together to form a training unit, with the time series information output of the previous layer used as the time series information input of the next layer (Figure 4). The depth of the structure can build a hierarchical feature representation of the input data, thereby improving the efficiency of model training. Since the time series of Deep-LSTM is very complex, the number of training iterations is large, and problems of overfitting and data redundancy are prone to occur. Therefore, the dropout function is added to the network framework, and a certain amount of loss can proportionally occur during each iteration of training. The output of the LSTM structure at the current moment is passed into a fully connected dense layer to obtain the final output.

![Figure 4. Deep-LSTM neural network memory architecture.](image)

2.3.2. Construction of the Deep-LSTM Proxy Model

In this study, a 3-layer Deep-LSTM was constructed to develop the proxy model for predicting the gas flooding performance. The “return_sequences” line indicates whether to return the value of the hidden state of the last time step or the value of the hidden state of each time step in the output sequence. When “return_sequences” is “True”, it returns the
value of the hidden state of each time step, when “return_sequences” is “False”, it returns the value of the hidden state of the last time step. In this study, the “return_sequences” of the first two layers of LSTM were set to “True”, and the “return_sequences” of the last layer of LSTM was set to “False”. The learning rate determines whether and when the objective function converges to a local minimum. Smaller learning rates tend to increase the risk of overfitting, while larger ones may slow convergence. Batch size affects the training accuracy and speed of the model, and its value may affect computer performance. If there are too many neurons, the calculation speed is slow and overfitting is easy; if there are too few neurons, the training accuracy will be affected. The loss value of the Dropout layer can effectively prevent overfitting, and if it is too large, it will lead to poor training accuracy. The Epoch value represents the number of times of training. If it is too small, it will underfit, and if it is too large, it will overfit. These hyperparameters affect each other, so there is an optimal combination for the model training. In the LSTM layer, the activation function of the neuron was set to be the linear rectification function (Relu). The optimizer used Adam \[24,25\]. The mean square error (MSE) and coefficient of determination (R^2) were used to evaluate the accuracy of the proxy model.

The hyperparameters of the Deep-LSTM (e.g., the number of hidden layers, learning rate and training times, etc) may affect training efficiency and accuracy \[26\]. Therefore, this study used the Bayesian optimization to determine the optimal values of the hyperparameters. The details of the hyperparameter optimization process are given in references and therefore are not repeated in this paper \[27–30\]. The hyperparameters that were optimized and their ranges are given in Table 4.

### Table 4. Bayesian Optimization Hyperparameters.

| Parameters                     | Range               | Optimal Parameters |
|--------------------------------|---------------------|--------------------|
| Learning rate                  | 0.0001–0.1          | 0.001              |
| Batch size                     | 20–100              | 32                 |
| The number of neurons in the first layer | 10–200              | 24                 |
| The number of neurons in the second layer | 10–200              | 48                 |
| The number of neurons in the third layer | 10–200              | 60                 |
| Loss function value(Dropout)   | 0.05–0.5            | 0.2                |
| Epoch                          | 10–200              | 100                |

2.4. Optimization of the Well Control Parameters for Gas Flooding

The goal of well control parameter optimization is to find optimal values for injection and production parameters that maximize oil production or economics. For the continuous gas flooding problem, the optimization variables for each gas injection well include gas injection start time, gas injection rate, and total gas injection volume; the optimization variable for each water injection well is the water injection rate; the optimization variable for production wells is all production wells under certain constraints and liquid production rate under certain conditions (e.g., bottom hole pressure, BHP). For the production of constant fluid volume in every well, the optimization variables are:

\[
x_g = [I_{t1}, \cdots, I_{ti}, \cdots, I_{tn}, I_{v1g}, \cdots, I_{vigi}, \cdots, I_{vgn}, I_{cg1}, \cdots, I_{cgni}, \cdots, I_{cgn}]
\]

\[
x_w = [I_{vwi}, \cdots, I_{vw1}, \cdots, I_{vwzi}]
\]

\[
x_p = [P_{L1}, \cdots, P_{Li}, \cdots, P_{Lm}]
\]

where \(x_g\) is the optimization variable for each gas injection well; \(x_w\) is the optimization variable for each water injection well; \(x_p\) is the optimization variable for each production well; \(I_{ti}\) is gas injection timing of the \(i\)th gas injection well; \(I_{vigi}\) is gas injection rate of the \(i\)th gas injection well; \(I_{cgni}\) is the Cumulative gas injection of the \(i\)th gas injection well; \(I_{vwzi}\) is the water injection rate of the \(i\)th water injection well; \(P_{Li}\) is the fluid production of the \(i\)th
production well; \( z \) is the number of water injection wells; \( n \) is the number of gas injection wells; \( m \) number of producer wells.

In the actual production processes of different oil reservoirs, the injection and production systems have certain constraints. For example, the daily gas injection rate of the gas injection well depends on the actual gas injection capacity of the on-site compressor. In order to ensure that the numerical simulation results are within a reasonable range, it was necessary to add corresponding constraints to the optimization variables in the optimization mathematical model. The constraints for the well control optimization of gas flooding are summarized in Table 5.

### Table 5. Mathematical Model Constraints.

| Constraint Type                          | Constraint Variables          | Constraint Expression |
|------------------------------------------|-------------------------------|------------------------|
| Inequality Boundary Condition Constraints| Gas injection timing  \( l_{gi} \)  |  \( l_{gim} \leq l_{gi} \leq l_{gimax}, i = 1, 2, \ldots n \) |
|                                           | Gas injection rate \( v_{gi} \) |  \( v_{gimin} \leq v_{gi} \leq v_{gimax}, i = 1, 2, \ldots n \) |
|                                           | Total gas injection \( P_{li} \) |  \( P_{limin} \leq P_{li} \leq P_{limax}, i = 1, 2, \ldots n \) |
|                                           | Liquid production rate \( P_{Li} \) |  \( P_{Limin} \leq P_{Li} \leq P_{Limax}, i = 1, 2, \ldots n \) |
|                                           | Water injection speed \( v_{wi} \) |  \( v_{wimin} \leq v_{wi} \leq v_{wimax}, i = 1, 2, \ldots n \) |

On the basis of the construction of the Deep-LSTM proxy model, we further combined the covariance matrix adaptive evolutionary algorithm (CMA-ES) with the Deep-LSTM model to optimize the injection and production parameters of hydrocarbon gas flooding in the Baoshaceng reservoir. The CMA-ES is a stochastic non-gradient algorithm that has strong global optimization ability [31,32]. The specific optimization process is shown in Figure 5 and is presented briefly as follows:

**Step 1**: Initialize the optimization algorithm and the optimization variables;

**Step 2**: Train the Deep-LSTM model and then predict NPV;

**Step 3**: Automatic optimization of CMA-ES algorithm;

**Step 4**: Determine whether the termination condition is reached;

**Step 5**: If the prediction accuracy of the Deep-LSTM model meets the requirements, output optimal well control parameters and NPV; otherwise, repeat from Step 2 to Step 4 (e.g., update the training set, retrain the model, restart the optimization algorithm).

![Figure 5. CMA-ES optimization process.](image-url)
3. Results and Discussion

3.1. Development of the Proxy Model

Figure 6 shows the model accuracy’s dependence (in terms of the determination coefficient $R^2$) on the number of samples at varying numbers of layers. It can be seen that for a given number of layers, the model accuracy shows a gradually increasing trend with the increase in the number of samples. However, when the number of samples is larger than approximately 800, increasing the number of samples results in a very limited elevation in the model accuracy. It can be also seen from Figure 6 that the number of layers exerts a profound effect on the model accuracy. For example, for a given number of samples of 800, the $R^2$ values of the single-, two-, three- and four-layer Deep-LSTM network are 0.967, 0.975, 0.989 and 0.986, respectively. By comparison, it can be found that the three-layer Deep-LSTM model has higher accuracy and stability than models with single, two and four layers. In this regard, the three-layer Deep-LSTM architecture was selected in this study.

![Figure 6. Comparison of (a) Effect of the number of samples and layers on the model accuracy and (b) Effect of the number of samples and layers on the model MSE.](image)

With the aim of optimizing the Deep-LSTM architecture, the values of the hyperparameters were further optimized. The results showed that the optimal time step, single batch output, number of neurons in a single layer, dropout loss value and training round were 3, 32, 24, 48, 60, 0.2 and 100, respectively. Figure 7 depicts the scatter plot of predicted versus true NPVs with the optimized Deep-LSTM. It can be seen that the data points are located around the 45-degree line. The corresponding $R^2$ and MSE are 0.9954 and 0.018, respectively, indicating the relatively high accuracy of the model.

In order to verify the reliability and superiority of the Deep-LSTM constructed in this study, the Deep-LSTM model was compared with the classic machine-learning algorithms XGBoost and FCNN. These two methods are the most commonly used machine-learning algorithms and have been widely utilized in the energy field [9,33]. Before comparison, the hyperparameters of XGBoost and FCNN were also optimized by the Bayesian optimization method. The training effects of different algorithm test sets are shown in Figure 8. Obviously, the training effect of the FCNN and XGBoost test set was not as good as Deep-LSTM. As can be seen from Table 6, comparing the accuracy and error of model predictions for different methods, the Deep-LSTM established in this paper is better than XGBoost and FCNN in terms of accuracy, robustness and generalization ability.
3.2. Optimization of the Well Control Parameters

We used the CMA-ES optimization algorithm, set the maximum number of iterations to 600 and the number of populations to 20, calculated NPV based on the constructed Deep-LSTM model, and optimized the injection and production parameters for the H reservoir. As shown in Figure 9, due to the randomness of the algorithm, the solid lines denote the average results for ten independent runs.

Table 6. Comparison of NPV Training Accuracy of Different Algorithms.

| Method       | MSE     | \(R^2\)  |
|--------------|---------|----------|
| Deep-LSTM    | 0.018   | 0.9954   |
| FCNN         | 0.059   | 0.9815   |
| XGBoost      | 0.081   | 0.9731   |

Figure 7. Comparison of (a) The cross plot of predicted versus true NPV based on the Deep-LSTM model (test set training effect) and (b) Loss function value changes during training.

Figure 8. Test set training effect comparison (a) XGBoost model (b) FCNN model.
Figure 8. Test set training effect comparison (a) XGBoost model (b) FCNN model.

Table 6. Comparison of NPV Training Accuracy of Different Algorithms.

| Method          | MSE    | $R^2$   |
|-----------------|--------|---------|
| Deep-LSTM       | 0.018  | 0.9954  |
| FCNN            | 0.059  | 0.9815  |
| XGBoost         | 0.081  | 0.9731  |

3.2. Optimization of the Well Control Parameters

We use the CMA-ES optimization algorithm, set the maximum number of iterations to 600 and the number of populations to 20, calculate NPV based on the constructed Deep-LSTM model, and optimize the injection and production parameters for the H reservoir.

As shown in Figure 9, Due to the randomness of the algorithm, the solid lines denote the average results for ten independent runs.

Figure 9. Comparing the process of optimizing NPV using different methods.

Table 7 is the comparison of the optimization results of four gas injection wells, Table 8 is the comparison of the optimization results of 12 water injection wells, and Table 9 is the comparison of the optimization results of 21 production wells.

Table 7. Comparison of optimization results of 4 gas injection wells.

| Optimization Parameters | Optimization Variables | Numerical Simulation | Deep-LSTM Model |
|-------------------------|------------------------|----------------------|-----------------|
| gas injection start     | $l_1$                  | 3365                 | 3482            |
| time (days)             | $l_2$                  | 1                    | 1               |
|                         | $l_3$                  | 3495                 | 3330            |
|                         | $l_4$                  | 1111                 | 954             |
|                         | $l_{tg1}$              | 51,681.95            | 63,797.47       |
| gas injection rate      | $l_{tg2}$              | 114,304.78           | 115,261.37      |
| (m$^3$/d)               | $l_{tg3}$              | 95,569.58            | 88,207.42       |
|                         | $l_{tg4}$              | 87,742.34            | 89,058.18       |
|                         | $l_{tg5}$              | 368,426.22           | 236,419.96      |
| gas injection volume    | $l_{cg1}$              | 193,615,238.47       | 144,773,030.62  |
| (m$^3$)                | $l_{cg2}$              | 193,615,238.47       | 144,773,030.62  |
|                         | $l_{cg3}$              | 65,753,357.66        | 42,146,090.84   |
|                         | $l_{cg4}$              | 55,483,180.47        | 57,906,751.61   |

Table 8. Comparison of optimization results of 12 water injection wells.

| Water Injection Rate (m$^3$/d) | $I_{wv1}$ | $I_{wv2}$ | $I_{wv3}$ | $I_{wv4}$ | $I_{wv5}$ | $I_{wv6}$ |
|-------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| Numerical simulation           | 150.00    | 138.61    | 150.00    | 148.44    | 145.96    | 124.09    |
| Deep-LSTM model                | 150.00    | 134.56    | 148.20    | 149.18    | 150.00    | 125.27    |

| Water Injection Rate (m$^3$/d) | $I_{wv7}$ | $I_{wv8}$ | $I_{wv9}$ | $I_{wv10}$ | $I_{wv11}$ | $I_{wv12}$ |
|-------------------------------|-----------|-----------|-----------|------------|------------|------------|
| Numerical simulation           | 100.54    | 65.26     | 56.00     | 98.66      | 140.36     | 142.38     |
| Deep-LSTM model                | 136.31    | 77.60     | 65.32     | 91.25      | 146.33     | 1355.60    |
Table 9. Comparison of optimization results of 21 production wells.

| Production Well Rate (m³/d) | P₁  | P₂  | P₃  | P₄  | P₅  | P₆  | P₇  | P₈  | P₉  | P₁₀ | P₁₁ |
|-----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Numerical simulation        | 149 | 122.3 | 142 | 150.1 | 83.5 | 150.1 | 150.1 | 124.1 | 150.1 | 149.7 | 85.5 |
| Deep-LSTM model             | 142.5 | 139 | 130.1 | 139 | 77 | 150.1 | 143.7 | 119 | 143.3 | 135 | 88.2 |

| Production Well Rate (m³/d) | P₁₂ | P₁₃ | P₁₄ | P₁₅ | P₁₆ | P₁₇ | P₁₈ | P₁₉ | P₂₀ | P₂₁ |
|-----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Numerical simulation        | 108.1 | 98 | 91.6 | 132.8 | 73.9 | 142.6 | 69.1 | 1 | 57.9 | 120.2 |
| Deep-LSTM model             | 100.1 | 109.7 | 82.5 | 142.0 | 66.1 | 145 | 70 | 1 | 59 | 128.8 |

Figure 10. Comparison of (a) NPV and (b) cumulative oil production using different methods.

Figure 11. Comparison of oil saturation after well control optimization using (a) conventional numerical simulation and (b) the Deep-LSTM proxy model.

Table 10 compares optimization results based on the Deep-LSTM and conventional numerical simulation. It can be seen from the table that compared with the optimization results based on numerical simulation, the prediction error of NPV is 2.1%, and the prediction error of cumulative oil production is 3.5%. However, the optimization efficiency of
the Deep-LSTM model was improved by more than 90%. In summary, the optimization method based on the Deep-LSTM model established in this paper significantly improved optimization efficiency while ensuring optimization accuracy, which proves the effectiveness and superiority of the optimization method based on the Deep-LSTM model. At the same time, the Deep-LSTM model we established in the optimization process has the characteristics of repeatable optimization.

### Table 10. Comparison of optimization results of different methods.

| Methods        | NPV (Billion) | Cumulative Oil ($10^5 \text{ m}^3$) | CPU Time          |
|----------------|---------------|--------------------------------------|-------------------|
| Numerical simulation | 20.65         | 12.25                                | 345.38 h          |
| Deep-LSTM      | 20.24         | 11.92                                | 5 h 42 min        |
| Error (%)      | 2.1           | 2.7                                  | /                 |

#### 4. Conclusions

This study used Deep-LSTM to construct a proxy model, which was then used for optimizing well control parameters of the gas flooding EOR process. It is shown that the prediction accuracy of the Deep-LSTM model (in terms of $R^2$) can reach 0.995. The constructed Deep-LSTM model was then integrated with the classical CMA-ES algorithm to establish a fast optimization method of the well control parameters for the gas flooding process. The method was applied to the Baoshaceng reservoir in Tarim Oilfield, showing comparable accuracy (with an error of less than 3%) but significantly improved efficiency (reduced computational duration of ~90%) against the conventional numerical simulation method. At the same time, the Deep-LSTM optimization model established in this paper has the advantage of repeated use, and can repeatedly and quickly calculate the optimization of injection and production parameters for the Baoshaceng reservoir, thereby providing effective technical support for on-site development decisions at the Baoshaceng reservoir.

**Author Contributions:** Conceptualization, Q.F. and J.Z.; methodology, S.W.; software, K.W.; validation, X.Z. and K.W.; formal analysis, K.W.; investigation, D.Z. and A.Z.; resources, A.Z.; data curation, S.W.; writing—original draft preparation, K.W.; writing—review and editing, J.Z. and X.Z.; visualization, J.Z.; supervision, Q.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was financially supported by the Major Science and Technology Project of CNPC (Grant No. ZD2019-183-007).

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

**References**

1. Sheng, J.J. Enhanced oil recovery in shale reservoirs by gas injection. *J. Nat. Gas Sci. Eng.* 2015, 22, 252–259. [CrossRef]
2. Qi, Z.; Liu, T.; Xi, C.; Zhang, Y.; Shen, D.; Mu, H.; Dong, H.; Zheng, A.; Yu, K.; Li, X. Status Quo of a CO$_2$-Assisted Steam-Flooding Pilot Test in China. *Geofluids* 2021, 2021, 9968497. [CrossRef]
3. Li, X.; Wang, S.; Yuan, B.; Chen, S. Optimal Design and Uncertainty Assessment of CO$_2$ WAG Operations: A Field Case Study. In Proceedings of the SPE Improved Oil Recovery Conference, Tulsa, OK, USA, 14–18 April 2018.
4. Ghedan, S.G. Global laboratory experience of CO$_2$-EOR flooding. In Proceedings of the SPE/EAGE Reservoir Characterization and Simulation Conference, Abu Dhabi, United Arab Emirates, 19–21 October 2009.
5. Chen, B.; Reynolds, A.C. Ensemble-based optimization of the water-alternating-gas-injection process. *SPE J.* 2016, 21, 0786–0798. [CrossRef]
6. Liu, S.; Agarwal, R.; Sun, B.; Wang, B.; Li, H.; Xu, J.; Fu, G. Numerical simulation and optimization of injection rates and wells placement for carbon dioxide enhanced gas recovery using a genetic algorithm. *J. Clean. Prod.* 2021, 280, 124512. [CrossRef]
7. Chakra, N.C.; Song, K.-Y.; Gupta, M.M.; Saraf, D.N. An innovative neural forecast of cumulative oil production from a petroleum reservoir employing higher-order neural networks (HONNs). *J. Pet. Sci. Eng.* 2013, 106, 18–33. [CrossRef]

8. Wang, S.; Chen, Z.; Chen, S. Applicability of deep neural networks on production forecasting in Bakken shale reservoirs. *J. Pet. Sci. Eng.* 2019, 179, 112–125. [CrossRef]

9. Gu, J.; Liu, W.; Zhang, K.; Zhai, L.; Zhang, Y.; Chen, F. Reservoir production optimization based on surrogate model and differential evolution algorithm. *J. Pet. Sci. Eng.* 2021, 205, 108879. [CrossRef]

10. Wang, S.; Qin, C.; Feng, Q.; Javadpour, F.; Rui, Z. A framework for predicting the production performance of unconventional resources using deep learning. *Appl. Energy* 2021, 295, 117016. [CrossRef]

11. Hui, G.; Chen, S.; He, Y.; Wang, H.; Gu, F. Machine learning-based production forecast for shale gas in unconventional reservoirs via integration of geological and operational factors. *J. Nat. Gas Sci. Eng.* 2021, 94, 104045. [CrossRef]

12. Huang, R.; Wei, C.; Wang, B.; Yang, J.; Xu, X.; Wu, S.; Huang, S. Well performance prediction based on Long Short-Term Memory (LSTM) neural network. *J. Pet. Sci. Eng.* 2022, 208, 109686. [CrossRef]

13. Javadi, A.; Moslemizadeh, A.; Moluki, V.S.; Fathianpour, N.; Mohammadzadeh, O.; Zendehboudi, S. A Combination of Artificial Neural Network and Genetic Algorithm to Optimize Gas Injection: A Case Study for EOR Applications. *J. Mol. Liq.* 2021, 339, 116654. [CrossRef]

14. Shields, M.D.; Zhang, J. The generalization of Latin hypercube sampling. *Reliab. Eng. Syst. Saf.* 2016, 148, 96–108. [CrossRef]

15. Zhang, K.; Wang, Z.; Chen, G.; Zhang, L.; Yang, Y.; Yao, C.; Wang, J.; Yao, J. Training effective deep reinforcement learning agents for real-time life-cycle production optimization. *J. Pet. Sci. Eng.* 2022, 208, 109766. [CrossRef]

16. Kim, J.; Lee, K.; Choe, J. Efficient and robust optimization for well patterns using a PSO algorithm with a CNN-based proxy model. *J. Pet. Sci. Eng.* 2021, 207, 109088. [CrossRef]

17. Zaac, D.; Kang, Z.B.; Jian, H.; Dwac, D.; Ypac, D. Accelerating reservoir production optimization by combining reservoir engineering method with particle swarm optimization algorithm. *J. Pet. Sci. Eng.* 2022, 208, 109692. [CrossRef]

18. Bhanja, S.; Das, A. Impact of Data Normalization on Deep Neural Network for Time Series Forecasting. *arXiv* 2018, 1812.05519.

19. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Hasan, M.; Van Essen, B.C.; Awval, A.A.; Asari, V.K. A state-of-the-art survey on deep learning theory and architectures. *Electronics* 2019, 8, 292. [CrossRef]

20. Sherwani, F.; Ibrahim, B.; Asad, M.M. Hybridized classification algorithms for data classification applications: A review. *Egypt. Inform. J.* 2021, 22, 185–192. [CrossRef]

21. DiPietro, R.; Hager, G.D. Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 503–519.

22. Sagheer, A.; Kotb, M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* 2019, 323, 203–213. [CrossRef]

23. Bao, R.; He, Z.; Zhang, Z. Application of lightning spatio-temporal localization method based on deep LSTM and interpolation. *Measurement* 2022, 189, 110549. [CrossRef]

24. Jais, I.K.M.; Ismail, A.R.; Nisa, S.Q. Adam optimization algorithm for wide and deep neural network. *Knowl. Eng. Data Sci.* 2019, 2, 41–46. [CrossRef]

25. Nguyen, L.C.; Nguyen-Xuan, H. Deep learning for computational structural optimization. *ISA Trans.* 2020, 103, 177–191. [CrossRef] [PubMed]

26. Liu, X.; Wu, J.; Chen, S. A Context-Based Meta-Reinforcement Learning Approach to Efficient Hyperparameter Optimization. *Neurocomputing* 2022, 478, 89–103. [CrossRef]

27. Wu, J.; Chen, X.-Y.; Zhang, H.; Xiong, L.-D.; Lei, H.; Deng, S.-H. Hyperparameter optimization for machine learning models based on Bayesian optimization. *J. Electron. Sci. Technol.* 2019, 17, 26–40.

28. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Processing Syst.* 2012, 25. Available online: https://arxiv.org/abs/1206.2944 (accessed on 1 February 2022).

29. Eggersperger, K.; Feurer, M.; Hutter, F.; Bergstra, J.; Snoek, J.; Hoos, H.; Leyton-Brown, K. Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In *Proceedings of the NIPS workshop on Bayesian Optimization in Theory and Practice, Lake Tahoe, NV*, USA, 10 December 2013.

30. Victoria, A.H.; Maragatham, G. Automatic tuning of hyperparameters using Bayesian optimization. *Evol. Syst.* 2021, 12, 217–223. [CrossRef]

31. Rios, L.M.; Sahinidis, N.V. Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Glob. Optim.* 2013, 56, 1247–1293. [CrossRef]

32. Hansen, N.; Müller, S.D.; Koumoutsakos, P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* 2003, 11, 1–18. [CrossRef]

33. KUANG, L.; He, L.; Yili, R.; Kai, L.; Mingyu, S.; Jian, S.; Xin, L. Application and development trend of artificial intelligence in petroleum exploration and development. *Pet. Explor. Dev.* 2021, 48, 1–14. [CrossRef]