Computers & Geosciences 164 (2022) 105126

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet

Yanrui Ning [a,*], Hossein Kazemi [a], Pejman Tahmasebi [b]

[a] Department of Petroleum Engineering, Colorado School of Mines, USA
[b] Department of Petroleum Engineering, University of Wyoming, USA

ARTICLE INFO

Keywords:
Machine learning
Time series forecasting
Oil production prediction
ARIMA
LSTM

ABSTRACT

It is challenging to predict the production performance of unconventional reservoirs because of the sediment heterogeneity, intricate flow channels, and complex fluid phase behavior. The traditional oil production prediction methods (e.g., decline curve analysis and reservoir simulation modeling forecasting) are subjective. This paper presents a machine learning-based time series forecasting method, which considers the existing data as time series and extracts the salient characteristics of historical data to predict values of a future time sequence. We used time series forecasting because of the historical fluctuations in production well and reservoir operations. Three algorithms were studied and compared to address the limitations of traditional production forecasting: Auto-Regressive Integrated Moving Averages (ARIMA), Long-Short-Term Memory (LSTM) network, and Prophet. This study starts with the representative oil production data from a well located in an unconventional reservoir in the Denver-Julesburg (DJ) Basin. 70% of the data was used for model training, whereas the remaining 30% of data was used to evaluate the performance of the above-mentioned methods. Then, the decline curve analysis and reservoir simulation modeling forecasting were applied for comparison. The advantages of the machine-learning models include a simple workflow, no prior assumption about the reservoir type, fast prediction, and reliable performance prediction for a typical fluctuating declining curve. More importantly, the 'Prophet' model captures production fluctuation caused by winter impact, which can attract the operator's attention and prevent potential failures. This has rarely been explored and discussed by previous studies. The application of ARIMA, LSTM, and Prophet methods to 65 wells in the DJ Basin show that ARIMA and LSTM perform better than Prophet—probably because not all oil production data include seasonal influences. Furthermore, the wells in the nearby pads can be studied using the same parameter values in ARIMA and LSTM for predicting oil prediction in a transferred learning framework. Specifically, we observed that ARIMA is robust in predicting the oil production rate of wells across the DJ Basin.

1. Introduction

Machine learning (ML) methods have recently received much attention in energy resources because of the following reasons. First, the cost-effectiveness of ML methods makes it possible to extract maximum value from massive datasets generated in subsurface reservoir characterization, drilling, completion, production, etc. Second, ML algorithms using the classification of data, linear regression, and more complex neural networks guide operators to explore the role of geologic diversity and operating parameters on enhancing oil recovery (EOR) and subsurface management. Third, some ML models achieve higher prediction accuracy when compared to physical and empirical models as they can discover complex and hidden patterns in data more efficiently (Noshi et al., 2018b). Specifically, time series forecasting is applied in oil production prediction due to the historical fluctuations in production well and reservoir operations (Bai and Tahmasebi, 2021).

The commonly used ML methods in exploration and production fall into supervised learning and unsupervised learning. Supervised learning requires a labeled training dataset where each sample i has one or multiple input features/variables (x_i) and an output variable (y_i). An algorithm then learns the function that maps these input-output pairs and finally it is used to predict the test (unseen) dataset with similar features. Supervised learning is further grouped as 1) Regression problem, including linear regression, Support Vector Machine (SVM)

* Corresponding author.
E-mail address: yning@mines.edu (Y. Ning).

https://doi.org/10.1016/j.cageo.2022.105126
Received 29 November 2021; Received in revised form 11 March 2022; Accepted 2 May 2022
Available online 6 May 2022

regression, Artificial Neural Networks (ANN), etc. and 2) Classification problem, such as logistic regression, SVM classification, Random Forest classification, Gradient Boosting Machines (GBM), etc. For example, Bakshi et al. (2017) combined completion parameters with a novel nonlinear regression model to predict 3-month and 18-month cumulative oil production. Ahmadi (2016) developed an SVM regression to model the environmental effects on drilling fluid rheology performance. You et al. (2020) coupled ANN regression with multi-objective optimizers to empower the decision-making process of the CO_2-EOR projects. Sneed (2017) used a high-performance Random Forest (RF) regression model to predict electrical submersible pump (ESP) lifespan in the Delaware Basin. Kellogg et al. (2018) applied multiple classification models, including Logistic Regression, SVM, Random Forest, and GBM, to predict a producer acid maintenance job's economic level. Unsupervised learning contains only input data (X) and no corresponding output variable (y). The prevalent unsupervised learning problem in the oil industry is K-Means Clustering (Noshi, 2018a). But it has been as widely used as the supervised learning model. Zhao et al. (2017) used the seeded K-Means algorithm to sort remaining oil types over a high water cut period. All such methods and developments are reviewed elsewhere (Tahmasebi et al., 2020). Ruse et al. (2021) used the Gradient Boosting algorithm to predict Young's moduli, Poisson's ratios, and the minimum horizontal stress, which aids in selecting the best interval for hydraulic fracturing.

Traditionally, unconventional oil reservoir performance is predicted using two methods: empirical decline curve analysis (DCA) and science-based reservoir simulation (RS) models. The most common DCA prediction is the Arps method (Arps, 1945) given as (1), where q_t is the production rate at time "t", q_i is the initial production rate, D_i is the initial decline rate, b is the decline exponent, and t is the production time. Arps method can be applied both to conventional and unconventional reservoirs; however, it is not easy to determine the rate exponent "b" which varies throughout the reservoir and changes over time (Gupta et al., 2014). Furthermore, because DCA only uses a few parameters to predict lifelong production, it tends to generate only similar prediction curves for the wells located in the same neighborhood or for the wells with similar geology.

$$q_t = \frac{q_i}{(1 + bD_i t)^{\frac{1}{b}}} \tag{1}$$

Ever since the late-1990s, science-based reservoir simulation (RS) has been used to predict reservoir performance of large fields. Reservoir simulation modeling generates viable predictions when the underlying physical architecture and the rock and fluid flow properties data are reliable, which leads to extensive and detailed reservoir characterization data— a time-demanding endeavor. In unconventional reservoirs, the reservoir permeability is very small (e.g., 0.001 mD) and heterogeneous; thus, in general, each well can act independently of the distant wells. Hence, each well's performance is essentially local to the well based on the well stimulation efficacy and the drainage volume's permeability, porosity, stimulation volume, and hydrocarbon properties. Thus, we believe predicting oil and gas production from unconventional shale reservoirs could fall into the domain of 'time series forecasting' because of the large number of wells in a given lease which is contrary to the conventional reservoirs. Such a large amount of data leads itself to the 'ML domains' which have not been adequately established for this particular forecasting problem. Aside from the above complexities, most of the available methods consider the process of forecasting without considering the effect of seasonality and just as a simple time-series prediction. The problem of forecasting is, however, more complex that just considering a 1D data. In such cases, one needs to include the effect of other factors such as the effect of time and, in a larger scale, seasonality, which is what we have utilized in this study.

It should be note that time series ML, on the other hand, is different from conventional regression-based ML because the time series output is a sequence of time-dependent values rather than independent values without time ordering. In this paper, we examined three time-series forecasting methods to predict the future oil production performance of 65 wells across the Denver-Julesburg Basin (DJ Basin). This study starts with the oil rate matching of a specific well using ML methods. Then the ML results and computing requirements are compared with the conventional reservoir simulation results. Following that, three ML methods are applied to 65 wells on four pads having various stratigraphic columns, formation properties, fault distribution, stress orientation, and fluid properties. Conclusions are drawn at the end.

2. Methodology

Three time-series models were employed to predict the oil production performance of a well in the DJ Basin. Time-series is a series of data points in time order—thus, a sequence of discrete-time data. The models are Auto-Regressive Integrated Moving Averages (ARIMA), Long-Short Term Memory (LSTM) recurrent neural network, and Prophet model. ARIMA is a statistical technique that works well when stationary dataset. A time-series dataset is stationary if the statistical properties such as mean, and variance remain relatively constant on the entire domain. A simple and intuitive example is that the probability of flipping a fair coin is 50% heads up, regardless of whether it is flipped today or tomorrow. LSTM learns from the past behavior of existing data and then predicts the future behavior. It is excellent at learning from long-term dependencies, making it possible to consider initial peak production when forecasting later production. Prophet model forecasts the nonlinear trends with yearly, weekly, daily seasonality, and holiday effects. It also handles outliers and works best when historical data contains periodic seasonal effects. For example, this method can predict monthly airline passenger numbers of an airport, which are strongly affected by holidays and seasons.

2.1. ARIMA

Fig. 1 illustrates the typical flowchart of an ARIMA model. The first step is to check if a sequence of data is stationary. If the original series is stationary, then the integration term "d" equals 0, meaning the ARIMA model has no differencing orders. Otherwise, it needs to transform the data, such as taking differences of the data, until the data becomes stationary. For example, d = 1 suggests the time series is stationarized by first-order differencing. The transformed data series should then pass the white noise test to make sure the data sequence can be predicted. White noise examples include whirring fan and radio static, which are random signal series containing all frequencies at an equal intensity and cannot be predicted. Following that, an ARIMA model will be compiled by selecting values of the remaining two hyperparameters, namely the autoregressive (AR) term "p" and the moving-average (MA) term "q". AR process represents the current value of a time series that can be obtained from previous values of the same time series. Examples of the AR process include stock prices and global temperature (Salvi, 2019). AR term "p" can be defined using the Partial Auto-correlation Function (PACF) plot, which finds a correlation of residuals with its lagged version. The number of non-zero partial autocorrelations gives the order of the AR term "p" (PennState, 2014). MA process indicates the regression error is linearly defined by the error terms that occurred in the past (Box et al., 2015). For instance, the change of crude oil price due to hurricanes can be thought of as a MA process. Autocorrelation Function (ACF) plot is used to identify the MA term "q". ACF plot describes how well the present value of the time series is related to its previous values. The lag beyond which the ACF cuts off, which is the number of non-zero autocorrelations, indicates the order of the MA term "q" (Duke, 2014). Because the automatic information criteria, such as Akaike Information Criterion (AIC), Bayes Information Criterion (BIC), and Hannan–Quinn information criterion (HQIC), balance simplicity and parametric parsimony while fitting the model well, AIC and BIC are usually used to search for better models. The ARIMA (p, d, q) model with a lower AIC or

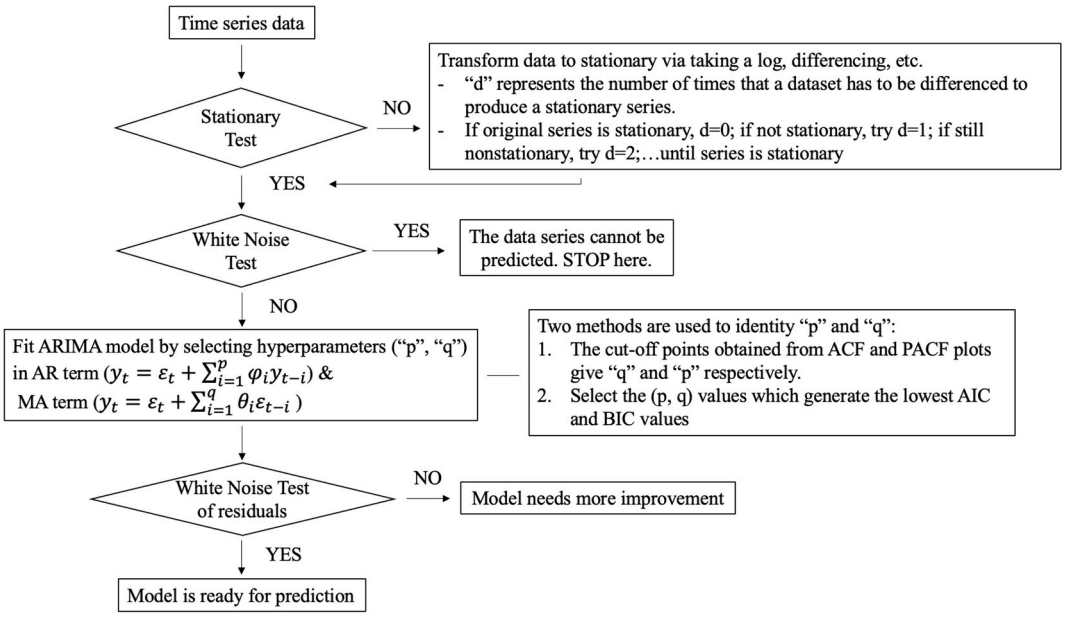


Fig. 1. A typical flowchart of an ARIMA model.

BIC value is a better model (Mohammed et al., 2015). Finally, the model is evaluated by applying a white noise test on residuals, where random residuals indicate the model adequately fits the data and is ready for prediction. Otherwise, if residuals do not look like white noise, a modified model is needed by updating the hyperparameters. The details will be discussed in the following sections.

2.1.1. Stationarity test

Stationarity means that the statistical properties of a process generating a time series do not change over time. ARIMA model is applicable only when a time series is stationary. If the stationary criterion is not met, transforming data is necessary via detrending, differencing, decomposition, etc., with a purpose to stationarize the time series (Srivastava, 2015). Two methods are often used to test stationarity: Rolling Statistics and Augmented Dickey Fuller (ADF) test. Rolling Statistics is a visual test to plot the moving average or moving standard deviation of a particular time period and check whether these statistic values vary with time. ADF test is a statistical test with a null hypothesis that time series is non-stationary. If the test statistic result is less than critical values under different confidence levels, the null hypothesis can be rejected, demonstrating the series is stationary.

Practical time series is often non-stationary. For example, the declining oil production trend represents the mean decreases over time; the extremely low temperature at wintertime may indicate production loss due to mechanical failures or less routine maintenance. Therefore, it is necessary to eliminate trend and seasonality so that the transformed series is stationary for forecasting, and then the forecasted values should be converted to the original scale by applying the trend and seasonality back.

Multiple methods are effective in eliminating trends and seasonality. Transformation such as taking a log or square/cubic root can penalize larger values more than smaller values. Differencing, either first order or higher order differencing, removes the series dependence on time and stabilizes the mean of a time series, which considerably reduces or eliminates trend and seasonality. Taking averages of a certain time period, such as monthly or weekly average, removes noise. In specific applications, it is necessary to take multiple steps of data transformation to obtain stationary time series (Jain, 2016).

2.1.2. White noise test

Before simulating the transformed time-series and making predictions, it is necessary to introduce the concept of 'white noise' for two reasons (Brownlee, 2017). First, white noise means a sequence of randomly independent numbers that cannot be predicted. A time series can be reasonably modeled only when it is not white noise. Second, the forecasted residuals should be white noise if good predictions are made by a forecast model. The Ljung-Box test is a statistical technique for the white noise test. The null hypothesis is that the data series is independently distributed (white noise), whereas the alternative hypothesis is that the data series is not alone distributed and exhibits serial correlation.

2.1.3. AR and MA terms

AR term refers to the lags of the dependent variable. For instance, if a high initial oil rate is observed, we would expect the oil production in the following months to be high as well. For a given zero-mean discrete time series Y_t, the AR process of an order p is written as (2), where ε_t is white noise errors with zero mean and constant variance, φ_i is the parameter of the AR model, y_{t-i} indicates the lags (Hu, 2002; Eshel, 2003). An AR term of 3, AR (p = 3), means the dependent variable y at time point t, y_t, is predicted by three values: y_{t-1}, y_{t-2} and y_{t-3}. There is no autoregression term if p equals 0.

$$y_t = \varepsilon_t + \sum_{i=1}^{p} \varphi_i y_{t-i} \tag{2}$$

MA term is another way to model the variable y_t at time t. It refers to lagged forecast errors in the prediction equation. For instance, a mechanical failure of an oil well negatively affects the oil production not only at the time it takes place but also in the near future. An MA process of an order q is described by (3), where ε_t is white noise error at time t, θ_i is the parameter of the MA model, q is the order of the MA model. An MA term of 1, MA(q = 1), means errors ε_t and ε_{t-1} will be used to predict y_t.

$$y_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{3}$$

2.2. LSTM

A recurrent neural network (RNN) is a densely connected neural network that processes data with periodic patterns. Typically, RNN suffers from vanishing gradient and exploding gradient issues, making it

challenging to learn long-term dependencies (Olah, 2015). To overcome this problem, LSTM introduces memory units/cells into the network.

Like other neural networks, LSTM has an input layer, one or multiple hidden layers, and an output layer. The input layer contains initial data for the neural network, while the output layer produces the results. The intermediate layer is the hidden layer. In LSTM, the hidden layer is a chain of repeating cells of the neural network, as illustrated in Fig. 2, where "A" represents a memory cell at each time step. Within each cell, there are three types of gates: forgot gate, input gate, and output gate. All three gates contain activation functions like sigmoid (σ) or *tanh* function, deciding how much information will be passed down to the sequence chain. First, the forget gate (f_t) decides what information to throw away from the previous cell state C_{t-1}. As listed in (4), forgot gate considers the hidden state from the previous LSTM cell (h_{t-1}) and the input of the current time step (x_t), then the sigmoid activation (σ) squishes values between 0 and 1. Multiply any value by zero, causing this value to be "forgotten". Any number multiplied by one, causing this value to be "kept". Second, the input gate determines what information will flow into and be stored in the cell state (C_t). It includes two parts. The sigmoid layer, (5), decides which value to update, whereas the *tanh* layer, (6), pushes values between -1 and 1 and creates new candidate values $\widetilde{C}_t$ that could be added to the cell state. The next step is to calculate the new cell state C_t using (7), where certain information of the old cell state (C_{t-1}) is dropped and new information is added. The third gate is the output gate, deciding what information will be passed through to the predictor. A sigmoid function, (8), describes how much information will be let through, whereas the following *tanh* function, (9), ensures the values of the hidden state are always in the interval $(-1, 1)$. Due to the three gates within each memory unit, LSTM is able to learn long-time dependencies. Ultimately, the last hidden state obtained from the hidden layer is passed into the output layer for prediction (Zhang et al., 2019). It should be noted that W is weight and b is bias in the following equations through which a ML is constructed mathematically.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t]\right) + b_f \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t]) + b_i \tag{5}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t]) + b_C \tag{6}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{7}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t]) + b_o \tag{8}$$

$$h_t = o_t * tanh(C_t) \tag{9}$$

Fig. 3 illustrates a typical flowchart to construct an LSTM model and make predictions. The first step is data preprocessing. Values in a dataset are converted to the float type and then normalized to accelerate the training. Then the data sequence is spitted into training and test datasets, so that the model is developed based on the training dataset and the accuracy can be verified using the test dataset. Following that, each dataset will be divided into input/output samples using which the LSTM can learn. For each sample, input X are a specified number of previous time steps, whereas output y is used as the following-step prediction being learned by LSTM. One simple example is that the data sequence of [10, 20, 30, 40, 50] is divided into two input/output patterns (x_1, y_1) and (x_2, y_2). The input X is (x_1 = [10, 20, 30], x_2 = [20, 30, 40]), and the corresponding y is (y_1 = [40], y_2 = [50]). Here, three time-steps are used as input while one time-step is used as output for one-step prediction. The second step is to construct an LSTM model, which includes the input layer, hidden layer, and output layer. The LSTM input layer is specified by the three-dimensional 'input shape' argument in the first hidden layer. Hence the training input X should be reshaped to three-dimensional, with the format of (number of samples, number of timesteps, number of features). Here, the number of samples refers to the sample size in the training dataset; the number of timesteps indicates how many steps in each sample are used for prediction; the number of features equals one because there is only one variable (oil rate). The LSTM hidden layer contains parameters such as the number of hidden layers and the number of memory cells in each hidden layer. The last layer is an LSTM dense output layer that makes a prediction. The third step is to fit the LSTM model into the training dataset. A loss

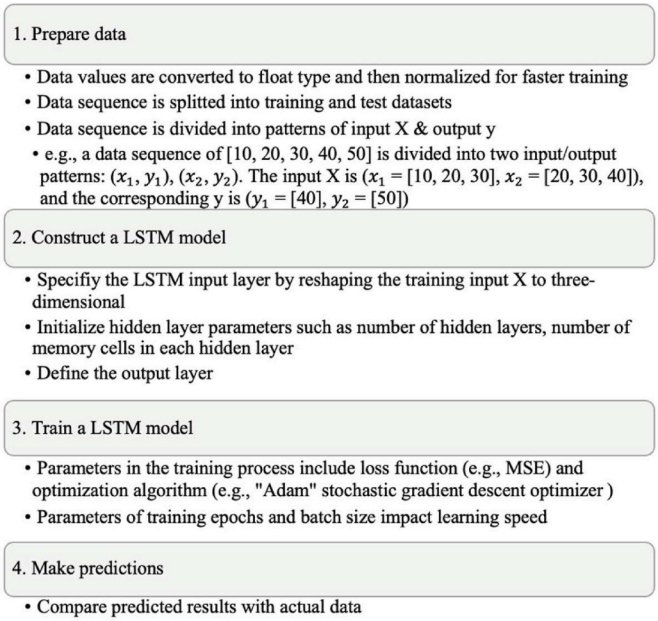


Fig. 3. A typical LSTM model flowchart for time series forecasting.

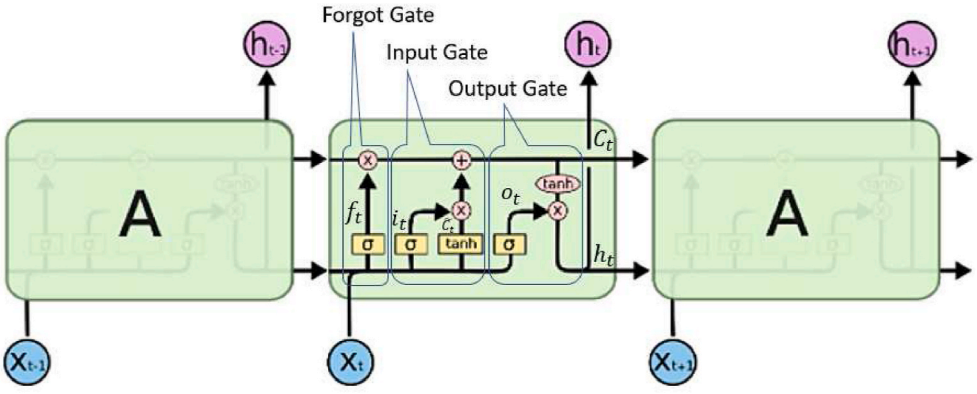


Fig. 2. The hidden layer of an LSTM network contains a chain of memory cells. Each cell has three gates (modified from Olah, 2015).

function, here Mean Squared Error (MSE), and an optimization algorithm such as "Adam" stochastic gradient descent optimizer are used. The number of epochs and batch size are also defined here, which impact how quickly the model learns the training dataset (Brownlee, 2018). Once the model is fit, the final step is to use the model for prediction.

2.3. Prophet model

Prophet model is an open-source automatic forecasting tool developed by Facebook in 2017. It is specialized in modeling nonlinear time series data with yearly/weekly/daily seasonality and holiday effects. It also handles missing data and outliers, which makes the matching process almost automatic and performs better than other approaches (Prophet, 2017; Taylor et al., 2018). The Prophet forecasting model is represented by (10), where $g(t)$ is linear or logistic growth curve of non-periodic changes, $s(t)$ is periodic changes such as yearly seasonality, $h(t)$ is holiday effect, ε_t is an error term. Oil production generally declines over time, meaning that the seasonality term and holiday effect might not be that significant. This will be reflected when modifying hyperparameters to fit the model.

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t \tag{10}$$

2.4. Evaluation criteria

R-squared is not appropriate and adequate for nonlinear regression (Spiess et al., 2010). Therefore, other commonly used evaluation metrics, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), are used to evaluate the matching results. There is no definite answer about which metric of the two is a better indicator of modeling performance, though there have been long discussions (Chai and Draxler, 2014; Willmott and Matsuura, 2005). Both metrics range from 0 to infinity, where lower values represent better models. RMSE penalizes large errors more than minor errors. Hence, outliers or large errors will lead to a bigger RMSE value. MAE measures the average magnitude of errors. MAE does not necessarily increase with increasing error-magnitude variance, whereas RMSE increases steadily (Willmott and Matsuura, 2005). (11) and (12) list the RMSE and MAE of predicted value $\hat{y}_t$ and true value y_t at time t, over a period of T time steps.

$$RMSE = \sqrt{\sum_{t=1}^{T} \frac{(\hat{y}_t - y_t)^2}{T}} \tag{11}$$

$$MAE = \frac{1}{T} \sum_{t=1}^{T} |\hat{y}_t - y_t| \tag{12}$$

3. Oil production prediction

The oil production data are from horizontal Well A (API # 05-123-37149) located in the DJ Basin, Colorado. The production curve is a typical unconventional reservoir decline curve. The monthly production data spans 09/2013 to 09/2020. 70% of the data, consisting of 60 data points over the first six years, was used as the ***training dataset*** while the production data from the remaining two years were used for ***prediction***. In the actual production data series, none of the 85 months has empty or zero production. Hence the oil production dataset with lots of noisy data is kept as it is. Nevertheless, two months in the test dataset, 04/2019 and 05/2019, are shut off for the majority of the time. Each of them only has two days and one day on, respectively.

3.1. ARIMA

The procedure of building an ARIMA model for time series analysis includes three steps: 1) stationarize the series, 2) find optimal hyperparameters to build the ARIMA model, and 3) evaluate the model and make predictions.

3.1.1. Stationarize the time series

Oil production decreases over time, meaning that the time series is not stationary. The dataset is transformed by applying a log function, which is followed by first-order differencing. The original oil production curve and the transformed curve are shown in Fig. 4.

To test the stationary level of the transformed time series, rolling statistics and ADF method are applied. The rolling statistics plotted in Fig. 5 indicates that the rolling mean (black dash curve) is very close to 0. The mean keeps constant over time, supporting the transformed series is stationary. Additionally, according to ADF test results, the p-value of $1.03e^{-9}$ is lower than the significance level of 0.05, which indicates the alternative hypothesis of stationary time series should be accepted. Moreover, the ADF test statistic value of −6.94 is lower than the critical value at 1% significance level, which is −3.51. This also demonstrates that the transformed time series is stationary.

3.1.2. White noise test

To detect whether the transformed time series is white noise, Ljung-Box Statistics is applied to the transformed time series. The p-value of 0.003 is less than the significance level of 0.05, indicating the null hypothesis of white noise should be rejected. Hence, the transformed time series is not white noise, meaning it can be modeled and predicted.

3.1.3. Hyperparameters selection and model creation

As discussed earlier, three hyperparameters are specified in an ARIMA model: p, q, and d. When transforming the time series to a stationary dataset, the order of differencing is 1, implying the hyperparameter d equals 1. The hyperparameters p and q are initially determined using PACF and ACF plots specifically, as shown in Fig. 6. The blue bands are significance confidence bands, within which the function value is regarded as 0. The ACF plot has two non-zero values, cutting off after 2nd lag, meaning it is mostly a MA (q = 2) process. The PACF plot has a cut-off after the 3rd lag, suggesting an AR (p = 3) process (Bogacka, 2007; Jain, 2016). Moreover, the results generated from AIC, BIC, and HQIC statistical methods imply the best values of (p, q) are (2, 2), (0, 1), and (0, 3). After a few iterations, the (p, d, q) values of (0, 1, 1) comes out to be the combination with the least AIC and BIC values. Therefore, values of (0, 1, 1) are used as the orders of (p, d, q) in the final ARIMA model.

3.1.4. ARIMA model evaluation and prediction

In Fig. 7, the solid gray curve represents simulated training data, which is mostly overlapped with the actual production (black dots), indicating the model is representative. Furthermore, the Ljung-Box test is applied to the residuals of the ARIMA model to determine whether residuals are random. The Ljung-Box statistics show that the p-value is larger than 0.05, meaning residuals are white noise and the ARIMA model provides an adequate fit to the data.

Then this model is used for prediction for two more years. The simulated results are presented by the solid blue curve in Fig. 7. The predicted test data generates a RMSE of 4.97 and a MAE of 4.37. When digging into the blue curve, it is found that the first half period, ranging from 09/2018 to 08/2019, is better matched when compared to the second half. This is further demonstrated by the lower RMSE value (3.63) and lower MAE value (2.87) of the first one-year period.

3.2. LSTM

Considering LSTM is sensitive to the scale of input data, oil production data is normalized into the range of 0–1 during training, and then the predicted values are rescaled back. After multiple iterations, the following LSTM model is used to fit the oil production series: one input layer, one hidden layer with four memory cells, along with one dense

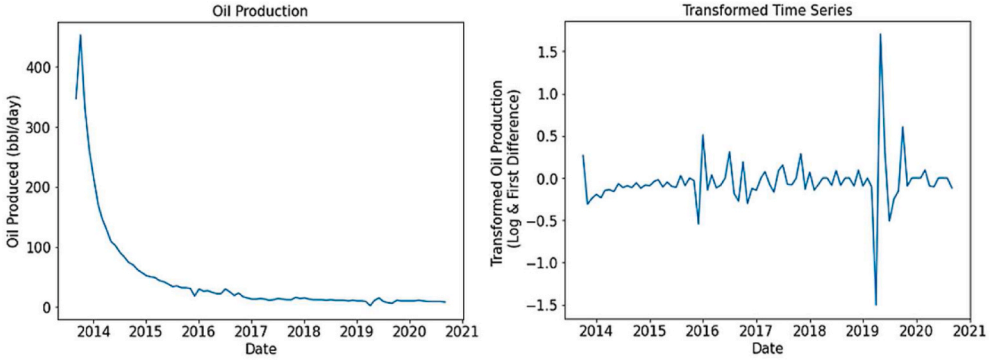


Fig. 4. Oil production decline curve (left) and the transformed time series (right).

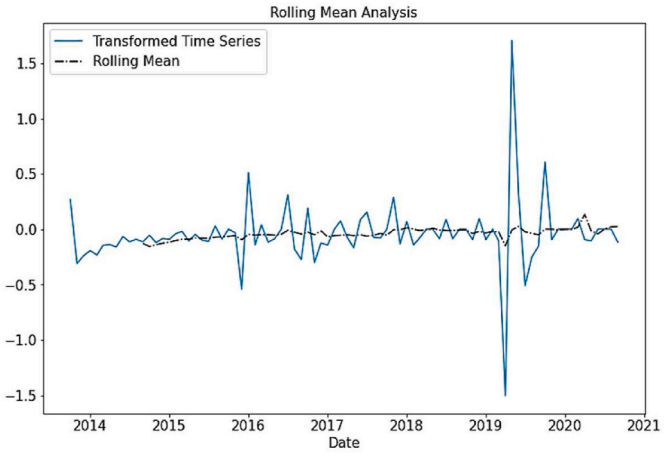


Fig. 5. Rolling mean analysis of the transformed time series.

output layer that makes a prediction. In each memory cell of the hidden layer, the default activation function is used. The model is compiled using the MSE loss function and the "Adam" optimization algorithm. When training, two previous time steps are used as input variables to predict the following time step. The model is trained for 100 epochs with a batch size of 1. Once the model is fit, its performance is estimated on the training dataset. In Fig. 8, the simulated training dataset (solid gray curve).overlaps with the black dots of actual oil production, indicating the model is well-matched and representative. Then this model is ready for prediction, resulting in a solid blue curve as shown in Fig. 8, which has an RMSE value of 2.96 and an MAE value of 2.19. Because two months, 04/2019 and 05/2019, only have two days and one day on respectively, the production rate of the two months does not follow the previous decline trend. The solid blue curve of Fig. 8 shows LSTM model is able to capture this unusual production change due to being exposed to a similar pattern in the training phase (around 2016).

3.3. Prophet model

Aside from the default values for hyperparameters, such as growth trend and auto yearly seasonality, two more hyperparameters are tuned in the Prophet model. First, the hyperparameter "changepoint_prior_scale" determines trend flexibility. The default value is 0.05. Decreasing this value will result in the trend underfit a time series while increasing the value will make the trend more flexible to overfit. Second, the hyperparameter "seasonality_prior_scale" controls the flexibility of the seasonality. The default value is 10. A smaller number shrinks the level of seasonality, whereas a larger value allows the seasonality to fit large fluctuations.

Cross-validation is applied to tune the two hyperparameters, which are evaluated on the RMSE average over a 365-day horizon period. The values of 1 for "changepoint_prior_scale" and 0.001 for "seasonality_prior_scale" generate the lowest RMSE. Hence the two values are used in the final Prophet model. The solid blue curve in Fig. 9 presents the predicted production, which has an RMSE of 2.32 and an MAE of 1.34.

Fig. 10 compares prediction results at different seasonality levels ("seasonality_prior_scale" = 0.001 or 0.002). While the simulation curve (solid blue line) on the left plot of Fig. 10 is smooth and follows the declining trend well, the right plot of Fig. 10 shows the simulated production fluctuates around 09/2018 and 09/2019 with the gap of one year. If we look into the right plot, it is observed that the actual production (black dots) around 09/2018 declines slowly, but the model aggressively predicts there is wild production fluctuation. Nevertheless, this model correctly captures the production change around 09/2019, suggesting seasonality existence. That is to say, the Prophet model of the right plot also predicts the actual production data well, though it has a slightly higher RMSE value (2.63) and MAE value (1.75). This suggests

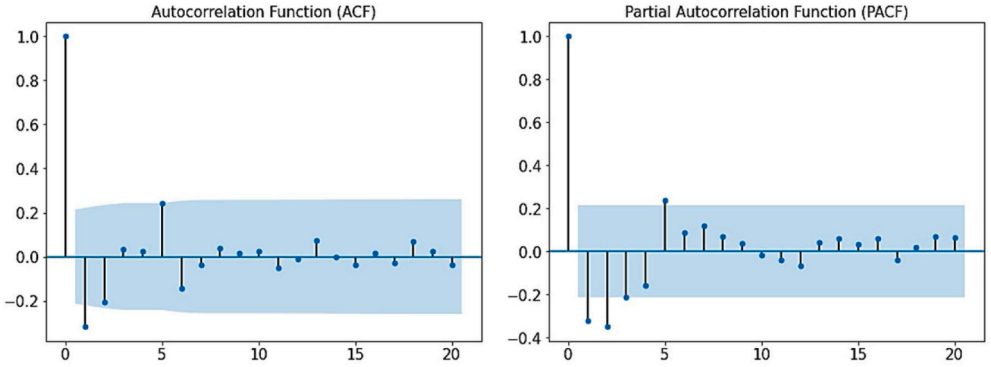


Fig. 6. Selection of hyperparameters p and q using PACF and ACF plots.

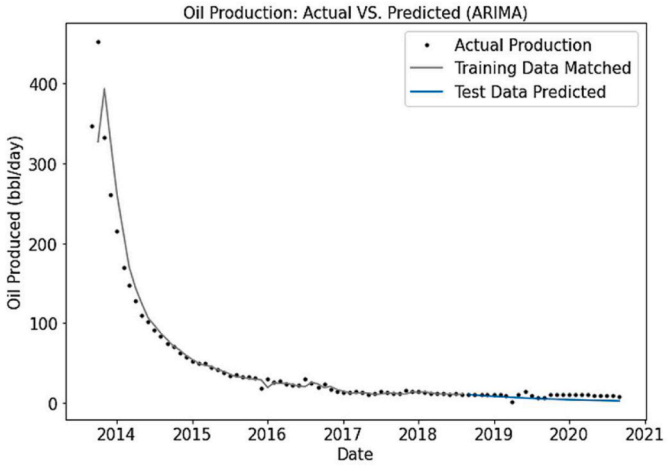


Fig. 7. Comparison of actual oil production (black dots), ARIMA simulated training data (solid gray curve), and ARIMA predicted test results (solid blue curve). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

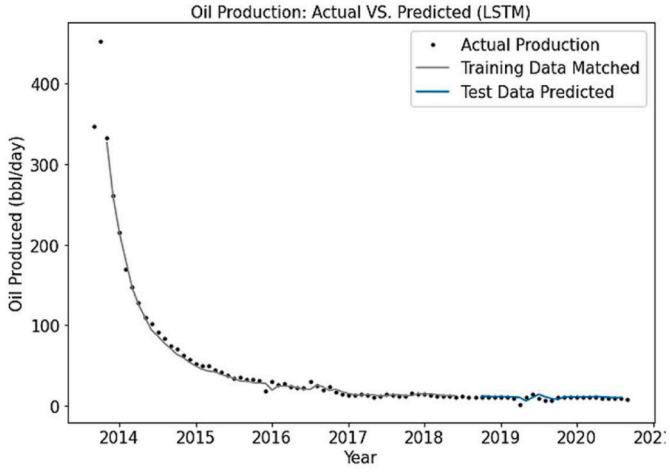


Fig. 8. Comparison of actual production (black dots) and LSTM simulated training (solid gray curve) and test (solid blue curve) results. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

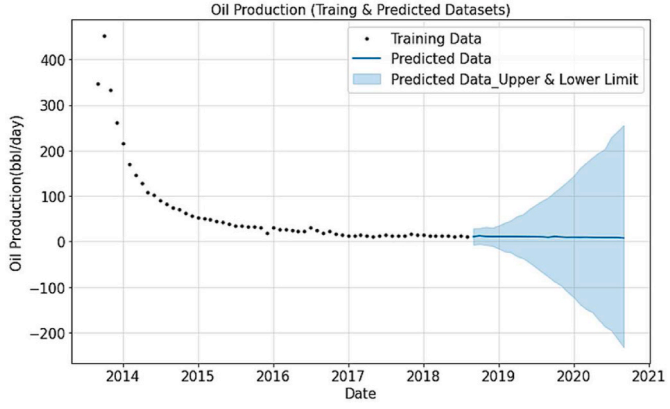


Fig. 9. The solid blue curve of Prophet modeled results follows the previous declining trend. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

that when evaluating modeling performance, it is a good idea to consider RMSE and MAE metrics along with other criteria, such as the general trend and unusual fluctuation.

To explain the simulated seasonal fluctuations on the solid blue curves of Fig. 10, the yearly trend of the training data is illustrated in Fig. 11. It indicates the production from January to August is stable while the production starting from September fluctuates wildly. When looking back to analyze the training data, it is found that the oil production around 09/2016 and 09/2017 shifts and does not follow the previous decline trend well. This could be attributed to a sudden temperature decrease in Colorado's early winter, which may adversely impact routine operations of facilities or equipment and thus cause production loss. That is to say, the Prophet model is good at predicting the general declining trend as well as capturing seasonal impact.

3.4. Arps Method

The Arps Method is also used to simulate oil production. As mentioned in (1), three parameters are included: the initial production rate (q_i), the initial decline rate (D_i), and the decline component (b). After multiple testing of the three parameters, the oil production predicted by Arps method (solid blue curve) is presented in Fig. 12. When matching, the first month that only has 12 days of production is ignored, as such the initial production rate equals the production rate of the second month, which is 453 bbl/day. The blue curve in Fig. 12 has an initial decline rate (D_i) of 0.014 and a decline component (b) of 0.9. The predicted data of the last two years has an RMSE of 3.24 and an MAE of 2.47.

3.5. Reservoir simulation method

In the reservoir simulation study in DJ Basin, a complicated dual-porosity compositional model was built using the data of geology, seismic survey, well logs and core experiments (Ning, 2017). It took about one year to collect and tune all the data to build the model and make the model representative. When using a 20-processor workstation, this model took about 12 h to run. Two and half years of daily oil production data (black dots in Fig. 13) was input into the model, while the blue solid curve represents the simulated oil results. Calculated bottom hole pressure was input into the model as the constraint during the history matching process. Overall, the simulated oil production overlaps with the actual production trend, though the first quarter's production data is mismatched, meaning either the inputted initial bottom hole pressure calculation is not accurate enough, or the input parameters related with the hydraulically stimulated fractures may need more improvement. The modeling simulation results of the last two years show an RMSE of 4.03 and an MAE of 3.26.

4. Comparison of various methods

We used time series forecasting because of the historical fluctuations in producing well and reservoir operations. When compared with traditional oil production prediction methods (Arps method and reservoir simulation modeling), the three time-series forecasting methods (ARIMA, LSTM, and Prophet) have the following benefits:

1) When applying the three time-series methods for forecasting, in-depth knowledge of reservoir and production mechanisms is not necessary. Time series honors the historical production decline trend and noisy characteristics that specifically belong to the well being evaluated. For example, the LSTM model captures the unusual production change in the year 2019 (Fig. 8). The Prophet model is able to predict the production fluctuation caused by inherent seasonality (Fig. 10). However, the Arps method tends to predict a smooth decline trend while the reservoir simulation method requires high accuracy of input parameters.

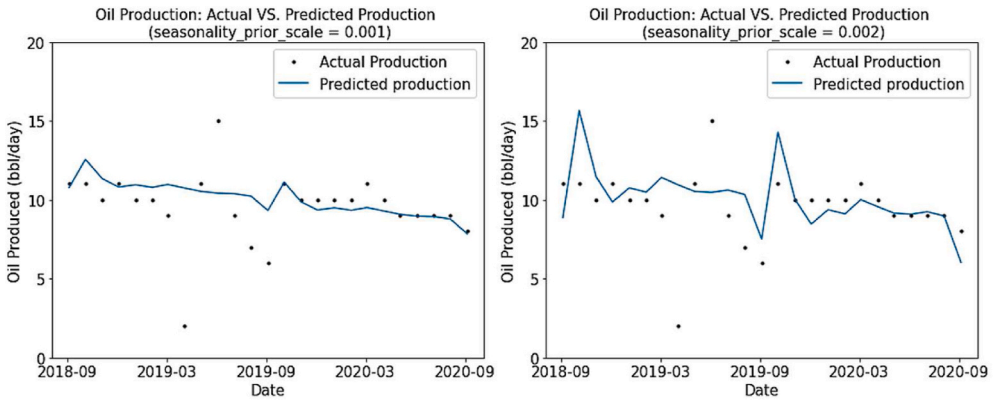


Fig. 10. Oil production prediction at different seasonality levels. The Prophet model of the right plot, which has a stronger seasonality level, follows the general decline trend and correctly captures the wild fluctuations around 09/2019.

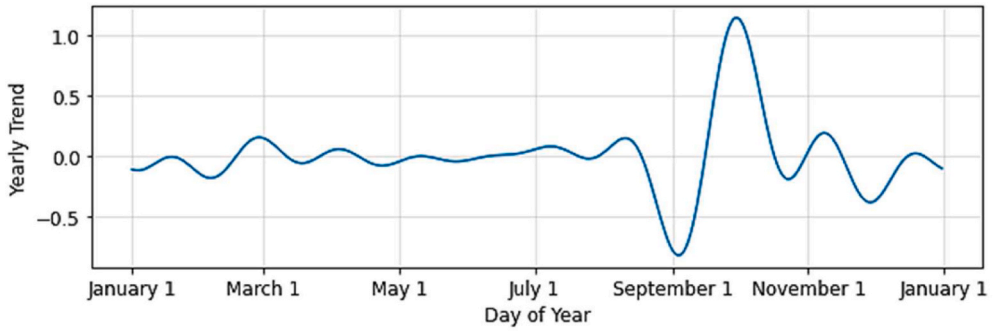


Fig. 11. Yearly trend generated by Prophet model indicates wild fluctuations at early wintertime.

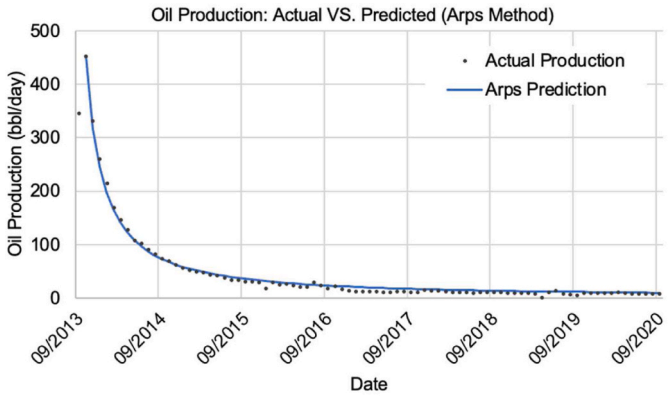


Fig. 12. Comparison of actual production (black dots) and simulated production (solid blue curve) using Arps Method. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

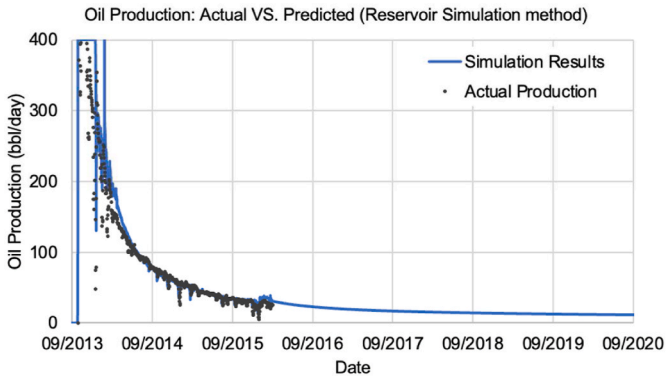


Fig. 13. Daily oil production (black dots) is used as the input of a reservoir simulation model. Simulated results from a reservoir model are shown as the blue solid curve. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

2) It is rather fast to train the ML models and make predictions. The predicted results follow the existing declining trend well and are overlapped with the training data, demonstrating these models are representative.
3) ARIMA model is simple to implement. A typical oil production declining curve is easily transformed to a stationary series by applying the log function and first-order difference. The value 1 for the integration term "d" is good for oil production prediction. The remaining two hyperparameters, autoregression order "p" and moving average order "q", can be calculated automatically using multiple information criteria.
4) LSTM model has no pre-requisites. It is able to simulate the nonlinear decline trend without preprocessing the original dataset.
5) The Prophet model is convenient because of its fixed structure. Only two hyperparameters need more adjustment when predicting oil production. In addition to the parameter values that are estimated using cross-validation, other nearby values may also generate good prediction results (Fig. 10). More importantly, the Prophet model is able to discover the implicit seasonal effect that occurred previously.

Despite the above advantages, each model has its own limitations. First, ARIMA needs prior transformation to stationary time series before modeling. When compared with the 2-year production results, the 1-

year production results predicted by the ARIMA model are better matched with the actual production, which indicates the ARIMA model is more excellent at making short-term predictions on oil production. Next, the LSTM model is a nonlinear RNN model, suggesting it might not be straightforward to choose the most appropriate neural network architecture during the model creation process and requires considering a lot of alternatives. Furthermore, the Prophet model may correctly predict or may exaggerate the seasonal effect, as illustrated in Fig. 10.

According to the 2-year period forecasting RMSE and MAE results listed in Table 1, the Prophet model stands out, followed by the LSTM model. This is probably because the oil production rate does not decline smoothly and has inherent seasonality. Both Prophet and LSTM models are excellent at predicting the declining trend. More importantly, the Prophet model is able to capture the production fluctuation that occurred around September, while the LSTM model also predicts the unusual production change. The presence of noise data further demonstrates that these models are robust enough to capture the general declining trend, as well as predict the seasonal impact.

5. Multiple well investigation in DJ Basin

Considering unconventional reservoirs are heterogeneous, three ML methods (ARIMA, LSTM, and Prophet) were applied to the 65 wells in four pads (A, B, C, D) in DJ Basin (Fig. 14). Four pads are in areas with various stratigraphy, formation properties, fault distribution, stress orientation, and fluid properties, which allows our proposed ML models to have access to a more diverse data while it can also make the training and testing challenging. Pad A and Pad D are in the South and North border townships of the currently producing area of the DJ Basin. All 65 wells have different trajectories, production history, and completion strategies. For each well, 70% of the available monthly production data were used as the training dataset, whereas the remaining data were regarded the test dataset. The same parameter values in Table 1 were applied to the prediction mode of oil production for the 65 wells. The goal was to evaluate the viability of the three time series approaches.

Fig. 15 presents the box plots of RMSE of the results for the test datasets when the three ML methods were applied to predict the oil production of the 65 wells. For all four pads, ARIMA (blue box) and LSTM (orange box) perform better than Prophet (green box). It is probably because Prophet is good at modeling time-series data with seasonality, but not all oil production curves show seasonality effect.

Table 1
Hyperparameters, RMSE, and MAE results of each model.

	Parameters	RMSE of 2-year period forecast	MAE of 2-year period forecast
ARIMA model	p = 0 d = 1 q = 1	4.97	4.37
LSTM model	number of hidden layers = 1 number of memory cells in the hidden layer = 4 epochs = 100 batch_size = 1	2.96	2.19
Prophet model	changepoint_prior_scale = 1 seasonality_prior_scale = 0.001	2.32	1.34
	changepoint_prior_scale = 1 seasonality_prior_scale = 0.002	2.63	1.75
Arps method	q_i = 453 bbl/day D_i = 0.014 b = 0.9	3.24	2.47
Reservoir simulation method	Porosity = 6%–10% Matrix permeability $\approx 10^{-5}$ mD Fracture permeability $\approx 10^{-3}$ mD	4.03	3.26

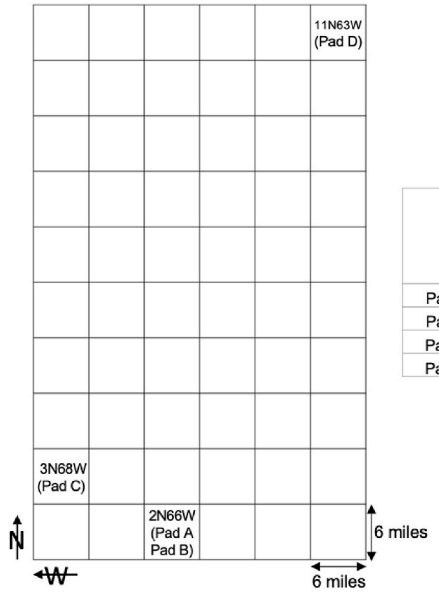
The median RMSE values of the ARIMA and LSTM methods for all four pads are in the range of 3–10, indicating that ARIMA and LSTM methods can predict the oil production trend with different errors. The median RMSE values of Pads A-B are smaller than those of Pads C-D, which might be caused by two reasons: 1) wells in Pads A-B have longer production history, hence more data is available for training the corresponding models; 2) Pads A-B are 14–56 miles away from Pads C-D. It should be noted that the parameter values were initially obtained from the investigation of a well in Pad A and then used for modeling the production of all wells. It demonstrates that the same parameter values can be applied to predict the oil rate of wells in nearby pads with smaller errors, which indicates the generality ability of the proposed network. Further, for Pads A-B, the RMSE values from LSTM method are 1–2 lower than those of ARIMA method, implying LSTM is a little bit more representative of the oil decline curve in Pads A-B. However, the low median RMSE values show that ARIMA performs better than LSTM on Pad D. This is probably due to the values of the hyperparameters in ARIMA on Pad A are appropriate for the wells on Pad D, though the two pads are 56 miles apart. It may indicate that the ARIMA model with the parameter values in Table 1 is robust in predicting the oil production of wells in DJ Basin.

The LSTM model has been proved to be effective in predicting the oil production of wells on Pad B. However, the average R-squared value for the LSTM models is −2.05, which does not show an acceptable performance. This further demonstrates that R-squared is not an appropriate metric to evaluate non-linear regression models since it cannot capture all the complex relationships and, as mentioned earlier, one must include other statistical measures along each other.

6. Conclusions

This study used three time-series forecasting ML methods to predict a typical well's oil decline curve in an unconventional shale reservoir. Specifically, we used time-series forecasting because of the historical fluctuations in production well and reservoir operations. The following conclusions are resulted from this research:

1) This study takes the mystic out of ML for reservoir forecasting and the ML builds a solid basis for the time-series forecasting. When compared with the traditional Arps and reservoir simulation methods, ML methods are more convenient because three is no need to consider reservoir type, reservoir properties, and scientific details of the production mechanisms.
2) The three ML methods presented in this paper (ARIMA, LSTM, and Prophet) have simple workflows, are easy to learn, and predict reliable results for a typical fluctuating while declining production trends, in unconventional shale reservoirs. They were able to predict the declining oil production trend with different error magnitudes. More specifically, both LSTM and Prophet models can predict the unusual production changes, which cannot be achieved by traditional DCA and reservoir simulation methods.
3) When using the ARIMA method, the oil decline curves in shale reservoirs can be easily transformed into stationary time series via a log function followed by first-order differencing. ARIMA is more appropriate for predicting shorter-time oil production, i.e., next one-year period.
4) R-squared is not appropriate in evaluating nonlinear regression. RMSE and the MAE were used to measure time series forecasting performance. The three models generate the error values ranging from 2 to 5, indicating they are representative on predicting the oil rate of Well A. Nevertheless, it is better to consider RMSE and MAE metrics with other criteria such as the general trend and data fluctuations.
5) The Prophet model is able to reveal potential seasonal effects that would be helpful to the operators in advance to prevent possible



	Township-Range	Section	Distance to Pad A (miles)	Number of wells in each pad	Well trajectory direction	Well lateral length (miles)	Train size (number of monthly data used for training)
Pad A	2N66W	24	0	11	N-S	1	60
Pad B	2N66W	23	1	13	N-S	1	58
Pad C	3N68W	1	14	28	E-W	2	20
Pad D	11N63W	17	56	13	N-S	2	21

Fig. 14. The production data of 65 wells in four pads (A, B, C, D) were analyzed for prediction using the time series ML models.

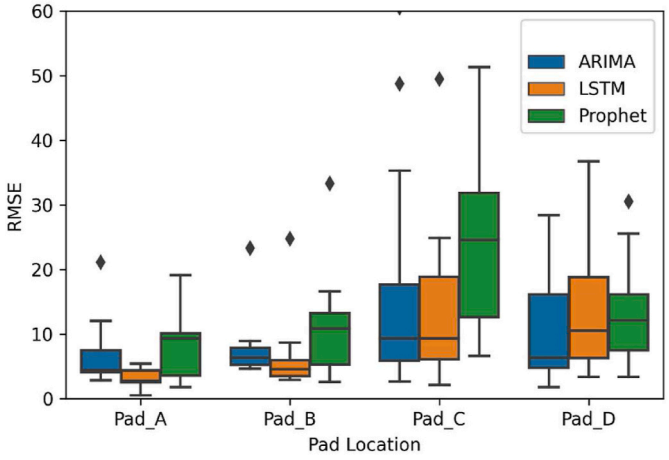


Fig. 15. Box plots of RMSE results when the ML methods were applied to the 65 wells in the four pads across the DJ Basin.

temperature shocks that had happened previously and may happen again.

6) When the three ML methods were extended to predict the oil production of 65 wells across the unconventional reservoirs, ARIMA and LSTM perform better than Prophet, implying that not all oil production curves have seasonality effect in the DJ Basin. Moreover, same parameter values in the ARIMA and LSTM methods can be applied to predict the oil rate of wells in nearby pads with small errors, which can be helpful in terms of less training and also the extendibility of the ML for dealing with new scenarios when they are not seen previously. For the two pads located in the North and South edge of the DJ Basin, ARIMA method with the same parameter values is robust in predicting the oil rate of wells across the DJ Basin.

Computing code and data availability

Contact: yning@mines.edu.

Hardware requirements: x86 64-bit CPU (Intel/AMD architecture), 16 GB RAM.

Program language: Python 3.

Software required: Google Colab

Program size: 3 Mb.

All the data and the source codes used in this study are available for downloading at the author's Github page.

The monthly oil production data is also available from the COGCC (Colorado Oil and Gas Conservation Commission) GIS online website manually. It can also be downloaded from the IHS Enerdeq Browser with active license.

Authorship contribution statement

Yanrui Ning: development of ideas, data acquisition, interpretation, and draft writing, Hossein Kazemi: conception, manuscript revision, Pejman Tahmasebi: critical revision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Nomenclature

q_i	initial production rate
b	decline exponent
p	autoregressive term
d	integration term
φ_i	parameter of an AR model
θ_i	parameter of an MA model

σ	sigmoid activation function
h_t	hidden state at time t
x_t	input at time t
$\widetilde{C}_t$	candidate value at time t
o_t	output gate at time t
y_t	true value at time t
D_i	initial decline rate
t	production time
q	moving-average term
ε_t	white noise error at time t
y_t	AR or MA value at time t
f_t	forgot gate at time t
W	weights
b	bias
i_t	input gate at time t
C_t	cell state at time t
$\widehat{y}_t$	predicted value at time t
X	input

References

Ahmadi, M.A., 2016. Toward reliable model for prediction drilling fluid density at wellbore conditions: a LSSVM model. Neurocomputing 211, 143–149.

Arps, J.J., 1945. Analysis of decline curves. In: Transactions of the AIME, 160, pp. 228–247, 01.

Bai, T., Tahmasebi, P., 2021. Efficient and data-driven prediction of water breakthrough in subsurface systems using deep long short-term memory machine learning. Comput. Geosci. 25 (1), 285–297.

Bakshi, A., Uniacke, E., Korjani, M., Ershaghi, I., 2017. A novel adaptive non-linear regression method to predict shale oil well performance based on well completions and fracturing data. April. In: SPE Western Regional Meeting. Society of Petroleum Engineers.

Bogacka, B., 2007. ACF and PACF of ARMA p,q). Retrieved from. http://www.maths.qmul.ac.uk/~bb/TS_Chapter6_2.pdf.

Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time Series Analysis: Forecasting and Control. John Wiley & Sons.

Brownlee, J., 2017. White Noise Time Series with Python. Retrieved from. https://machinelearningmastery.com/white-noise-time-series-python/#:~:text=What%20is%20a%20White%20Noise,other%20values%20in%20the%20series.

Brownlee, J., 2018. How to Develop LSTM Models for Time Series Forecasting. Retrieved from. https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/.

Chai, T., Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)? Geosci. Model Dev. Discuss. (GMDD) 7 (1), 1525–1534.

Duke, 2014. Retrieved from. http://people.duke.edu/~rnau/arimrule.htm.

Eshel, G., 2003. The yule walker equations for the AR coefficients. In: Internet resource, 2, pp. 68–73. http://www-stat.wharton.upenn.edu/~steele/Courses/956/Resource/YWSourceFiles/YW-Eshel.pdf.

Gupta, S., Fuehrer, F., Jeyachandra, B.C., 2014. Production forecasting in unconventional resources using data mining and time series analysis. September. In: SPE/CSUR Unconventional Resources Conference–Canada. Society of Petroleum Engineers.

Hu, L., 2002. ARMA model. Retrieved from. https://www.asc.ohio-state.edu/de-jong.8/note2.pdf.

Jain, A., 2016. A Comprehensive Beginner's Guide to Create a Time Series Forecast. Retrieved from. https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/.

Kellogg, R.P., Chessum, W., Kwong, R., 2018. Machine learning application for wellbore damage removal in the wilmington field. In: SPE Western Regional Meeting. Society of Petroleum Engineers. April.

Mohammed, A.A., Naugler, C., Far, B.H., 2015. Emerging Business Intelligence Framework for a Clinical Laboratory through Big Data Analytics. Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology: Algorithms and Software Tools. Elsevier/Morgan Kaufmann, New York, pp. 577–602.

Ning, Y., 2017. Production Potential of Niobrara and Codell: Integrating Reservoir Simulation with 4D Seismic and Microseismic Interpretation. Doctoral dissertation, Colorado School of Mines.

Noshi, C.I., Schubert, J.J., 2018a. The role of machine learning in drilling operations; a review. In: SPE/AAPG Eastern Regional Meeting. Society of Petroleum Engineers. October.

Noshi, C.I., Assem, A.I., Schubert, J.J., 2018b. The role of big data analytics in exploration and production: a review of benefits and applications. In: SPE International Heavy Oil Conference and Exhibition. Society of Petroleum Engineers. December.

Olah, C., 2015. Understanding LSTM networks. Retrieved from. https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

PennState, 2014. Retrieved from. https://online.stat.psu.edu/stat510/lesson/2/2.2.

Prophet, 2017. Retrieved from. https://facebook.github.io/prophet/docs/quick_start.html.

Ruse, C.M., Ahmadov, J., Liu, N., Mokhtari, M., 2021. An integrated analytics and machine learning solution for predicting the anisotropic static geomechanical properties of the Tuscaloosa marine shale. In: SPE/AAPG/SEG Unconventional Resources Technology Conference. OnePetro. July.

Salvi, J., 2019. Significance of ACF and PACF plots in time series analysis. In: Towards Daa Science, 27.

Sneed, J., 2017. Predicting ESP lifespan with machine learning. In: Unconventional Resources Technology Conference, Austin, Texas, 24-26 July 2017. Society of Exploration Geophysicists, American Association of Petroleum Geologists, Society of Petroleum Engineers, pp. 863–869. September.

Spiess, A.N., Neumeyer, N., 2010. An evaluation of R 2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach. BMC Pharmacol. 10 (1), 6.

Srivastava, T., 2015. A complete tutorial on time series modeling in R. Analytics Vid. Retrieved from. https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/.

Tahmasebi, P., Kamrava, S., Bai, T., Sahimi, M., 2020. Machine learning in geo-and environmental sciences: from small to large scale. Adv. Water Resour. 142, 103619.

Taylor, S.J., Letham, B., 2018. Forecasting at scale. Am. Statistician 72 (1), 37–45.

Willmott, C.J., Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Clim. Res. 30 (1), 79–82.

You, J., Ampomah, W., Sun, Q., 2020. Development and application of a machine learning based multi-objective optimization workflow for CO2-EOR projects. Fuel 264, 116758.

Zhang, A., Lipton, Z.C., Li, M., Smola, A.J., 2019. Dive into deep learning. Unpublished Draft. Retrieved, 19 Retrieved from,2019. https://d2l.ai/chapter_recurrent-modern/lstm.html.

Zhao, Y., Jiang, H., Li, J., Wang, C., Gao, Y., Yu, F., Su, H., 2017. Study on the classification and formation mechanism of microscopic remaining oil in high water cut stage based on machine learning. In: Abu Dhabi International Petroleum Exhibition & Conference. Society of Petroleum Engineers. November.