

Article

# Application of Gated Recurrent Unit (GRU) Neural Network for Smart Batch Production Prediction

**Xuechen Li** , **Xinfang Ma** , **Fengchao Xiao**, **Fei Wang**  and **Shicheng Zhang**

State Key Laboratory of Petroleum Resources and Prospecting & MOE Key Laboratory of Petroleum Engineering, China University of Petroleum (Beijing), Beijing 102249, China; 2019310154@student.cup.edu.cn (X.L.);

2020310175@student.cup.edu.cn (F.X.); wangfei@cup.edu.cn (F.W.); zhangsc@cup.edu.cn (S.Z.)

\* Correspondence: maxinfang@cup.edu.cn; Tel.: +86-133-6650-5131

Received: 14 October 2020; Accepted: 19 November 2020; Published: 22 November 2020



**Abstract:** Production prediction plays an important role in decision making, development planning, and economic evaluation during the exploration and development period. However, applying traditional methods for production forecasting of newly developed wells in the conglomerate reservoir is restricted by limited historical data, complex fracture propagation, and frequent operational changes. This study proposed a Gated Recurrent Unit (GRU) neural network-based model to achieve batch production forecasting in M conglomerate reservoir of China, which tackles the limitations of traditional decline curve analysis and conventional time-series prediction methods. The model is trained by four features of production rate, tubing pressure (TP), choke size (CS), and shut-in period (SI) from 70 multistage hydraulic fractured horizontal wells. Firstly, a comprehensive data preprocessing is implemented, including excluding unfit wells, data screening, feature selection, partitioning data set, z-score normalization, and format conversion. Then, the four-feature model is compared with the model considering production only, and it is found that with frequent oilfield operations changes, the four-feature model could accurately capture the complex variance pattern of production rate. Further, Random Forest (RF) is employed to optimize the prediction results of GRU. For a fair evaluation, the performance of the proposed model is compared with that of simple Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) neural network. The results show that the proposed approach outperforms the others in prediction accuracy and generalization ability. It is worth mentioning that under the guidance of continuous learning, the GRU model can be updated as soon as more wells become available.

**Keywords:** GRU; time series; production forecasting; RF; deep learning

## 1. Introduction

Production prediction plays an important role in decision making, development planning, and economic evaluation during the exploration and development period. Especially in unconventional reservoirs, accurate production estimation helps evaluate oil and gas potential and, at the same time, guides fracturing operation reasonably [1–3]. There are several commonly used methods for future production prediction.

Reservoir numerical simulation is a common method for production rate forecasting because of its reliable physical background [4,5]. However, it is time-consuming and demands many reservoir features, such as rock mechanical parameters and fluid property parameters [6]. Inaccurate or missing parameters will lead to unbelievable production predictions. Especially in conglomerate reservoirs, it is difficult to build a robust numerical model due to the effect of gravel as well as the complicated fracture propagation mechanism.

Decline curve analysis (DCA) is also frequently used in the oilfield, which is based on empirical production decline [7]. It is straightforward and fast compared with reservoir numerical simulation. Additionally, it does not need many static and dynamic data. However, it has many limitations to constrain its application [8,9]. First, traditional DCA can be utilized when the well has entered the production decline stage, which means the target well has to produce for at least several months. However, for wells that are newly put into operation, it is difficult to apply this method for rate prediction. Further, traditional DCA cannot take operational changes into account, which have a significant influence on production. Last but not least, as many production curves fluctuate heavily, selecting the start of fitting data manually is essential. The analyst's subjectivity will largely influence the analysis results [10].

Machine learning, also known as data mining, is one approach that could extract implicit knowledge from massive data. With the development of computer technology and intelligent oilfield construction, it has been widely used for production analysis in the oil and gas industry [11–13]. Conventional time-series prediction methods including Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), and Autoregressive Integrated Moving Average with Exogenous Variables (ARIMAX) are successfully applied for production prediction. They perform well in the production prediction of unconventional gas and oil reservoirs with regular changing patterns [14–16]. As a branch of machine learning, deep learning has shown excellent performance in various applications, such as refracturing candidate selection [17], pressure prediction [18–20], history matching [21,22], and production forecasting. Wang et al. established a deep fully connected neural network model to predict cumulative oil production of 6 and 18 months [23]. A total of 18 static parameters relating to well information, formation property, fractures, and fracturing property are used as input features, and values of cumulative production at fixed time points are used as output features. Luo et al. developed a deep learning model for the first-year production Barrel of Oil Equivalent (BOE) in the Bakken field and explored the influence of geology and completion parameters on BOE [24]. He established a data-driven model for Estimated Ultimate Recovery in Marcellus shale, considering reservoir characteristics and completion strategies [25]. However, these studies focus on the relationship between geological and fracturing features and cumulative production at a fixed time and do not involve time-series rate prediction.

Recurrent Neural Network (RNN) is an important part of deep learning, which focuses on time-series problems because of its advantages over sequence dependence. However, due to gradient exploding and vanishing, there is a length limitation when applying the RNN algorithm [26]. Subsequently, variants of RNN, such as Long Short-Term Memory (LSTM) [27] and Gated Recurrent Unit (GRU) [28] neural network, are proposed to deal with long sequence prediction problems. Various researches have been done in applying RNN and LSTM in tackling time-series problems in recent years. Based on the initial 3-month data, Zhan et al. established a three-layer LSTM model for sequent 700-day oil prediction with tubing pressure as latency information [29]. Lee et al. successfully utilized LSTM to predict monthly shale-gas production, considering the shut-in period [10]. Sun et al. compared LSTM with DCA in wells that have been producing for over 800 days and found that LSTM showed a better trend and less error calculation with tubing pressure as an external feature [30]. Song et al. combined LSTM with Particle Swarm Optimization to predict the daily oil rate of the volcanic reservoir [31]. The results show that LSTM could capture the changing patterns of production and outperforms DCA, RNN, and traditional time series forecasting methods. However, most wells used in the above models have a long production time with few field operations. Batch production forecasting for newly developed wells with multiple external features is rarely involved.

In this paper, GRU models with multiple external features and various prediction approaches were established for oil rate prediction of the M conglomerate oilfield in China for the first time. Our objective is to establish a production prediction model suitable for the whole well block, which enables batch prediction given limited production data. In the following section, a comprehensive data preprocessing was introduced in detail firstly, which is of vital importance to successful prediction. Then, the relevant

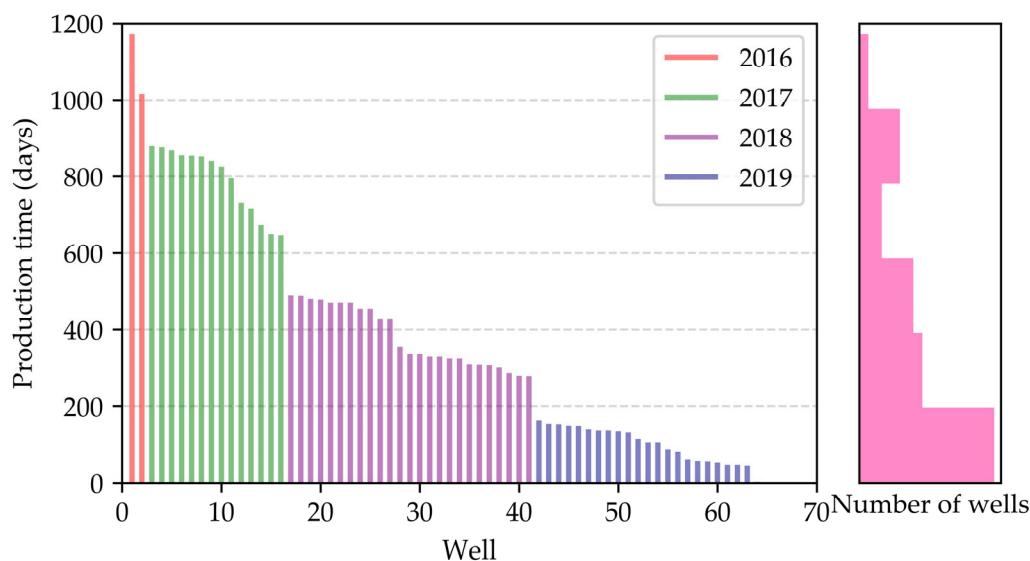
mechanism, the architecture, and three prediction approaches of the proposed model were explained. After model training, model validation, application, and correction with Random Forest (RF) were employed for future production rate forecasting successively. Additionally, continuous learning was introduced into prediction with available data increases. Finally, the main conclusions and prospects were presented.

## 2. Data Preprocessing

### 2.1. Data Set

In this study, we focus on the oil wells of X well block in M oilfield, China. M oilfield belonging to a conglomerate reservoir with a dozen well blocks. Among them, X is one of the earliest regions of scale development, owning more than 70 multistage hydraulically fractured horizontal wells since 2016. The size of gravel in X well block generally ranges from 2 mm to 8 mm with a maximum value of 16 mm, and the contents of gravel and sand are around 51% and 44%, respectively. According to the conventional logging test and core analyzing experiment, the permeability varies from 0.05 to 94.8 mD with an average value of 2.3 mD, and the porosity ranges from 4.3% to 15.3% with an average of 9.23%.

Figure 1 shows the production time of all the available wells. From 2016 to 2019, the number of fractured wells increases year by year with 2 wells in 2016, 14 wells in 2017, 26 wells in 2018, and 28 wells in 2019 (from January to September). More wells are waiting for drilling and fracturing shortly. As of the end of September 2019, only 13 wells had produced for more than 2 years, and 32.4% of the wells are less than 200 days. Short production time will increase the difficulty and uncertainty in applying conventional DCA. Raw data of the above wells are collected from the oilfield database, including production date, SI, CS, TP, oil production rate ( $q_o$ ), production method (PM), strokes, and so on (see Supplementary Materials). The ending of the production date is 2 September 2019.



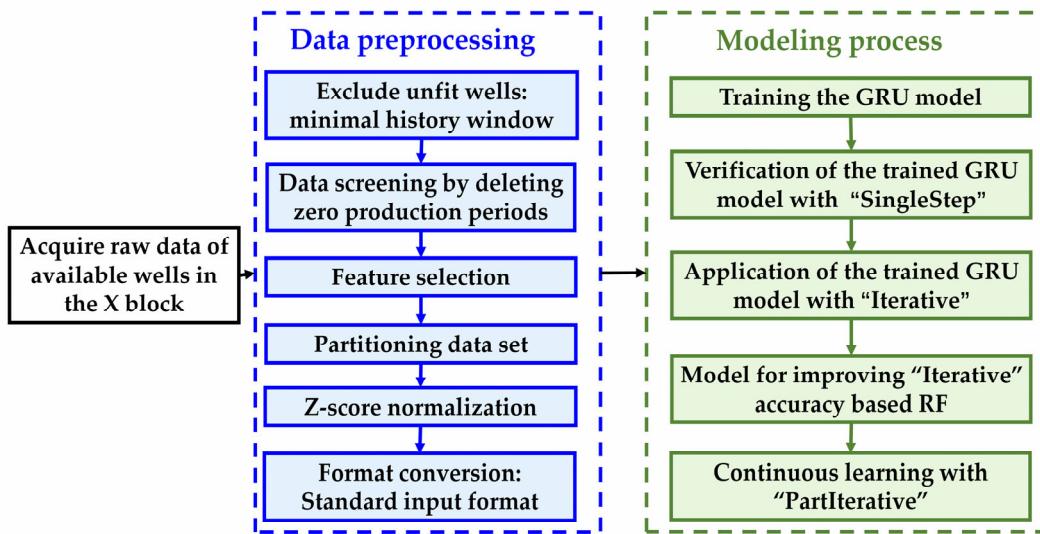
**Figure 1.** Production time of wells in X well block.

### 2.2. Workflow

The quality of the data set is of utmost importance because the core of data mining is data itself. Figure 2 depicts the workflow of data preparation (the blue box) and the modeling process (the green box). There are six steps in the preprocessing process.

Firstly, unfit wells, whose production time is less than the history window (HW), are excluded from the data set. HW refers to the number of days used to predict the production rate at the next day. If HW is too big, the number of samples will be decreased, resulting in underfitting problems. On the contrary, if HW is too small, some key information hidden in the long time series might be ignored,

which leads to poor prediction performance. As there are numerous newly developed wells, 14 days is selected as HW in this research, and then 5 wells are excluded from the data set.



**Figure 2.** Workflow of data preprocessing and model analysis.

Secondly, data screening is implemented on each well by deleting periods before oil production appears. These zero values do not contribute to models but will increase computing time. Remarkably, as an external feature, zero production periods caused by shut-in operation should not be excluded.

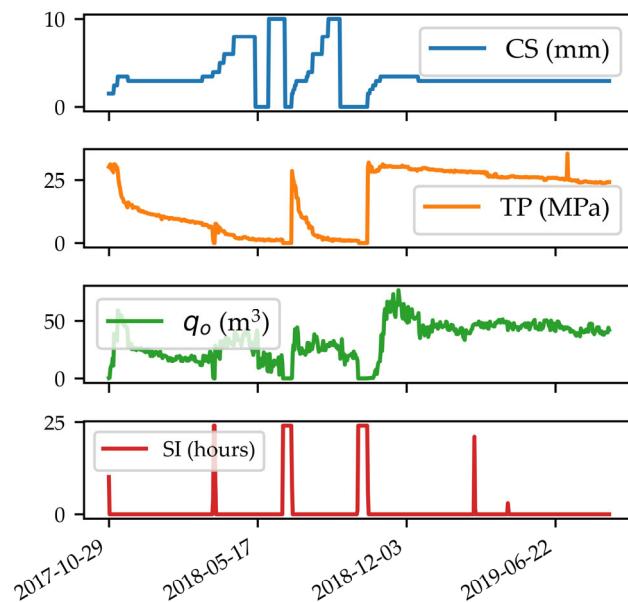
The third step of data preprocessing is to select appropriate input features. The available features obtained from the oilfield database include SI, CS, TP, PM, strokes, and  $q_o$ . By analyzing the historical data, PM and strokes are removed from available features because all the wells are gusher wells with unchangeable values. Figure 3 displays a typical production history with frequent operations. It is obvious that changing choke size and shut-in operation relate to sharp fluctuation in production rate and tubing pressure. It is essential to consider SI and CS as input features. The analysis results are consistent with the Pearson correlation coefficients heatmap, as shown in Figure 4. The coefficients between every two variables range from  $-1$  to  $1$ . The number  $1$  and  $-1$  represent a completely positive and negative relation, respectively. The number  $0$  means that there is no linear correlation between the two variables. It can be found that  $q_o$  has a positive relation with CS and TP, whereas it has a negative relation with SI. The largest absolute Pearson correlation coefficient is  $0.47$ , which is between  $q_o$  and TP, suggesting that TP has a larger impact on  $q_o$ .

After the above steps, a total of 65 wells are selected to implement data set partitioning. For different purposes, the data set is generally divided into the training set, validation set, and test set. The training data allow the model to learn the relationship between the input sequence and the output for good prediction. The validation set is utilized for neural network architecture optimization and hyperparameter optimization (i.e., learning rate, drop out ratio, regularization parameter of L2 norm). The test set is used for corroborating the prediction ability and robustness of the trained models. In this research, 52 wells are randomly selected as the training set, which accounts for 80% of the data set. The validation set and the test set contain 7 wells and 6 wells, respectively.

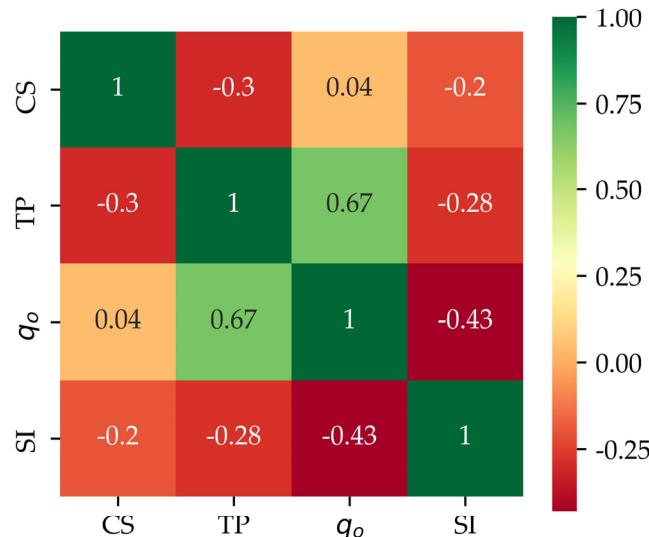
In the fifth step, data standardization is implemented based on z-score normalization to improve the efficiency of the trained model, as defined in Equation (1). One key point to note is that the validation set and test set should be normalized based on the mean and standard deviation of the training set only. The statistical property of input features is summarized in Table 1.

$$Fea_{nor} = \frac{Fea - Mean_{tra}}{Std_{tra}} \quad (1)$$

where  $Mean_{tra}$  and  $Std_{tra}$  are the mean and standard deviation of the training set,  $Fea$  denotes input features, and  $Fea_{nor}$  represents input features after z-score normalization.



**Figure 3.** Typical production history in X well block.



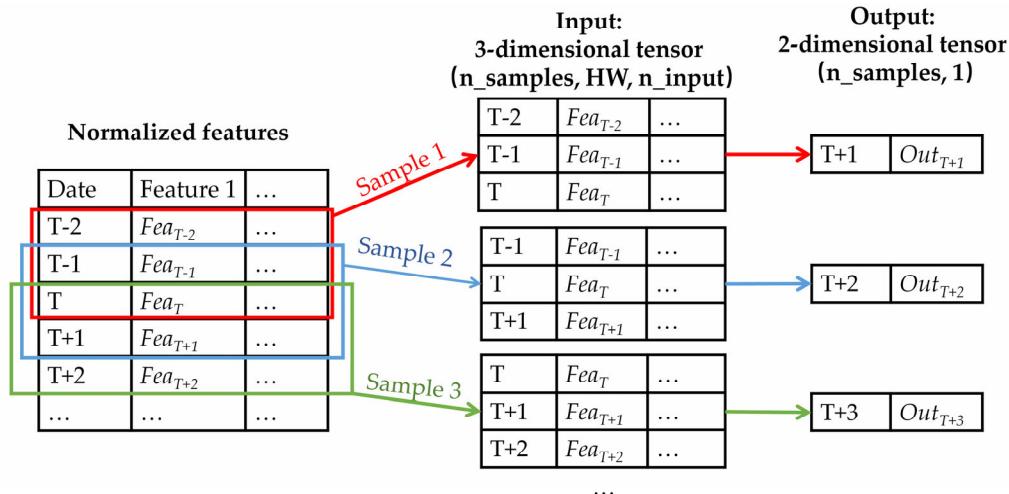
**Figure 4.** Heatmap of Pearson correlation coefficients of various variables.

**Table 1.** Statistical property of input features.

Features	Mean	Standard Deviation	Range
Oil production rate ( $q_o$ ) ( $m^3$ )	39.58	14.64	0–144.375
Tubing Pressure (TP) (MPa)	16.36	8.57	0–38
Shut-In Period (SI) (h)	0.33	2.68	0–24
Choke Size (CS) (mm)	3.16	0.90	0–10

As shown in Figure 5, the features, having been normalized, will be converted to a three-dimensional tensor to match the request of layers in the sixth step. As for input, the first dimension is the number of samples ( $n_{samples}$ ), the second dimension is the length of HW, and the third dimension is the number of input features ( $n_{input}$ ). HW is an important parameter, determining how many days are used to predict the output at the next timestep. To explain more clearly, we assume HW is 3 days,

which means that the model will use the input features of the 3 days ahead to predict the oil production rate of the fourth day. Similar to input, the output is converted to a two-dimensional tensor with  $n\_samples$  as the first dimension and the length of predictive days as the second dimension. Here, we just predict 1 day after the HW, so the second dimension is 1.

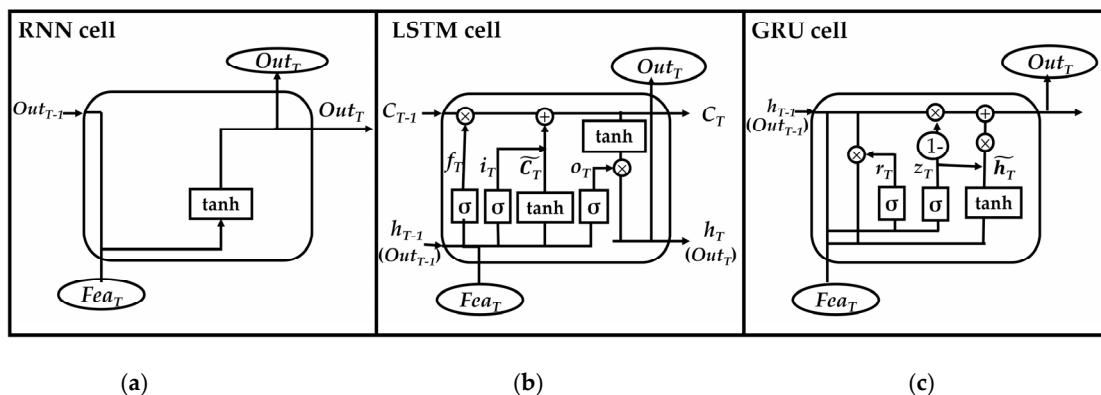


**Figure 5.** Illustration of format conversion of the input and output features.

### 3. Methodology

#### 3.1. RNN, LSTM, and GRU

Based on RNN, LSTM is designed to handle long-term dependency problems [27]. Later Cho et al. proposed a simpler variant GRU [28]. Figure 6 shows the cell architecture of RNN, LSTM, and GRU. Compared with the simple architecture of RNN, LSTM is more complex. The core ideas behind LSTM are the cell state (i.e.,  $c_{T-1}$ ,  $c_T$ ) and three gates, which distinguish LSTM from RNN [32].  $W$  and  $b$  represent the weight matrix and bias matrix of the gates, the subscript  $f$ ,  $i$ ,  $o$  denoting the forget gate, input gate, and output gate, respectively.

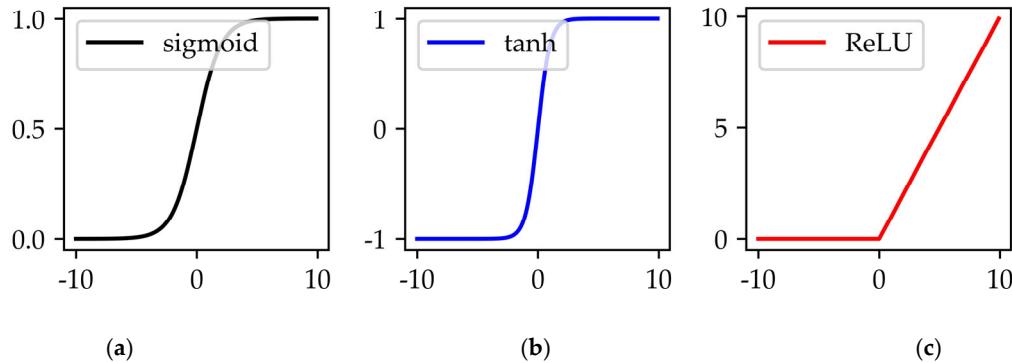


**Figure 6.** Comparison of Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU): (a) RNN; (b) LSTM; (c) GRU.

The cell state ( $c_T$ ) is a conveyor belt running through the top of the LSTM cell, which controls the flow of information across multiple LSTM cells. When the output of timestep  $T - 1$  ( $h_{T-1}$ ) and input of timestep  $T$  ( $Fea_T$ ) enters the cell, the forget gate ( $f_T$ ) will decide what information are not important and should be forgotten.

$$f_T = \sigma(W_f \cdot [h_{T-1}, Fea_T] + b_f) \quad (2)$$

where the symbol  $\sigma$  represents a sigmoid layer, outputting numbers between 0 and 1, given by  $\sigma(x) = (1 + e^{-x})^{-1}$ , as shown in Figure 7a. 0 means “let nothing through” while 1 means “let everything through”. It plays an important role in the three gates.



**Figure 7.** Diagram of three popular activation function: (a) sigmoid; (b) tanh; (c) ReLU.

There is an input gate ( $i_T$ ) combined with the input layer ( $\tilde{C}_T$ ), determining what new information should be stored in the cell state.

$$i_T = \sigma(W_i \cdot [h_{T-1}, Feat_T] + b_i) \quad (3)$$

$$\tilde{C}_T = \tanh(W_C \cdot [h_{T-1}, Feat_T] + b_C) \quad (4)$$

where  $\tanh$  denotes a tanh activation function, as shown in Figure 7b.  $W_C$  is the weight matrix of the input layer.  $b_C$  is the bias matrix of the input layer.

Next, the old cell state ( $C_{T-1}$ ) will be updated to the new cell state ( $C_T$ ) based on Equation (5). The symbol  $\odot$  represents the Hadamard product.

$$C_T = f_T \odot C_{T-1} + i_T \odot \tilde{C}_T \quad (5)$$

Ultimately, the output gate ( $o_T$ ) will decide what information should be output as follows:

$$o_T = \sigma(W_o \cdot [h_{T-1}, Feat_T] + b_o) \quad (6)$$

$$h_T = o_T \odot \tanh(C_T) \quad (7)$$

where  $h_T$  is the hidden state in the timestep  $T$ , which is equal to the value of  $Out_T$ .

On the basis of LSTM, GRU combines the forget gate and input gate into the update gate ( $z_T$ ), and merges the hidden state with cell state, which makes the architecture simpler and computation cheaper. There are only two gates in the cell architecture of GRU, reset gate ( $r_T$ ) and update gate ( $z_T$ ), which can be described as Equations (8) and (9) respectively:

$$r_T = \sigma(W_r \cdot [h_{T-1}, Feat_T] + b_r) \quad (8)$$

$$z_T = \sigma(W_z \cdot [h_{T-1}, Feat_T] + b_z) \quad (9)$$

where  $W_r$  is the weight matrix of the reset gate and  $b_r$  is the bias matrix of the reset gate.  $W_z$  is the weight matrix of the update gate and  $b_z$  is the bias matrix of the update gate.

Subsequently, the new memory cell state will be obtained by Equation (10):

$$\tilde{h}_T = \tanh(W_h \cdot [h_{T-1} \odot r_T, Feat_T] + b_h) \quad (10)$$

where  $W_h$  is the weight matrix of the new memory cell state,  $b_h$  is the bias matrix of the new memory cell state.

The gating signal ( $z_T$ ) ranges from 0 to 1. The closer the gating signal is to 1, the more data will be memorized, whereas the closer to 0, the more forgotten. Therefore, one single expression can control forgetting and inputting, generating output ( $h_T$ ):

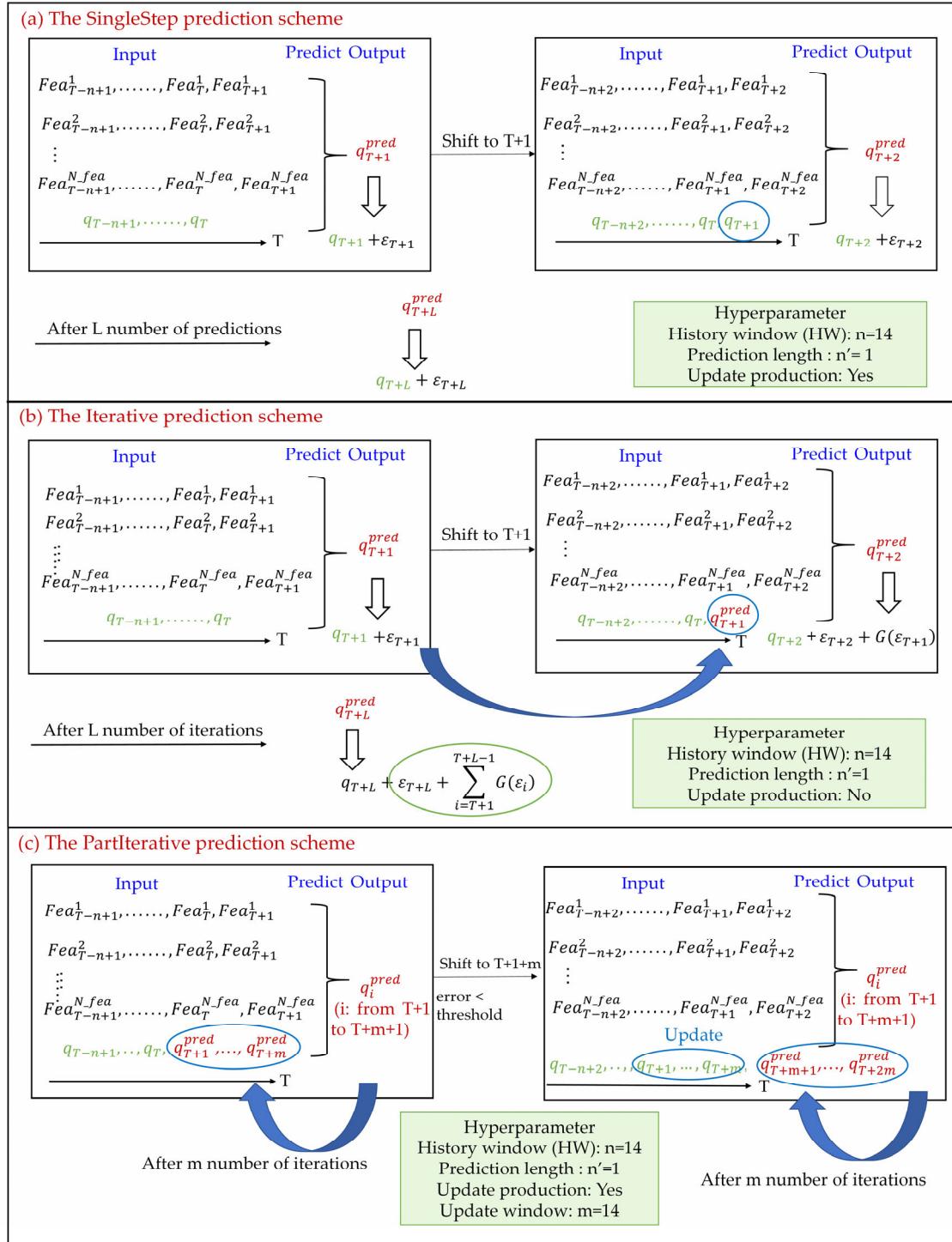
$$h_T = (1 - z_T) \odot h_{T-1} + z_T \odot \tilde{h}_T \quad (11)$$

### 3.2. "SingleStep", "Iterative", and "PartIterative" Prediction Schemes

As the architecture of GRU is flexible and changeable, we explored three prediction schemes: "SingleStep" (see Figure 8a), "Iterative" (see Figure 8b), and "PartIterative" (see Figure 8c). The three schemes were proposed for different purposes. The SingelStep approach was used for validating and evaluating model performance, which eliminating the influence of accumulative errors. The Iterative approach was carried out for long-term cumulative production prediction in the case of limited data. With oilfield data dynamically changing, the PartIterative approach was utilized for real-time production prediction. In Figure 8, the superscripts 1 – N of  $Fea$  represent the available input (exogenous) variables, the subscripts of  $Fea$  represent the sampling time, and  $q_{T+1}^{pred}$  represents the predicted value of oil production rate at time  $T + 1$ . To eliminate the influence of HW and prediction length, the HW was set to 14 days, and the prediction length was set to 1 day. To remove the effect of the number of iterations, the SingleStep approach was put forward to evaluate the model performance on the single-step error fairly. The production rate used as "input" was updated daily, which means that the new production was forecasted based on true values. In other words, given external features  $Fea$  from  $T - n + 1$  to  $T + 1$  and actual production value  $q$  from  $T - n + 1$  to  $T$ , we could predict the  $(T + 1)$ th production rate  $q_{T+1}^{pred}$ . When shifting to the next timestep, given external features  $Fea$  from  $T - n + 2$  to  $T + 2$  and actual production value  $q$  from  $T - n + 2$  to  $T + 1$ , we could predict the  $(T + 2)$ th production rate. Without replacing  $q_{T+1}$  with  $q_{T+1}^{pred}$ , the influence of the number of iterations was eliminated, and it was more straightforward to evaluate the model performance on a single timestep. In this way, the RMSE of a well throughout the whole production lifecycle could be calculated, which was used for demonstrating the validity of the model.

In the Iterative scheme, the daily production was not be updated and the predictions were implemented on the input features of the first HW and previously predicted production. Unlike the SingleStep scheme, when shifting to the next timestep, given external features  $Fea$  from  $T - n + 2$  to  $T + 2$  and actual production value  $q$  from  $T - n + 2$  to  $T + 1$ , we replaced  $q_{T+1}$  with  $q_{T+1}^{pred}$  for forecasting the  $(T + 2)$ th production rate  $q_{T+2}^{pred}$ . Then we moved the timestep to  $T + 3$ , replace  $q_{T+2}$  with  $q_{T+2}^{pred}$ , predicted the  $(T + 3)$ th production rate  $q_{T+3}^{pred}$ , and so on. As shown in Figure 8b, the first prediction generated an error  $\varepsilon_{T+1}$ , and the next timestep generated an error  $\varepsilon_{T+2} + G(\varepsilon_{T+1})$ . After L number of iterations, the error was accumulated to  $\varepsilon_{T+L} + \sum_{i=T+1}^{T+L-1} G(\varepsilon_i)$ .

To make full use of the dynamic production data, we combined the above two schemes and proposed the PartIterative approach, as depicted in Figure 8c. The update window represents the number of  $q$  updated at one time. This scheme starts with an iterative process. Given external features  $Fea$  from  $T - n + 1$  to  $T + 1$  and actual production value  $q$  from  $T - n + 1$  to  $T$ , we could predict the  $(T + 1)$ th production rate  $q_{T+1}^{pred}$ . Then we moved to the next time step, given  $Fea$  from  $T - n + 2$  to  $T + 2$ , actual production value  $q$  from  $T - n + 1$  to  $T$  and  $q_{T+1}^{pred}$ , we predicted the  $(T + 2)$ th production rate  $q_{T+2}^{pred}$ . Repeating the above iterative process for m times (size of update window), we updated the predicted value  $q_{T+1}^{pred}, \dots, q_{T+m+1}^{pred}$  into  $q_{T+1}, \dots, q_{T+m+1}$  and started the iterative process again and so on. Between iterations and updates, there is an adaptive system that compares the predicted values with the actual ones to judge whether the model should be retrained or not.



**Figure 8.** Diagram for the SingleStep, Iterative, and PartIterative prediction schemes: (a) the SingleStep approach; (b) the Iterative approach; (c) the PartIterative approach.

### 3.3. The Architecture of the GRU Model

In this paper, we utilized the GRU algorithm based on the python-based deep learning library Keras [33] configurated with the TensorFlow backend. The model consisted of the input, GRU, and dense layers. Hyperparameter optimization was not the focus of this research, so the neural network architecture was simply tuned according to the performance of the validation set. The number of neurons in the GRU layer was 32, big enough to learn the implicit information behind input and output. All the weight

matrixes were initialized by the Xavier initialization [34], and the bias matrixes were initialized to zeros. The activation function of the GRU layer is ReLU, as shown in Figure 7c. Adaptive moment estimation (Adam) was used as an optimizer to minimize the loss function. Combining the advantages of AdaGrad and RMSProp, it calculates the update weights and bias based on the first moment estimation and the second moment estimation meanwhile [35]. Least squares (L2 norm) regularization [36] and dropout technique [37] were also used to avoid overfitting. Table 2 shows the parameters of the proposed GRU models. The models were coded in Python 3.7 and executed on Intel® Core™ i7-4790 3.60 GHz CPU.

**Table 2.** The parameters of the trained GRU model.

Parameters	Value
Number of neurons	32 for the GRU layer; 1 for the dense layer
HW	14 days
Index of shuffle	100
Epoch	50
Batch size	50
Loss function	MSE
Optimizer	Adam
Activation function	ReLU
Regularization parameter of L2 norm ( $\lambda$ )	0.01
Dropout rate	0.1
Learning rate ( $\eta$ )	0.001

### 3.4. Random Forest

RF is an ensemble algorithm consisting of a collection of decision trees that calculate ensemble predicted values by averaging values of terminal nodes for regression problems [38]. Numerous studies have been implemented to prove that RF has high accuracy, good tolerance to noise and outliers, and the ability to avoid overfitting [39–41]. Further, it provides a novel method of determining feature importance, which could help us further understand the mechanism of fracturing stimulation. There are two straightforward methods for feature importance ranking: mean decrease impurity and mean decrease accuracy. The first method is implemented in the RF of Scikit-learn [42] and was adopted in this research. For every decision tree in the forest, the nodes were split at the most informative feature to maximize the information gain, as defined in Equation (12). Then the feature importance was equal to the sum over the number of splits that include the feature, relying on the number of samples it splits. The detailed implementations of RF can refer to [38,43].

$$IG(D_p, Fea) = I(D_p) - \frac{N_{left}}{N_p}I(D_{left}) - \frac{N_{right}}{N_p}I(D_{right}) \quad (12)$$

where  $Fea$  is the feature to perform the split,  $N_p$  is the number of samples in the parent node,  $N_{left}$  and  $N_{right}$  are the number of samples in child nodes separately,  $I$  is the impurity function, which refers to the variance for regression problems.  $D_p$  is the sample at the parent node,  $D_{left}$  and  $D_{right}$  are samples at child nodes.

### 3.5. Evaluation Indices

In this research, we adopted mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) as evaluation indices. MSE can evaluate the differences between actual values ( $y_{true}$ ) and predicted values ( $y_{pred}$ ), represented as Equation (13), which can be used to measure the overall performance of the trained model and help reduce forecasting error. Compared with MSE, RMSE represents the average error of a single sample, which helps to understand the physical

significance, as defined in Equation (14). MAE gives a direct measure of the difference between predicted outcomes and true values, as defined in Equation (15).

$$MSE = \frac{\sum_{i=1}^{n\_samples} (y_{pred} - y_{true})^2}{n\_samples} \quad (13)$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^{n\_samples} (y_{pred} - y_{true})^2}{n\_samples}} \quad (14)$$

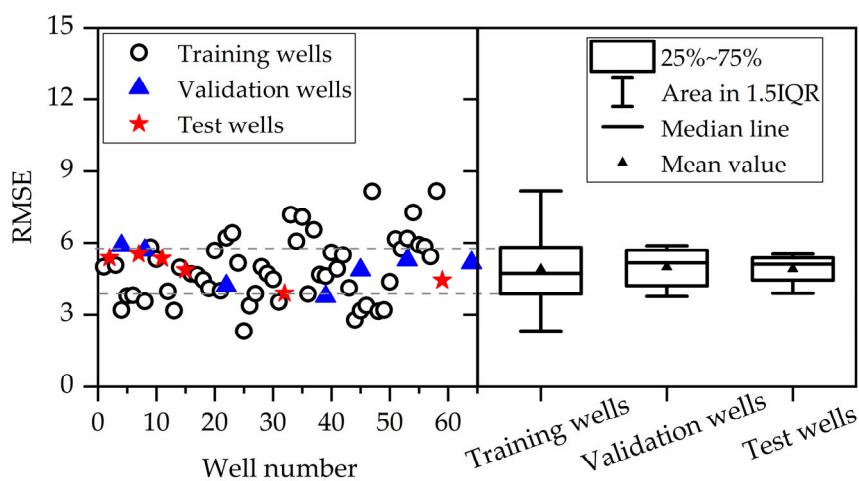
$$MAE = \frac{1}{n\_samples} \sum_{i=1}^{n\_samples} |y_{pred} - y_{true}| \quad (15)$$

## 4. Results and Discussion

### 4.1. Verification of the GRU Model with the SingleStep Approach

In this research, GRU is conducted for production rate prediction under multiple factors in the conglomerate reservoir for the first time. For a new well, given inputs of the first HW, the long-term production rate can be obtained according to the proposed well-block GRU model. Once the proposed model is trained with training wells, we can apply it to other wells without extra computational cost. For ease of understanding, the forecasting results have been denormalized (multiplying the original RMSE and the standard deviation of the training set).

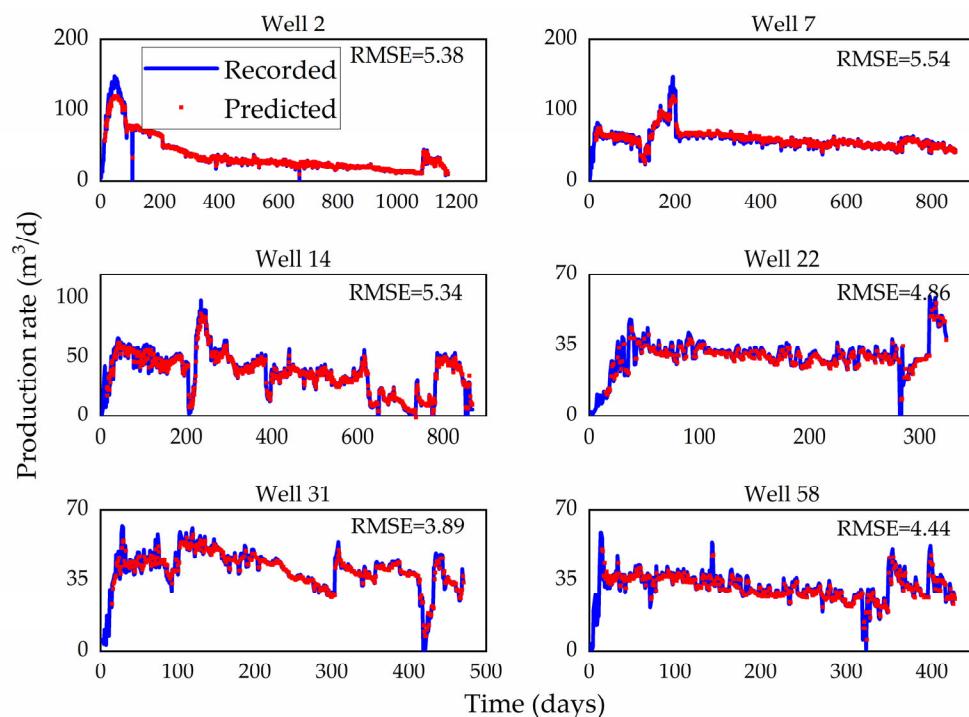
Figure 9 displays the RMSE of the proposed model on training wells, validation wells, and test wells. Gray dashed lines represent the area between 25% and 75% quantiles. As can be seen from Figure 9, RMSE for most wells is in the range of 4 to 6. The mean value of training wells, validation wells, test wells is 4.89, 4.98, and 4.91, respectively. The RMSE distribution of test wells coincides with that of the training wells, suggesting the generalization ability of the trained model.



**Figure 9.** Root mean square error (RMSE) of predicted results with the SingleStep approach.

To evaluate the generalization ability of the proposed model, we focus on the performance on the test set. Figure 10 shows the prediction results of the GRU model on six test wells with the SingleStep prediction approach. The SingleStep approach is used to eliminate the influence of accumulative errors caused by iterations. In other words, every red dot is predicted based on the actual inputs of the previous 14 days before the current prediction point. RMSE is utilized as an index to evaluate the single-step error. As we can see from Figure 10, most predicted points are very close to actual

production value, especially when the oil rate changes steadily, such as the oil rate of Well 2 from the 200th to 1100th day and Well 7 from the 200th to 700th day. However, the relative errors soar when enormous fluctuations or sudden variances occur, such as the oil rate of Well 2 at the 107th day, Well 7 around the 200th day, and frequent variances of Well 58, Well 22, and Well 31. The sudden drop or rise could lead to bigger uncertainties and errors for the prediction, which is rarely considered in previous studies. The prediction results show that the trained model can capture the complex and flexible variance of production, whereas the ability to capture sharp and sudden variances is limited. In the future, we can smooth the area when there is a brief and sudden change or extend the forecast unit from day to month to improve model accuracy.

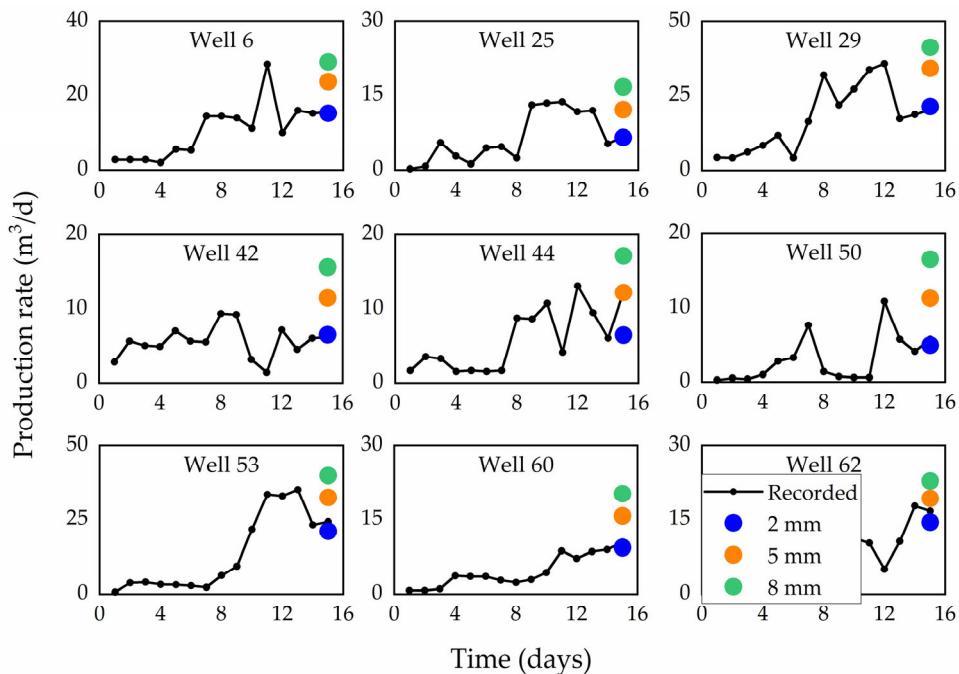


**Figure 10.** The predicted production of test wells with GRU and the SingleStep approach.

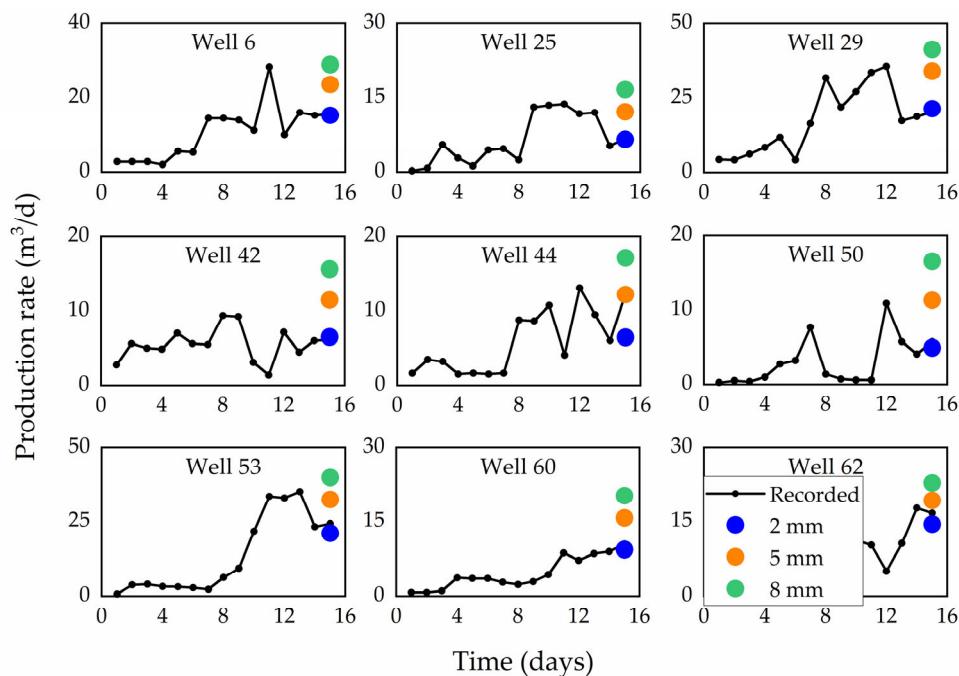
The proposed GRU model takes choke size and shut-in operation into account, providing a method to predict the oil rate under variable external features. The predicted production can be changed by CS and SI. In total, nine training wells were selected to test the sensitivity of changing external features to the production rate. As shown in Figure 11, when the CS is changed from 2 mm, 5 mm to 8mm, the larger the choke size, the higher the production rate will be. As the SI is modified from 0 h, to 12 h to 24 h, the oil rate shows that production will increase with the shut-in period prolonged (see Figure 12). It is reasonable because the pressure will build up during the shut-in period. The predicted results are consistent with field experience and existing understanding.

#### 4.2. Application of the GRU Model with the Iterative Approach

In application, we often need to predict a longtime production provided with limited production history. In this research, we also explored the Iterative approach, in which we repeat the iterations using the predicted production. More narrowly, given data of the first HW (14 days), we predict the production rate of the 15th day. Based on the actual rate from the 2nd to the 14th day and the predicted rate of the 15th day, we can predict the rate of the 16th day. Then based on the actual rate from 3rd to 14th day and the predicted rate from 15th to 16th day, we can predict the rate of the 17th. Repeat the above iterations, and we can get the predicted production rate at any point in time.



**Figure 11.** Effect of the CS on production rate.

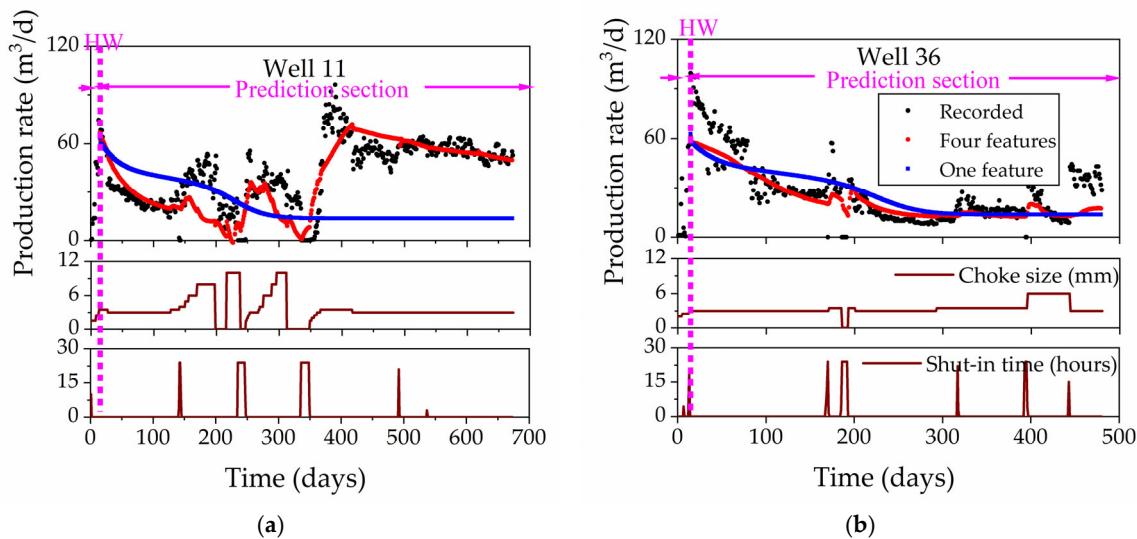


**Figure 12.** Effect of the SI on production rate.

The one-feature ( $q_0$ ) model and the four-feature ( $q_0 + \text{TP} + \text{SI} + \text{CS}$ ) model were firstly applied to two training wells. As shown in Figure 13, Well 11 and Well 36 are selected to show the influence of iterations under different features. The first 14 days serve as HW, and the rest is treated as the prediction section.

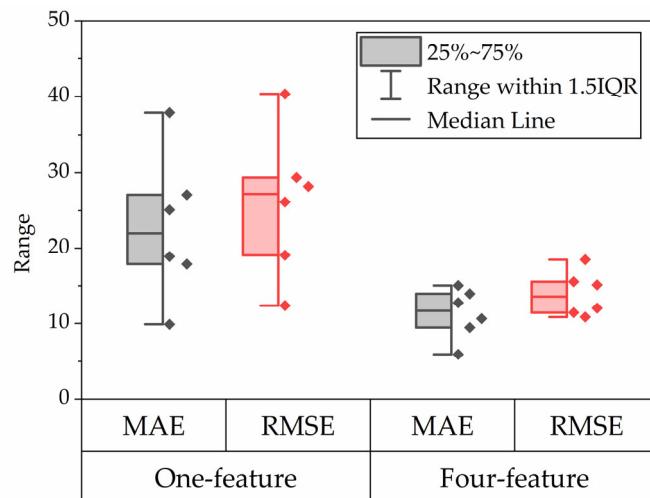
Firstly, we focus on the comparison of the two models. It can be found that the one-feature model can only capture the descent trend and are not sensitive to oilfield operations, which could lead to enormous errors. For Well 36 with a stable decline trend production and fewer operations, both models can predict a reasonable production and the four-feature model shows a better performance. Obviously, for Well 11 with an undulating production, the predicted points of the four-feature model coincide

with changing choke size and shut-in operation, pointing out the four-feature model can capture the complex variation pattern under multiple factors.



**Figure 13.** Comparison of the one-feature model and the four-feature model with the Iterative approach under multiple factors; (a) Well 11; (b) Well 36.

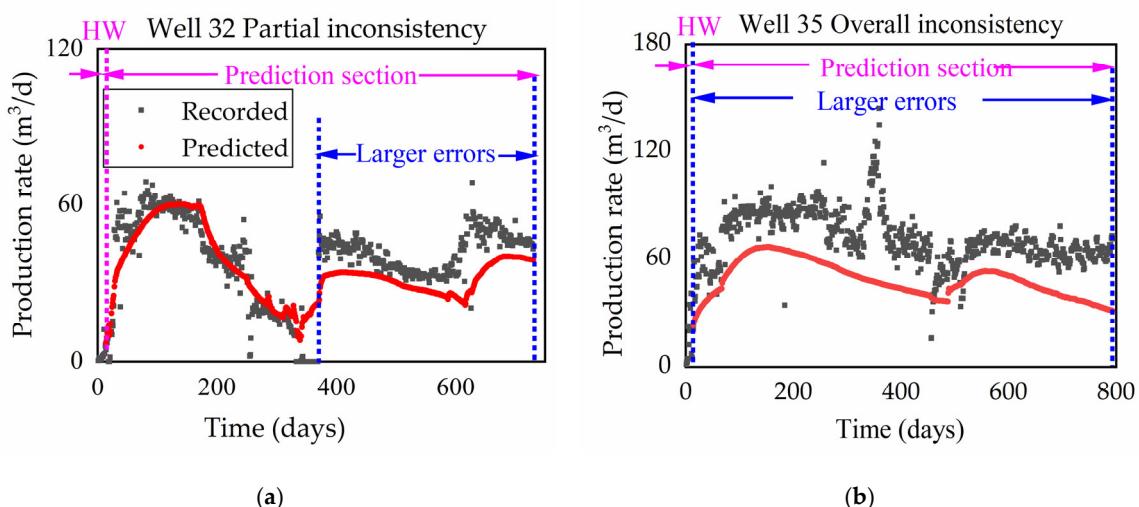
Figure 14 shows the half boxplot of the mismatch between the two GRU models on six test wells. The difference between the predicted and true production rate of test wells is quantified by MAE and RMSE. The mean value of RMSE from the four-feature model, 13.84, is much less than that of the one-feature model, which is 25.91. Similarly, the mean value of MAEs from the four-feature model, 11.21, is less than that of the one-feature model, which is 22.79. Many scholars have corroborated the feasibility of RNN and LSTM algorithms in production prediction, but few focus on the influence of external variables on prediction performance. In this research, we compared the two models and found that the ability of the four-feature model to capture variances and prediction accuracy outperforms the one-feature model.



**Figure 14.** Half boxplot of mean absolute error (MAE) and RMSE of the one-feature model and four-feature model with the Iterative approach on test wells.

Then, we focus on the performance of the four-feature model itself. It can be found from Figure 13 that the production data are not monotonically decreasing but display a complex variation pattern because of frequent operations. The proposed model could capture the complicated changes

and fluctuate with changing choke size and shut-in operation. However, the available production history, long prediction section, and frequent operations increase the difficulties of iterative prediction. Although errors of the SingleStep scheme are small, the errors become bigger if the Iterative approach is used. Figure 15 shows some typical prediction results of training wells with big relative errors. The first is partial inconsistency, which results from oilfield operations. The ability of the trained model to capture sharp and sudden variances is limited, which leads to predicted values usually lower than actual values when CS is changed. This phenomenon is obvious in Well 32. The shut-in operation around the 380th day results in a stable gap between the predicted and actual values after the operation. The second is the overall trend lower than actual points, such as Well 35, which might be caused by the diversity of formation and fracturing conditions. The changing trend of predicted points is consistent with the recorded production rate, but a stable difference exists during the whole prediction. Although all the wells belong to the same well block, the geological and fracturing features change with the distribution of fault blocks and the advance of stimulation technologies. Another possible explanation is that the production of the first 14 days is still on the rise, which will provide an inaccurate baseline for later iterative prediction. The errors also occur in test wells when using the Iterative approach for prediction. Although the accuracy of single-point prediction is pretty high, the performance becomes worse as the number of iterations increase. The mean RMSE of test wells with the Iterative approach is 13.91, 2.8 times of the SingleStep approach, which is 4.91. The increased error is the comprehensive effect of numerous iterations, sharp variances, and diversity of formation and fracturing.



**Figure 15.** Typical prediction results of training wells with big relative errors: (a) partial inconsistency; (b) overall inconsistency.

#### 4.3. GRU\_RF Model for Improving “Iterative” Accuracy

To improve the prediction accuracy, we set up an RF model to fit the errors between the predicted and true production rate. The allocation scheme of the training set, validation set, and test set is the same as that of the GRU model. Considering two kinds of errors, the inputs of RF include dynamic and static features relating to production, operations, formation, and fracturing, as listed in Table 3. The number of estimators is set to 50, and the maximum depth is 3.

The overall workflow of the GRU\_RF model is illustrated in Figure 16. The proposed model in the blue dashed box consists of two sub-models: the GRU model and the RF model. Each step in the flowchart is described in detail below:

- Step 1. Training the GRU model. After data preprocessing (as described in Figure 2), the data set are split into three parts, and the GRU model is fed by training wells and optimized by validation wells.

- Step 2. Preliminary forecasting with the GRU model and Iterative prediction approach. Based on historical production data of HW, we can get the time-series predicted production by continuous iterations.
- Step 3. Training the RF model. To revise the GRU model's predicted errors caused by iterations and diversity of formation and fracturing, the RF model is trained with the predicted errors between the predicted production of the GRU model and the true production as output. Remarkably, the RF model is fed by the same training wells as the GRU model.
- Step 4. Second forecasting with the RF model. Given inputs of production, operations, formation, and fracturing parameters, the correction errors can be predicted by the RF model.
- Step 5. Forecasting with the GRU\_RF model for test wells. The purpose is to use the trained model applicable to the whole well block to forecast the long-term production of new wells. Hence, given inputs of HW, we can directly get the predicted production (Step 2) and the predicted errors (Step 4) without training the model repeatedly, as shown in the deep red flow line in Figure 16. The predictions of the GRU\_RF model are obtained by subtracting the predicted errors of the RF model from the predicted production of the GRU model.

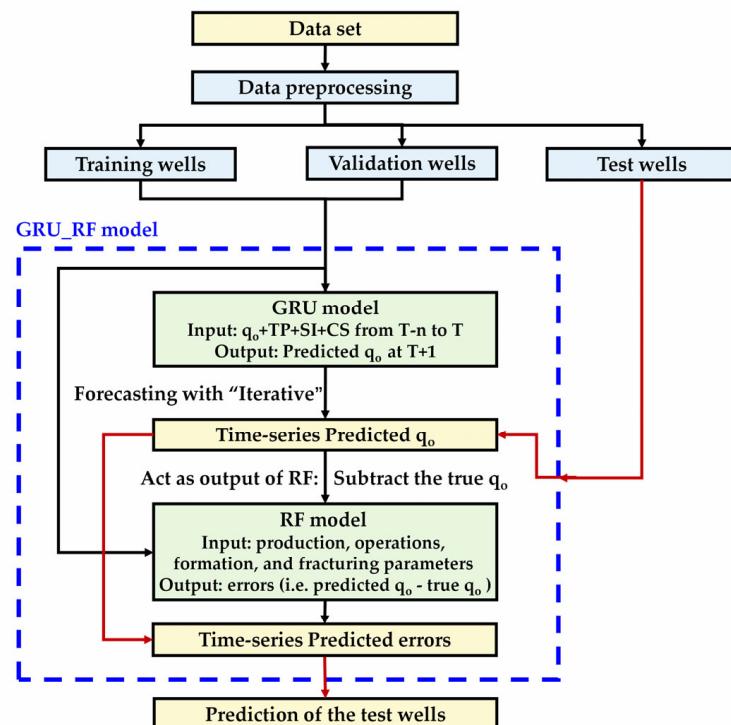
**Table 3.** The input and output features of the Random Forest (RF) model.

	Input Features	Symbol
Production	predicted production by GRU	Predicted $q_T$
	the predicted production of the previous 7 days by GRU	Predicted $q_{T-1}, q_{T-2}, q_{T-3}, q_{T-4}, q_{T-5}, q_{T-6}, q_{T-7}$
	Cumulative production of the first 14 days	$Q_{14 \text{ days}}$
	Cumulative production of the first 30 days	$Q_{30 \text{ days}}$
Operations	Number of iterations (production time)	$N_{\text{iter}}$
	Choke size	CS
	Shut-in time	SI
	changes of choke size	$CS_T - CS_{T-1}$
Fracturing	changes of shut-in time	$SI_T - SI_{T-1}$
	Total sand per well	Total sand
	Total liquid per well	Total liquid
Formation	Well lateral length	Length
	The minimal horizontal principle stress	MHPS
	Oil saturation	$S_{\text{oil}}$
	Porosity	Poro
Output	Well vertical depth	TVD
	the errors between the predicted and true production rate	Error

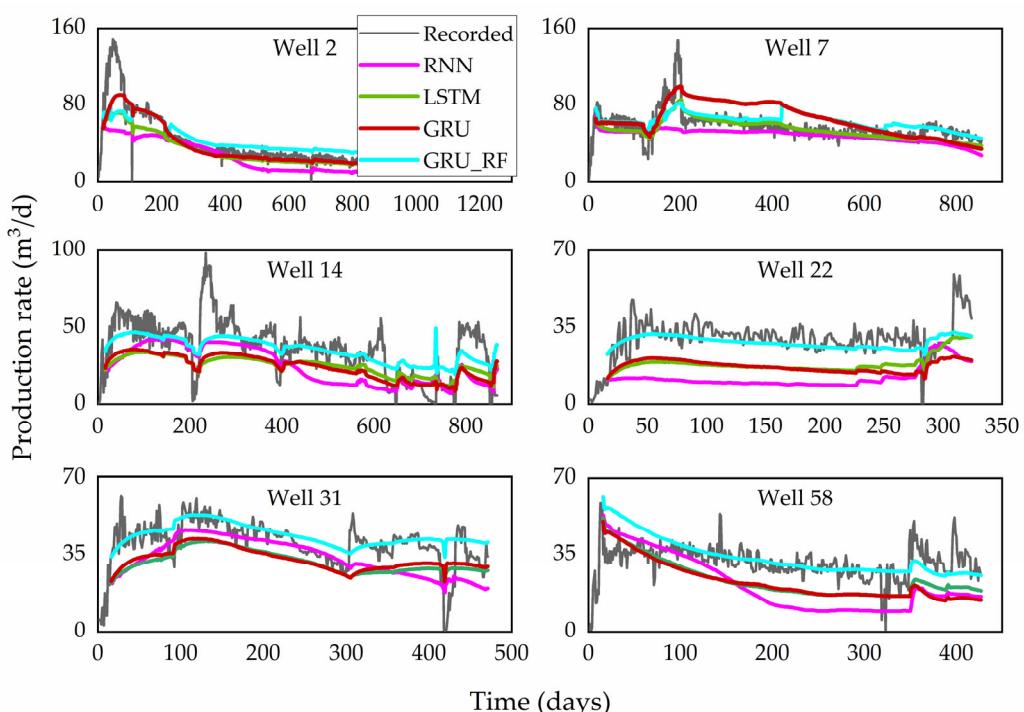
For a fair evaluation, the accuracy of the proposed GRU\_RF model is compared with that of the fully connected RNN, LSTM and the baseline GRU. To be consistent with GRU\_RF, data of the first 14 days are treated as training data, and the rest are test data. The model architecture, prediction approach, and external features of RNN and LSTM are the same as GRU. The values of parameters used to train RNN and LSTM are also the same as GRU, as listed in Table 2. For the RNN/LSTM layer, the number of neurons is 32. All the models are trained and utilized with the Iterative prediction approach and four-feature inputs. Figure 17 shows the prediction results of RNN, LSTM, GRU, and GRU\_RF. All of the results are consistent with the trend of the production for the test wells. However, the results of GRU\_RF are with smaller errors, suggesting pretty model generalizability.

Figure 18 compares the prediction performance of RNN, LSTM, GRU, and GRU\_RF on test wells. It is shown that GRU\_RF outperforms the other three methods. The mean RMSE of GRU\_RF is 10.67, smaller than the mean RMSE of RNN, LSTM, and GRU, which is 16.82, 13.49, and 13.91, respectively. The mean MAE of GRU\_RF is 7.55, less than that of RNN, LSTM, and GRU, which is 13.51, 10.60, and 11.29, respectively. In general, the effect of the GRU\_RF is better than RNN, LSTM, and GRU. It is

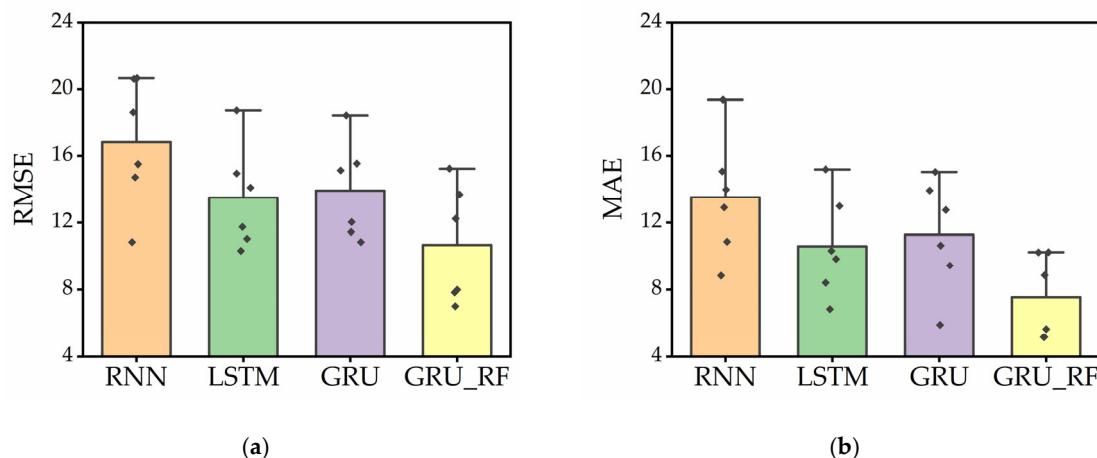
worth noting that only inputs of the first HW are available during the whole prediction. Although the predicted points are not in perfect agreement with the recorded production curve, it is exciting that the trained model provides a feasible method to predict long-term production for newly developed wells with limited production data. The traditional DCA and conventional prediction methods cannot be applied to new wells with little data easily.



**Figure 16.** The overall workflow of the proposed GRU\_RF model.



**Figure 17.** The predicted production of test wells with RNN, LSTM, GRU, and GRU\_RF.



**Figure 18.** Comparison of prediction performance of RNN, LSTM, GRU, and GRU\_RF on test wells with two evaluation indices: (a) RMSE; (b) MAE.

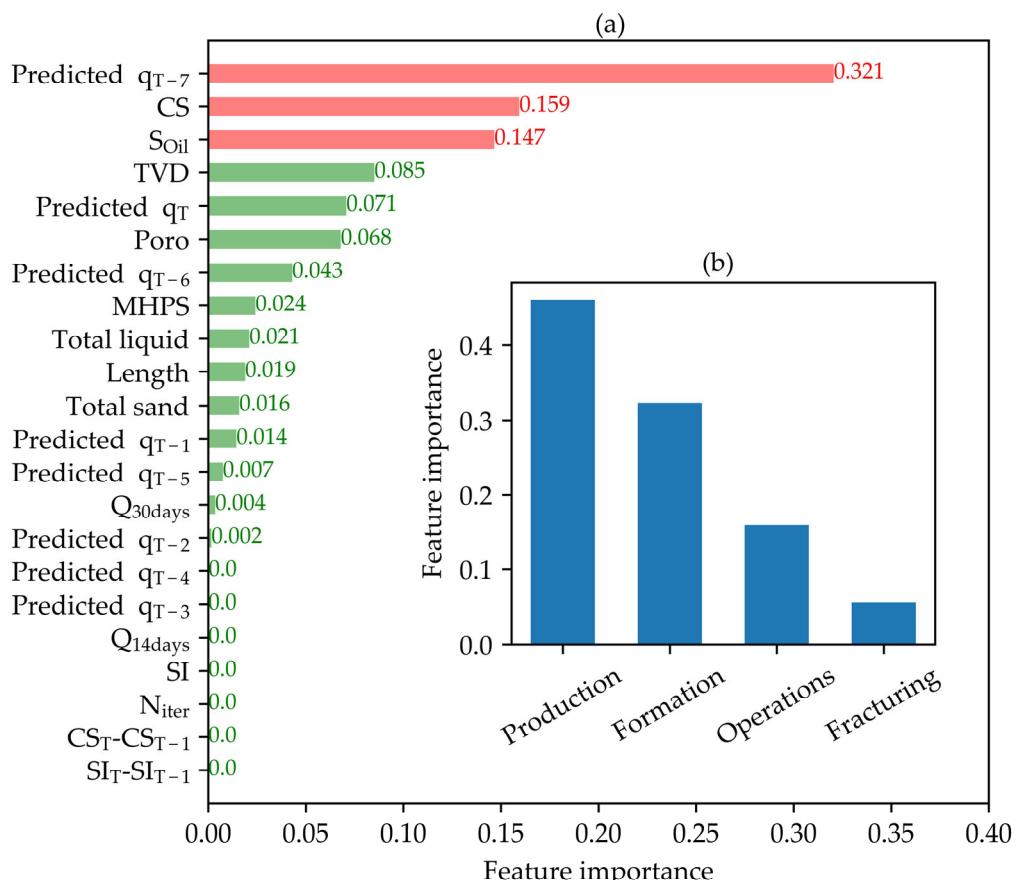
The proposed model has advantages over traditional DCA and conventional time-series prediction methods. Firstly, it can be applied to new wells with production time of over 2 weeks. For most of these new wells, the initial production rate shows an increasing trend because of the wellbore storage effect. Traditional DCA is nowhere to be utilized in wells without a declining trend [9]. The conventional time-series prediction methods (i.e., ARMA, ARIMA, ARIMAX) are linear models based on statistics [8,44]. Learning from the initial production will obtain a rising prediction trend contrary to the long prediction section, which will lead to huge errors. Therefore, neither traditional DCA nor conventional time-series prediction methods can achieve the production prediction of wells with short production time. Secondly, the proposed model can achieve batch forecasting of multiple wells. Additionally, it is much faster because it doesn't need to train the GRU\_RF model every time. However, the other two methods have to tune multiple model hyperparameters (such as initial decline rate, decline index, autoregression coefficient, moving average coefficient) with wells changed, which is time-consuming and cumbersome. Thirdly, it takes CS and SI into account, offering a way to analyze the effect of external features on the production rate. As we can see from Figures 11 and 12, the predicted rate will change with CS and SI changes. However, traditional DCA, ARMA and ARIMA cannot consider the effect of CS and SI on production rate. Although ARIMAX can consider external features, it has many restrictions in application due to strict statistical background [45], such as strong correlations between the dependent feature and external features, which constrain its application in production prediction. Finally, the proposed model prevents the subjectivity of analysts from affecting the predicted results. The same inputs will guarantee the same output. However, the predicted values of DCA vary with the selection of the starting point, which will result in great uncertainty for production prediction [10].

Figure 19 shows the feature importance ranking determined by RF. It can be seen from Figure 19a that the most important features for model revising are the predicted rate 7 days ago, CS and oil saturation, which account for more than 60% of feature importance. Figure 19b displays the importance scores of different feature types. Production, formation conditions, and oilfield operations play an important role in revising the production rate. It is consistent with the previous analysis of the cause of errors, in which the rising trend of initial production, various oilfield operations, and differences in formation and stimulation will affect the single well prediction. This infers that it is essential to take the geological and completion features into account when building a model for a whole well block.

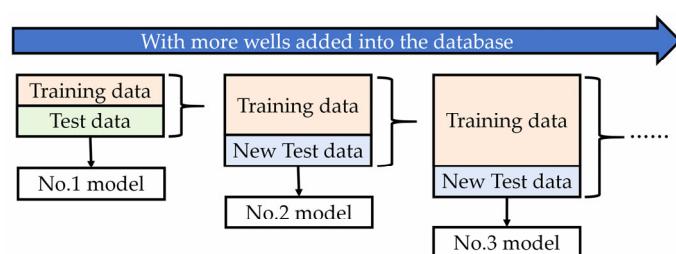
#### 4.4. Continuous Learning

Production is a dynamic and coupled process, related to fracture propagation, proppant failure, formation property variation, and oilfield operations. With the production data updated dynamically,

we plan to introduce continuous learning into production prediction. Continuous learning is the concept of learning continuously and adaptively about the external world and enabling the autonomous incremental development of ever more complex skills and knowledge [46,47]. Here continuous learning refers to updating the history window and trained models for future production predictions during the whole production period. For one new well, continuous learning refers to the PartIterative scheme, as depicted in Figure 8c, in which the HW will be updated as time goes on to make full use of production data. For multiple wells, continuous learning means that as data streams into the database, the model is updated as soon as enough new data are available, as shown in Figure 20. Firstly, the currently available data are divided into the training set and test set. The No.1 model is obtained based on current data. With new data added into the database, the old data serve as the training set, and new data become the test set. The No.2 model can be developed and the old data become the training set and so on. In this way, the well-block model could be updated continuously, similar to the cognitive process from simple to deep. In the future, we will further explore the continuous prediction to make full use of available data.



**Figure 19.** Feature importance ranking of RF: (a) sorted by single parameter; (b) sorted by parameter types.



**Figure 20.** The diagram for continuous learning in production prediction.

## 5. Conclusions

In this research, a novel hybrid model GRU\_RF considering multiple external factors was proposed for the production prediction of the X well block. A thorough data preprocessing and modeling procedure were explained, which can be migrated to other reservoirs for batch production prediction. The prediction unit and length are changeable with flexible model architecture.

The major conclusions are as follows:

- (1). The proposed GRU\_RF model provided a promising method for predicting long-term production of new-developed wells with the initial production, as evidenced by six test wells. Its generalization ability was satisfactory when applying to wells that do not join in the training set.
- (2). The model considering SI and CS provided a method to explore the effects of variable external features. The four-feature model outperformed the conventional one-feature model. It could accurately capture the complex variance pattern under multiple factors.
- (3). In addition to production and oilfield operations, formation and fracturing parameters significantly affected production prediction of multiple wells from one well block, as illustrated by feature importance ranking of RF. It was essential to combine statistic features with dynamic features into the time-series prediction.
- (4). The proposed model outperformed traditional DCA, conventional time-series methods, simple RNN, LSTM, and GRU. Given limited production, the model could achieve batch processing and provide a continuous learning method for real-time prediction.

**Supplementary Materials:** The following are available online at <http://www.mdpi.com/1996-1073/13/22/6121/s1>, 70 Excel databases containing well data recorded during this study.

**Author Contributions:** Methodology, X.L.; validation, X.L. and X.M.; investigation, X.L. and F.X.; data curation, X.L. and S.Z.; writing—original draft preparation, X.L. and F.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was jointly funded by the National Major Science and Technology Projects of China (2017ZX05049-006), the National Natural Science Foundation of China (51974332 and U1762210) and major projects in Karamay (2018ZD001B).

**Acknowledgments:** The authors would like to express their gratitude to the support of the National Major Science and Technology Projects of China, the National Natural Science Foundation of China and major projects in Karamay.

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

## References

1. Hennigan, H.W.; Canon, M.W.; Ziara, B.A. An interactive production forecasting and economic evaluation system. In Proceedings of the SPE Annual Technical Conference and Exhibition, San Antonio, TX, USA, 4–7 October 1981.
2. Wang, N.; Zhao, Q. China shale gas economic evaluation based on Monte Carlo simulation. In Proceedings of the 22nd World Petroleum Congress, Istanbul, Turkey, 9–13 July 2017.
3. Wright, J.D. Economic evaluation of shale gas reservoirs. In Proceedings of the SPE Shale Gas Production Conference, Fort Worth, TX, USA, 16–18 November 2008; pp. 136–145.
4. Wilson, K.C.; Durlofsky, L.J. Optimization of shale gas field development using direct search techniques and reduced-physics models. *J. Pet. Sci. Eng.* **2013**, *108*, 304–315. [[CrossRef](#)]
5. Yu, W.; Luo, Z.; Javadpour, F.; Varavei, A.; Sepehrnoori, K. Sensitivity analysis of hydraulic fracture geometry in shale gas reservoirs. *J. Pet. Sci. Eng.* **2014**, *113*, 1–7. [[CrossRef](#)]
6. Nwaobi, U.; Anandarajah, G. Parameter determination for a numerical approach to undeveloped shale gas production estimation: The UK Bowland shale region application. *J. Nat. Gas Sci. Eng.* **2018**, *58*, 80–91. [[CrossRef](#)]
7. Arps, J.J. Analysis of Decline Curves. *Trans. AIME* **1945**, *160*, 228–247. [[CrossRef](#)]

8. Olominu, O.; Sulaimon, A.A. Application of time series analysis to predict reservoir production performance. In Proceedings of the SPE Nigeria Annual International Conference and Exhibition, Lagos, Nigeria, 5–7 August 2014; Volume 1, pp. 569–582.
9. Tan, L.; Zuo, L.; Wang, B. Methods of decline curve analysis for shale gas reservoirs. *Energies* **2018**, *11*, 552. [[CrossRef](#)]
10. Lee, K.; Lim, J.; Yoon, D.; Jung, H. Prediction of shale-gas production at duvernay formation using deep-learning algorithm. *SPE J.* **2019**, *24*, 2423–2437. [[CrossRef](#)]
11. Ma, Z.; Leung, J.Y.; Zanon, S.; Dzurman, P. Practical implementation of knowledge-based approaches for steam-assisted gravity drainage production analysis. *Expert Syst. Appl.* **2015**, *42*, 7326–7343. [[CrossRef](#)]
12. Shaheen, M.; Shahbaz, M.; Ur Rehman, Z.; Guergachi, A. Data mining applications in hydrocarbon exploration. *Artif. Intell. Rev.* **2011**, *35*, 1–18. [[CrossRef](#)]
13. Wang, S.; Chen, S. Insights to fracture stimulation design in unconventional reservoirs based on machine learning modeling. *J. Pet. Sci. Eng.* **2019**, *174*, 682–695. [[CrossRef](#)]
14. Al-Fattah, S.M. Time series modeling for U.S. natural gas forecasting. In Proceedings of the 2005 International Petroleum Technology Conference, Doha, Qatar, 21–23 November 2005; pp. 973–975.
15. Gupta, S.; Fuehrer, F.; Jeyachandra, B.C. Production forecasting in unconventional resources using data mining and time series analysis. In Proceedings of the SPE/CSUR Unconventional Resources Conference—Canada, Calgary, AB, Canada, 30 September–2 October 2014; Volume 1, pp. 247–254.
16. Morgan, E. Accounting for serial autocorrelation in decline curve analysis of Marcellus shale gas wells. In Proceedings of the SPE/AAPG Eastern Regional Meeting, Pittsburgh, PA, USA, 7–11 October 2018.
17. Udegbe, E.; Morgan, E.; Srinivasan, S. From face detection to fractured reservoir characterization: Big data analytics for restimulation candidate selection. In Proceedings of the SPE Annual Technical Conference and Exhibition, San Antonio, TX, USA, 9–11 October 2017.
18. Li, Y.; Sun, R.; Horne, R. Deep learning for well data history analysis. In Proceedings of the SPE Annual Technical Conference and Exhibition, Calgary, AB, Canada, 30 September–2 October 2019.
19. Madasu, S.; Rangarajan, K.P. Deep recurrent neural network DRNN model for real-time multistage pumping data. In Proceedings of the OTC Arctic Technology Conference, Houston, TX, USA, 5–7 November 2018.
20. Quishpe, A.R.; Alonso, K.S.; Claramunt, J.I.A.; Barros, J.L.; Bizzotto, P.; Ferrigno, E.; Martinez, G. Innovative artificial intelligence approach in vaca muerta shale oil wells for real time optimization. In Proceedings of the SPE Annual Technical Conference and Exhibition, Calgary, AB, Canada, 30 September–2 October 2019.
21. Azamifard, A.; Rashidi, F.; Ahmadi, M.; Pourfard, M.; Dabir, B. Toward more realistic models of reservoir by cutting-edge characterization of permeability with MPS methods and deep-learning-based selection. *J. Pet. Sci. Eng.* **2019**, *181*, 106135. [[CrossRef](#)]
22. Etienam, C. 4D seismic history matching incorporating unsupervised learning. In Proceedings of the SPE Europec featured at 81st EAGE Conference and Exhibition, London, UK, 3–6 June 2019.
23. 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](#)]
24. Luo, G.; Tian, Y.; Bychyna, M.; Ehlig-Economides, C. Production optimization using machine learning in bakken shale. In Proceedings of the Unconventional Resources Technology Conference, Houston, TX, USA, 23–25 July 2018.
25. He, Q. Smart determination of estimated ultimate recovery in shale gas reservoir. In Proceedings of the SPE Eastern Regional Meeting, Lexington, KY, USA, 4–6 October 2017.
26. Hochreiter, S.; Munchen, T.U. the Vanishing Gradient Problem During Learning. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **1998**, *2*, 107–116. [[CrossRef](#)]
27. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
28. Cho, K.; Van Merriënboer, 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 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, 8 October 2014; pp. 1724–1734.
29. Zhan, C.; Sankaran, S.; LeMoine, V.; Graybill, J.; Mey, D.O.S. Application of machine learning for production forecasting for unconventional resources. In Proceedings of the SPE/AAPG/SEG Unconventional Resources Technology Conference, Denver, CO, USA, 22–24 July 2019.

30. Sun, J.; Ma, X.; Kazi, M. Comparison of decline curve analysis DCA with recursive neural networks RNN for production forecast of multiple wells. In Proceedings of the SPE Western Regional Meeting, Garden Grove, CA, USA, 22–26 April 2018.
31. Song, X.; Liu, Y.; Xue, L.; Wang, J.; Zhang, J.; Wang, J.; Jiang, L.; Cheng, Z. Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model. *J. Pet. Sci. Eng.* **2020**, *186*, 106682. [[CrossRef](#)]
32. Olah, C. Understanding LSTM Networks. Available online: <https://colah.github.io/posts/2015-08-Understanding-LSTMs> (accessed on 5 October 2020).
33. Chollet, F. Keras. Available online: <https://github.com/fchollet/keras> (accessed on 5 October 2020).
34. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res.* **2010**, *9*, 249–256.
35. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
36. Grave, É.; Obozinski, G.; Bach, F. Trace Lasso: A trace norm regularization for correlated designs. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 2187–2195.
37. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Phys. Lett. B* **1993**, *299*, 345–350.
38. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
39. Aulia, A.; Jeong, D.; Saaid, I.M.; Kania, D.; Shuker, M.T.; El-Khatib, N.A. A Random Forests-based sensitivity analysis framework for assisted history matching. *J. Pet. Sci. Eng.* **2019**, *181*, 106237. [[CrossRef](#)]
40. Biau, G.; Scornet, E. A random forest guided tour. *Test* **2016**, *25*, 197–227. [[CrossRef](#)]
41. Ishwaran, H.; Malley, J.D. Synthetic learning machines. *BioData Min.* **2014**, *7*, 1–12. [[CrossRef](#)] [[PubMed](#)]
42. Varoquaux, G.; Buitinck, L.; Louppe, G.; Grisel, O.; Pedregosa, F.; Mueller, A. Scikit-learn. *GetMobile Mob. Comput. Commun.* **2015**, *19*, 29–33. [[CrossRef](#)]
43. Scornet, E.; Biau, G.; Vert, J.P. Consistency of random forests. *Ann. Stat.* **2015**, *43*, 1716–1741. [[CrossRef](#)]
44. Zhang, P.G. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [[CrossRef](#)]
45. Herui, C.U.I.; Xu, P. Summer short-term load forecasting based on ARIMAX model. *Power Syst. Prot. Control* **2015**, *43*, 108–114.
46. Lomonaco, V. Why Continual Learning is the Key Towards Machine Intelligence. Available online: <https://www.oreilly.com/radar/why-continuous-learning-is-key-to-ai/> (accessed on 5 October 2020).
47. Ben, Y.; Perrotte, M.; Ezzatabadipour, M.; Corporation, O.P.; Ali, I. Real time hydraulic fracturing pressure prediction with machine learning. In Proceedings of the SPE Hydraulic Fracturing Technology Conference and Exhibition, the Woodlands, The Woodlands, TX, USA, 4–6 February 2020.

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).