Using Machine Learning Methods for Oil Recovery Prediction

B. Daribayev[1], D. Akhmed-Zaki[2], T. Imankulov[1], Y. Nurakhov[1], Y. Kenzhebek[1*]

[1] Al-Farabi Kazakh National University; [2] University of International Business

Summary

In recent years, machine learning methods have been widely used in various fields of science for big data processing. The application of machine learning in the oil industry is also actively expanding. To solve oil recovery problems, it is necessary to use geological models of reservoir fields. With increasing of the reservoir model complexity (size), the computing time also increases. Therefore, it takes longer to predict oil recovery. There are two approaches to solve this problem. The first approach is to develop an effective parallel algorithm taking into account the heterogeneity of computing systems. Many scientists from all over the world are developing parallel algorithms in this field. In particular, we have written many scientific papers. The disadvantage of using this approach is that when you change the initial data for the oil recovery prediction, you need to make calculations on supercomputers every time, which takes a lot of time and resources. The second approach is to use machine learning methods, which is the purpose of this paper.

This paper discusses approaches to using effective machine learning methods for oil recovery prediction. To train the system, we used historical data from the oil field and synthetic data obtained from surrogate models based on two wells (injection and production). Synthetic data were generated based on mathematical models (oil displacement models, enhanced oil recovery models) by varying the different geological parameters. This problem belongs to the "supervised learning" - type of machine learning. Supervised learning requires a complete set of marked data for training the model at all stages of its construction. When implementing the algorithm, we considered machine learning methods for solving the regression and classification problems.

As a result, it was discovered that compared to traditional computational experiments on a regular grid, calculations using machine learning methods are more productive.

Introduction

There is a lot of research related to increasing oil production using machine learning methods. In this work (Krasnov et al., 2018), the authors found out that the use of machine learning (ML) algorithms may turn out to be more productive in comparison with traditional calculations on a regular grid (Akhmed-Zaki et al., 2012; Danaev et al., 2015; Akhmed-Zaki et al., 2016; Imankulov et al., 2016). In (Guo et al., 2018; Sheheta et al., 2012), an approach to creating a proxy model based on machine learning methods was described, in particular, the random forest method was used (Breiman, 2001).

The work (Aliyuda and Howell, 2019) considers machine learning algorithms for estimating the oil production coefficient using a combination of engineering and stratigraphic parameters. For a data set consisting of 30 parameters, linear regression models and the support vector machine (SVM) method were applied. As a result, the data obtained were very close to the results of cross-validation. Thus, the authors of this work suggest that the methods considered by them can be used to predict future production.

The authors of (Mirzaei-Paiaman and Salavati, 2012; Jreou, 2012) considered the use of artificial neural network (ANN) for predicting oil production. The developed model (ANN) in this work (Mirzaei-Paiaman and Salavati, 2012) predicts oil production using three parameters. In addition, the accuracy of the model was compared with some popular correlations, therefore, the authors claim that the developed model is in excellent agreement with the actual measurement data. And in (Jreou, 2012), as an input layer of a neural network, 143 data sets of 6 parameters were used. Thus, using an artificial neural network such as feed forward neural network (FFNN), the authors obtained encouraging results for one of the considered oil fields in their study.

The study (Ristanto and Horne, 2018) examined various machine learning methods for predicting downhole pressure, oil production, and forecasting water cut in production tasks. The data set in this study was obtained using an ECLIPSE reservoir simulator. In this study, the authors applied ten different machine learning methods, and the effects of multiphase flow and data noise were also taken into account. Ridge regression and the support vector method showed the best results at any noise levels in their study.

There are several works related to the application of machine learning methods for processing data from permanent downhole gauges (PDG) (Horne, 2007). PDG data is very often noisy due to operational changes occurring in the well. It is believed that modern PDG can record data every second, so after a few months of work, very large data is created that is difficult to process. Therefore, the authors wanted to develop a reliable method for processing data from permanent downhole sensors. Liu and Horne (Liu and Horne, 2011; Liu and Horne, 2012) used a simple core and approaches to data analysis based on the convolutional core method for interpreting data from permanent downhole sensors. The authors showed that the convolutional core method perfectly removes noise, but it turned out that this algorithm works very slowly (Liu and Horne, 2013; Liu and Horne, 2013). The authors of (Tian and Horne, 2019) also examined the use of machine learning methods for interpreting pressure, flow rate, and temperature data from permanent downhole sensors. In this paper, three machine learning methods were applied, such as linear regression (LR), core method, and ridge core regression. In addition, the authors showed that machine learning can simulate the generated data from the PDG, even when the physical model is complex.

In (Cao et al., 2016), the use of machine learning methods for predicting productivity of existing and new wells is considered. The authors built an ANN that predicts the productivity of wells using their own history. However, the authors do not claim that ANN prediction is a substitute for empirical or numerical simulation for predicting well production. The work suggests that ANN prediction should be used to provide confidence in data-based prediction methods. There is another work in which machine learning methods were built to predict Montney and Duvernay well production (Shengnan, 2019). Several ML methods were considered, of which the random forest method was identified by the most accurate model for their task. This method gave the authors higher forecast accuracy due to

the absence of over-fitting problems, and the determination coefficient R^2 was 0.75 for Montney and 0.68 for Duvernay wells.

This article discusses methods for predicting oil production using machine learning methods. Regression algorithms are implemented: linear and polynomial. Comparison and analysis of the forecasting results, based on the above methods is made. The training and test data are modeled synthetically, based on the two-dimensional Buckley-Leverett model.

Method and Theory

This section describes the machine learning methods. And also the generation of a data set for training and testing is considered.

Problem statement

The purpose of this study is to use machine learning methods to predict oil production. A synthetic data set was obtained using the mathematical model of Buckley-Leverett, which is used to calculate the hydrodynamics and determine the distribution of saturation in oil problems.

The Buckley-Leverett model is written as follows:

$$m\frac{\partial s}{\partial t} + div\vec{v}_1 = q_1$$

$$-m\frac{\partial s}{\partial t} + div\vec{v}_2 = q_2$$

$$\vec{v}_i = -K_0\frac{f_i(s)}{\mu_i}\nabla P$$

where K_0 – permeability tensor, s- water saturation, q_i - source or sink, f_i and μ_i - relative phase permeabilities and viscosities of liquids of the corresponding phases, which are dependences of the following form:

$$f_1(s) = s^{3.5};\ f_2(s) = (1-s)^{3.5};$$

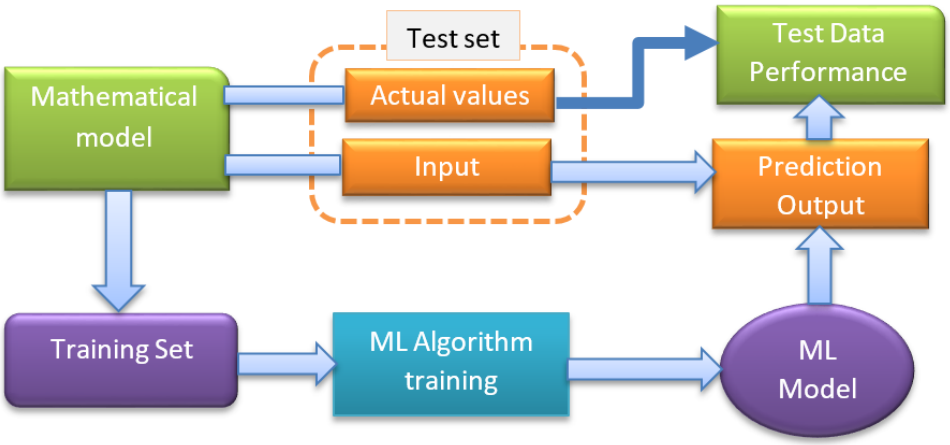

Figure 1 *Workflow of building a machine learning model.*

Figure 1 describes the process of building a machine learning model in this study. In this work, the obtained synthetic data from a mathematical model were divided into a training and test sample. Four

parameters were taken as input parameters of the machine learning model, and the oil recovery coefficient was taken as the output parameter.

Dataset generation

As input parameters, various combinations of parameters of the oil production problem (porosity, viscosity of the oil phase and absolute rock permeability) were taken (Table 1). And as the output parameter, the value of the oil recovery coefficient was chosen.

Parameters	**Number of variation**
Porosity	41
Viscosity of oil phase	41
Permeability	6

Table 1. *Input parameters.*

Thus, in this work, the number of sample pairs is 41 * 41 * 6 = 10086. Using the Buckley-Leverett model, 6 synthetic data packets were generated for various permeability indices. Each data packet contains the values of viscosity, porosity and oil recovery coefficient (if we take into account the data for each time layer, then the total amount of data is 403440). Oil viscosity varies in the range 0.1 - 0.5, porosity in the range 0.1 - 0.3, and various permeability options (Figure 2).

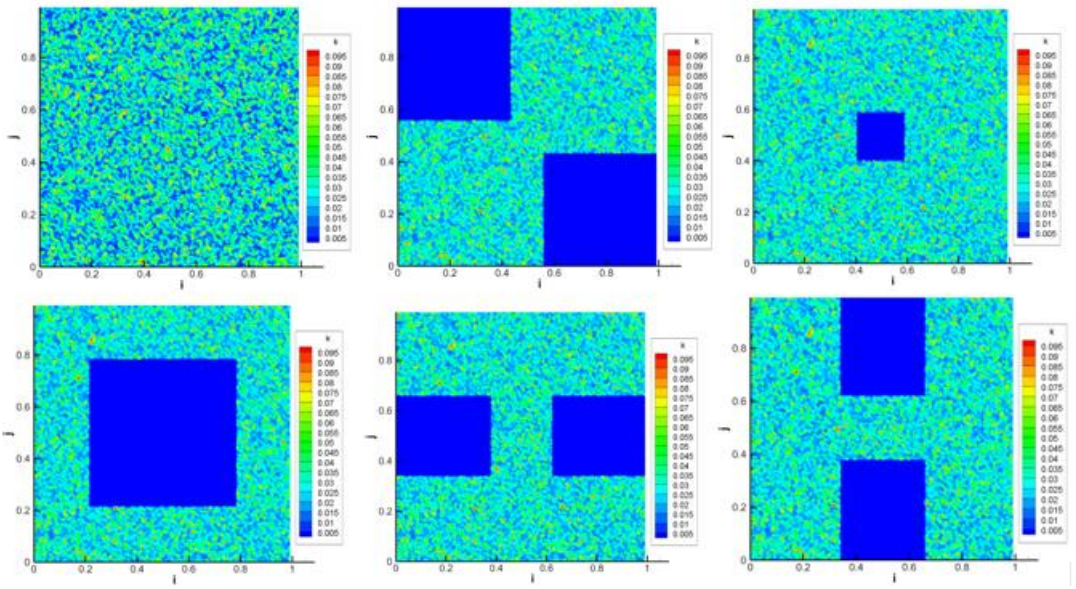

Figure 2 *6 options for absolute permeability.*

Machine Learning Methods

In this paper, we consider the task of learning with a teacher, which is one classes of machine learning problems. Our task belongs to the class of regression problems in terms of machine learning methods. A synthetic data set obtained from a mathematical model: absolute permeability k, porosity p, viscosity μ, time iteration t and oil recovery coefficient η. In our case, the oil recovery coefficient is represented as the objective function y, and the other four data are presented as signs of x.

$$x^{(i)} = \begin{matrix} k^{(i)} \\ p^{(i)} \\ \mu^{(i)} \\ \lfloor t^{(i)} \rfloor \end{matrix}, \quad i = 1, \dots, m \tag{1}$$

where $x^{(i)}$ is sign of i^{th} training example.

$$y^{(i)} = \eta^{(i)},\ i = 1, \dots, m \tag{2}$$

where $k^{(i)}$, $p^{(i)}$, $\mu^{(i)}$ and $t^{(i)}$ are absolute permeability, porosity, viscosity, temporary iteration on i^{th} data, and m is the number of training examples (training example m = 403440). Thus, x is an $(n_x + 1) \times m$ matrix, and the objective function y is an $m \times 1$ vector. The regression model can be written as follows:

$$y^{(i)} = h\left(x^{(i)}\right) + \varepsilon^{(i)}, i = 1, \dots, m \tag{3}$$

where model h describes a pattern between x and y, and ε^i is a model error and measures some discrepancies. Consider the subgenus about methods.

Linear Regression. In multiple linear regression, the hypothesis function h is described as follows:

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1 + \cdots + \theta_{n_x} x_{n_x} \tag{4}$$

where n_x is the number of features (in our case, n_x= 4 given from Eq.1). θ is a $n_x + 1) \times 1$ vector with model parameters (coefficients). The parameter θ_0 corresponds to $x_0 = 1$. Eq.4 can be written as follows:

$$y^{(i)} = \theta_0 + \theta_1 k^{(i)} + \theta_2 p^{(i)} + \theta_3 \mu^{(i)} + \theta_4 t^{(i)} + \varepsilon^{(i)},\ i = 1, \dots, m \tag{5}$$

To evaluate regression models, a quadratic loss function is often chosen. The mean square error (MSE) is often used as an estimate of the loss function between the target and the predicted function:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - y_{pred}^{(i)})^2 \tag{6}$$

Using linear regression, the model was trained with four input parameters and oil recovery coefficient. As a result, the trained model predicts the value of the oil recovery coefficient based on test data. Although multiple linear regression is very simple, the model has several good advantages. The linear regression model frees the engineer from the need for good physics knowledge in this study. This model is well-trained and highly interpreted, since all independent variables of multiple regression directly affect the target function. Consequently, the influence of input parameters is easily detected and visualized.

Polynomial Regression. Polynomial regression is essentially a type of regression in which the ratio of the independent features of x and the dependent objective function y is modeled as a polynomial of n-th degree.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_1 x^2 + \cdots + \theta_n x^n,\ i = 1, \dots, n \tag{7}$$

where n is the degree of the polynomial that is used to transform the linear regression model. Polynomial regression may have a non-linear curve, but the model is still considered linear, since the model parameters associated with the attributes are linear.

In this work, polynomial regression (PR) is used as a special case of multiple linear regression. Since, an increase in the n degree of the polynomial adds data nonlinearity to linear regression. However, this does not mean that with an increase in the degree of polynomial, the model will learn even better. There are problems with under-fitting and over-fitting. To select the optimal model, one needs to find a compromise between displacement and dispersion.

Regression models are used to solve over-fitting problems in regression models. A regression model that uses L1 regularization is called Lasso Regression, and a model that uses L2 is called Ridge Regression. The regularization of L2 adds the coefficient of quadratic magnitude to the loss function and is presented in the following form:

$$\sum_{i=1}^{m} (y_i - \sum_{j=1}^{n_x} x_{ij}\theta_j)^2 + \lambda \sum_{j=1}^{n_x} \theta_j^2 \quad (8)$$

where λ is a setting parameter. A well-chosen value of paramatra λ helps to avoid the problem of over-fitting. A regularization of L1 adds the absolute value of magnitude to the loss function and is presented in the following form:

$$\sum_{i=1}^{m} (y_i - \sum_{j=1}^{n_x} x_{ij}\theta_j)^2 + \lambda \sum_{j=1}^{n_x} |\theta_j| \quad (9)$$

Artificial neural network. In this work, an artificial neural network with one hidden layer was used. In this study, porosity, viscosity of the oil phase, absolute rock permeability, and time iteration were set as input parameters for ANN. The oil recovery coefficient was set as the output parameter of the neural network.

In addition to inputs and outputs, a neural network consists of one hidden layer. By trial and error, it was found that four neurons in the hidden layer are the optimal number for ANN. Thus, in summary, a neural network consists of three layers: an input layer (carrying four neurons), one hidden layer (carrying four neurons) and one output layer (carrying one neuron) (Figure 3). In addition, relu was used as an activation function for the hidden layer.

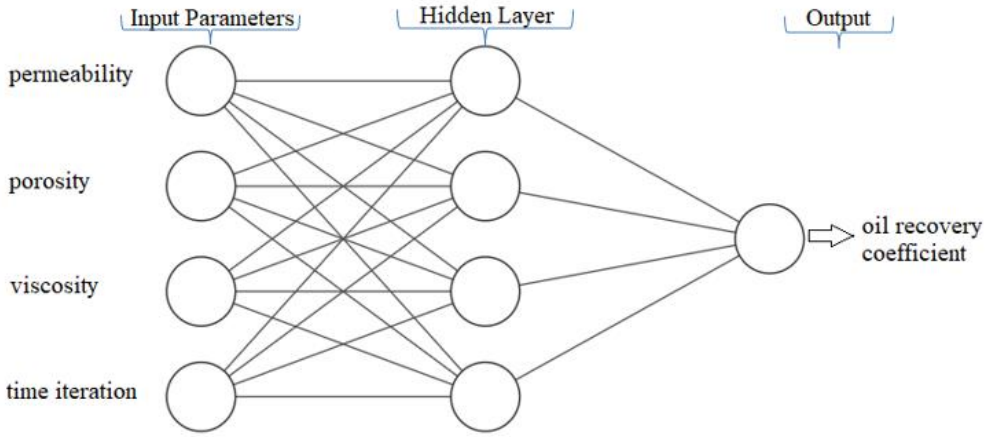

Figure 3 *Neural network architecture used in this study.*

Results and Discussion

The data was divided into a training and test set. For training, 8069 sets (80%) of the total data were used, and for the test the remaining 2017 pairs (20%).

Python was chosen as the runtime environment for machine learning. As mentioned earlier, the total number of sample pairs is 10,086 models. Each sample pair consists of 40 oil recovery factor values. As a result, we have many test pairs, however, in this paper, the results will be shown only for some. Figure 4 shows the result of one test sample pair for LR.

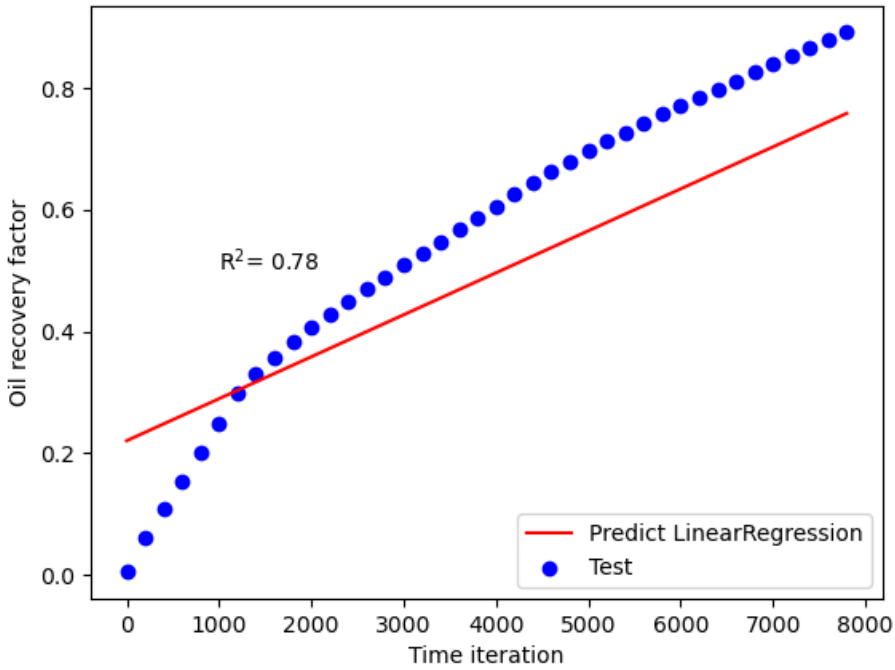

Figure 4 *Oil recovery coefficient on a linear regression model.*

Using polynomial regression (PR) increases the complexity of the model. For training with polynomial properties, it is important to choose the desired model, that is, the degree of the polynomial. Learning a linear regression model using the polynomial properties degree = 2 gives the following result for this set (Figure 5):

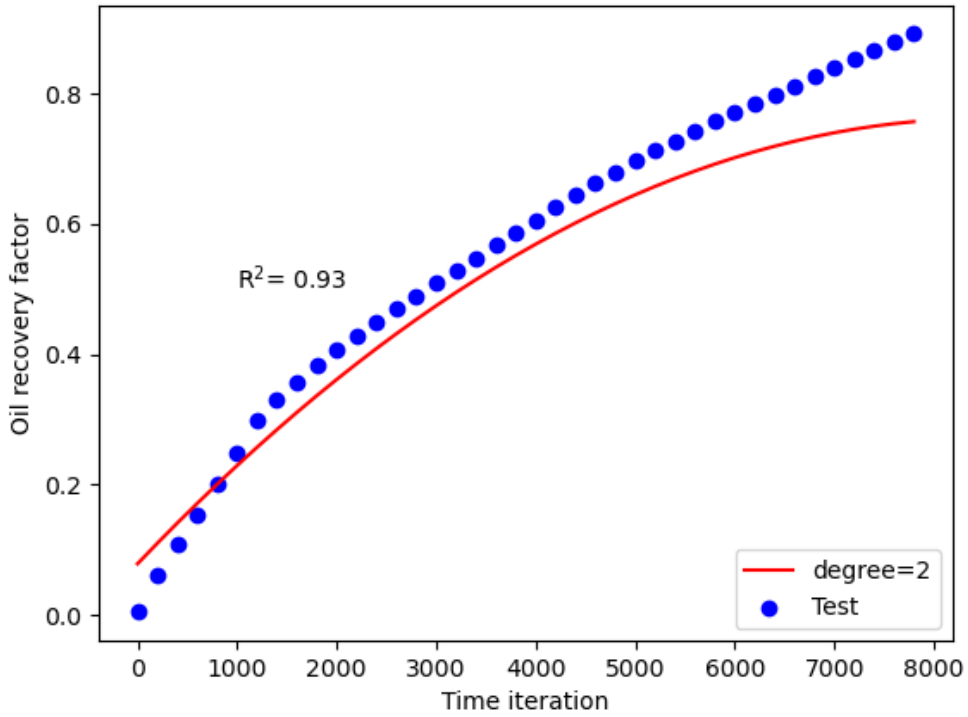

Figure 5 *Oil recovery coefficient on a quadratic polynomial regression model.*

From figure 4, we can see that the predicted LR function does not capture all patterns in the data. Consequently, the LR model has an example of under-fitting. The estimates used for MSE regression

and the determination coefficient R^2 give the following indicators for this set: MSE linear regression is 0.01317 and determination coefficient $R^2$is 0.78.

From this figure 5, it is clear that a quadratic polynomial model trains data better than a linear model. The MSE regression estimates and the determination coefficient R^2 give the following indicators: MSE degree = 2 is 0.0039 and determination coefficient R^2 is 0.93.

In addition, it can be noted that MSE has decreased, and the determination coefficient R^2 has increased compared to the linear model. The following table shows the average MSE score for all 20% of the test sets (Table 2).

Machine learning algorithms	Test sets (20%) MSE
LR	0.0037
Polynomial Regression (PR) degree=2	0.0016
Polynomial Regression (PR) degree=3	0.0084

Table 2 *MSE score for all test case pairs.*

Polynomial regression with the property degree = 3 gives the following result (Figure 6):

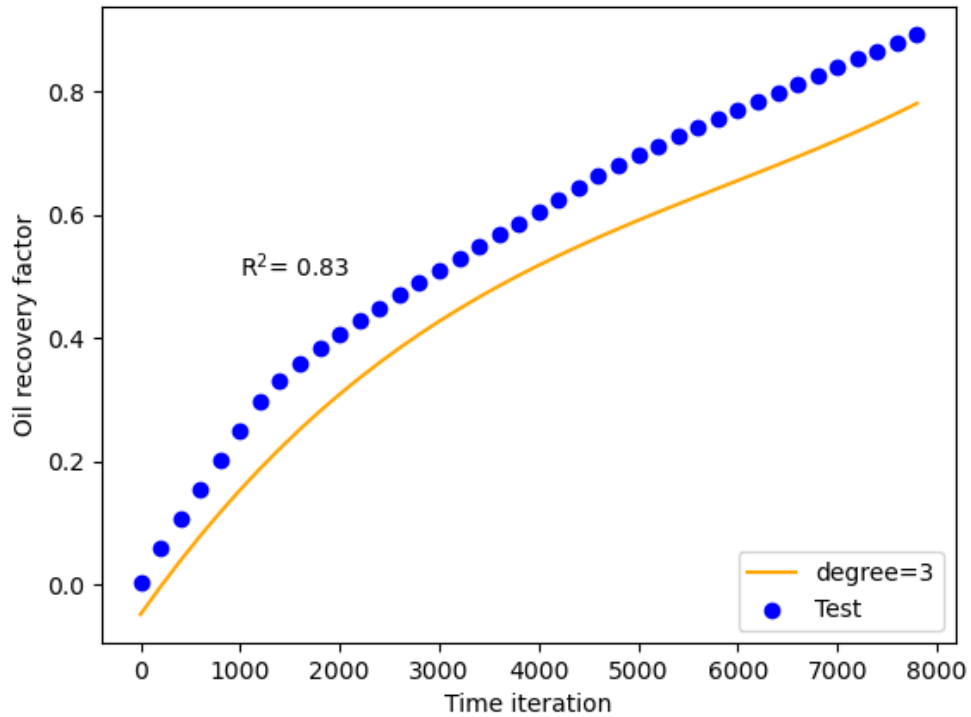

Figure 6 *Oil recovery coefficient on a cubic polynomial regression model.*

From this figure 6 it is noticeable that a cubic polynomial model predicts data worse than a quadratic model. The MSE regression estimates and the determination coefficient R^2 give the following indicators:

MSE degree = 3: 0.00985
Determination coefficient R^2: 0.83

Moreover, one can notice that the MSE, on the contrary, increased, and the determination coefficient R^2 decreased compared to the quadratic model. Thus, the quadratic model is the most optimal for this test sample. However, this pattern is true only for this pair. For the remaining pairs from the entire test sample, the results may be different. This is because the polynomial cubic model in our case has over-fitting due to its high dispersion. From Figure 6, it is noticeable that the cubic model does not match the data well. However, for example, for some test pairs, the cubic model is trained much better (Figure 7). The following table shows the average R^2 score for 80% of training sets and 20% of test sets (Table 3).

Machine learning algorithms	Train sets (80%) R^2	Test sets (20%) R^2
LR	0.87	0.91
PR degree=2	0.95	0.96
PR degree=3	0.97	0.79
PR degree=3 with L1	0.96	0.92

Table 3 *Evaluation of R^2 for all pairs of training and test set..*

From this table, it is noticeable that cubic polynomial regression is trained fairly well with the training set, but the determination coefficient R^2 on test data decreases due to the high dispersion between the data sets at degree = 3.

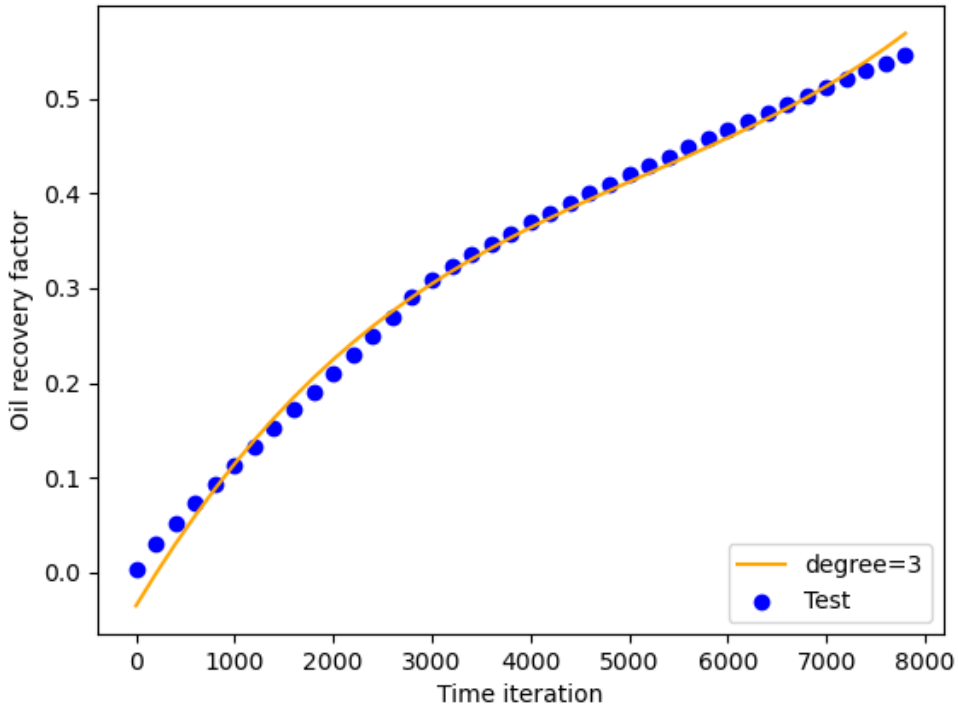

Figure 7 *Oil recovery coefficient on a cubic polynomial regression model.*

It is noticeable from Figure 7 that cubic polynomial regression predicts the objective function for this pair quite well, however, due to over-fitting, other pairs from the test sample may not correspond well to the data as shown in Figure 6. And from Figure 8 it is noticeable that using regularization L1 has been improved cubic polynomial model for this pair. Using L1 regularization, the optimal value of λ was selected. Using the L1 regularization, the cubic polynomial model predicts the function for all test data rather well than the simple cubic model. This can be seen from table 3.

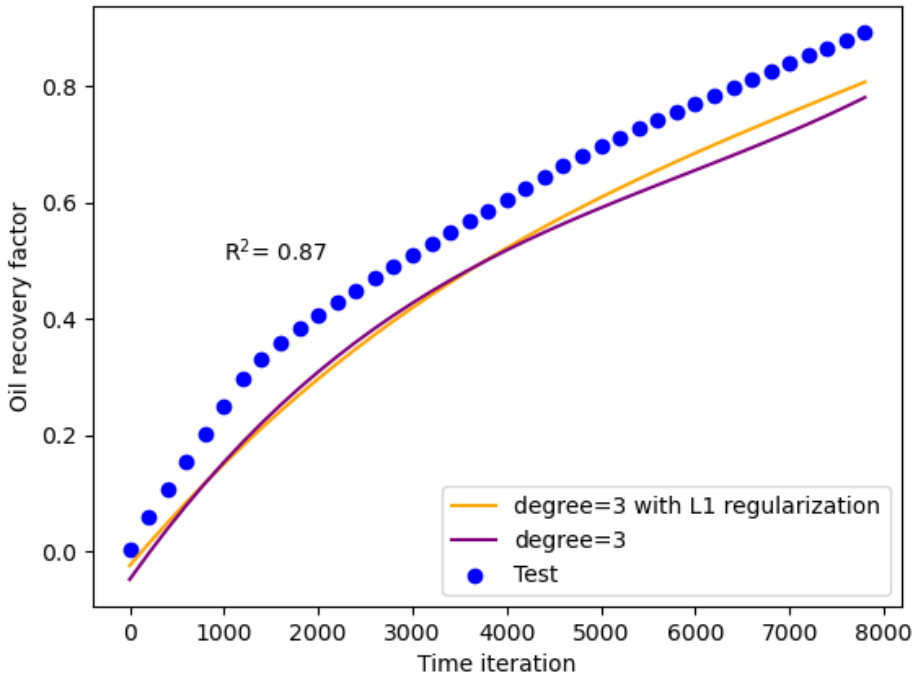

Figure 8 *Oil recovery coefficient on a cubic model of polynomial regression with L1.*

The artificial neural network was trained using 8069 data sets. Each data set consists of 40 oil recovery factor values. In this study, artificial neural network (ANN) hyperparameters were optimally matched. To prevent retraining, the EarlyStopping function was used with the parameter patience = 1, where learning stops at the number of epochs without improvements. ANN was built using the TensorFlow library of the Keras API specification. Keras is a high-level API for building and training models with TensorFlow support. As a layer building in Keras, the Sequential model was used. For one pair of test data, the ANN used shows the following result (Figure 9):

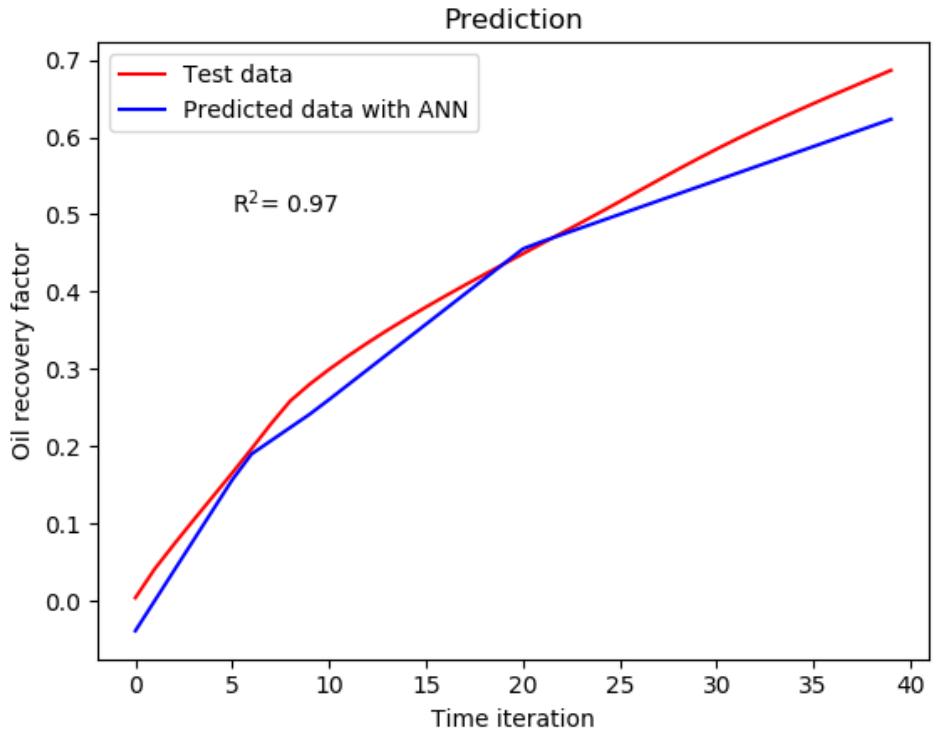

Figure 9 *Oil recovery prediction using an artificial neural network.*

From this figure, you can see that the constructed neural network predicts quite well for this test pair. The MSE regression estimates and the determination coefficient R^2 give the following indicators (Table 4):

Metrics	Train sets (80%)	Test sets (20%)
R^2	0.96	0.97
MSE	0.0016	0.0011

Table 4 *Evaluation Neural Network of for all pairs of training and test set.*

ANN prediction for all 2017 test case pairs (80,680 data) shows the following result (Figure 10):

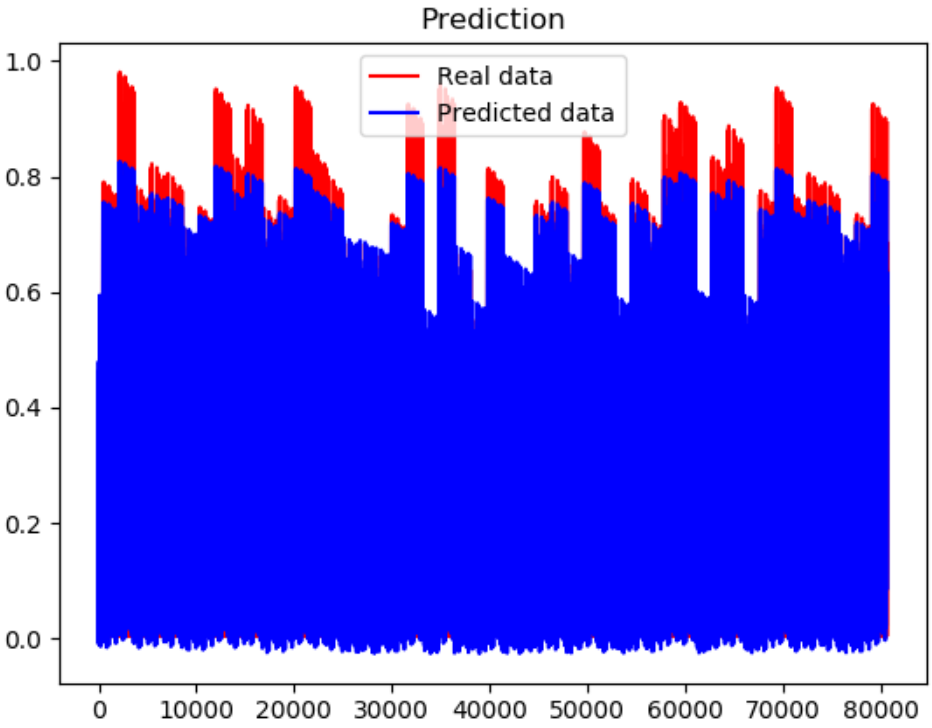

Figure 10 *Oil recovery prediction using ANN for all test pairs.*

Conclusions

This article was devoted to the application of machine learning methods for predicting oil production. In this study, multidimensional linear regression with polynomial properties was used as a machine learning method. Although linear regression is simple, this model is well-trained and highly interpreted. Different degrees of polynomial regression were tested, and it was also revealed that for our synthetic data, the quadratic polynomial model is quite well trained and predicts the value of the oil recovery coefficient. For the quadratic polynomial regression model, the determination coefficient R^2 is 0.96, which is a pretty good result for the test data. An artificial neural network with one hidden layer with optimally selected hyperparameters was built. For the constructed neural network, the determination coefficient R^2 was 0.97, which is slightly better than the quadratic model of polynomial regression.Thus, it is assumed that the considered machine learning methods in this article may be useful for predicting oil production.

Acknowledgments

This research was funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. BR05236447).

References

Akhmed-Zaki, D., Danaev, N., Mukhambetzhanov, S., Imankulov, T. [2012]. Analysis and evaluation of heat and mass transfer processes in porous media based on Darcy-Stefan's model. ECMOR 2012 - 13th European Conference on the Mathematics of Oil Recovery.

Akhmed-Zaki, D.Z., Imankulov, T.S., Matkerim, B., Daribayev, B.S., Aidarov, K.A., Turar, O.N. [2016]. Large-scale simulation of oil recovery by surfactant-polymer flooding. Eurasian Journal of Mathematical and Computer Applications, 4 (1), pp. 12-31.

Aliyuda, K., Howell, J. [2019]. Machine Learning Algorithm for Estimating Oil Recovery Factor Using a Combination of Engineering and Stratigraphic Dependent Parameters. Interpretation. 7. 1-34. 10.1190/int-2018-0211.1.

Breiman, Leo. [2001]. Random Forests Machine Learning 45 (1): 532. DOI:10.1023/A:1010933404324.

Cao, Q., Banerjee, R., Gupta, S., Li, J., Zhou, W., Jeyachandra, B. [2016]. Data Driven Production Forecasting Using Machine Learning. 10.2118/180984-MS.

Danaev, N., Akhmed-Zaki, D., Mukhambetzhanov, S., Imankulov, T. [2015]. Mathematical modelling of oil recovery by polymer/surfactant flooding. Communications in Computer and Information Science, 549, pp. 1-12.

Guo, Z., Reynolds, A. C., Zhao, H. [2018]. A Physics-Based Data-Driven Model for History-Matching, Prediction and Characterization of Waterflooding Performance. Society of Petroleum Engineers. doi:10.2118/182660-MS.

Horne, R. N. [2007]. Listening to the Reservoir—Interpreting Data From Permanent Downhole Gauges. J Pet Technol 59 (12): 78–86. SPE-103513-JPT. https://doi.org/10.2118/103513-JPT.

Imankulov, T.S., Akhmed-Zaki, D. [2016]. Computer modelling of non-isothermal, multiphase and multicomponent flow by using combined EOR technologies. 15th European Conference on the Mathematics of Oil Recovery, ECMOR 2016.

Jreou, G. [2012]. Application of neural network to optimize oil field production. Asian Transactions on Engineering. 3. 1-9.

Krasnov, F., Glavnov, N., Sitnikov, A. [2018]. A Machine Learning Approach to Enhanced Oil Recovery Prediction. 10.1007/978-3-319-73013-4_15.

Liu, Y., Horne, R. N. [2011]. Interpreting Pressure and Flow Rate Data From Permanent Downhole Gauges Using Data-Mining Approaches. Presented at the SPE Annual Technical Conference and Exhibition, Denver, 30 October–2 November. SPE-147298-MS. https://doi.org/10.2118/147298-MS.

Liu, Y., Horne, R. N. [2012]. Interpreting Pressure and Flow-Rate Data From Permanent Downhole Gauges by Use of Data-Mining Approaches. SPE J. 18 (1): 69–82. SPE-165346-PA. https://doi.org/10.2118/165346-PA.

Liu, Y., Horne, R. N. [2013]. Interpreting Pressure and Flow Rate Data From Permanent Downhole Gauges Using Convolution-Kernel-Based DataMining Approaches. Presented at the SPE Western Regional & AAPG Pacific Section Meeting 2013 Joint Technical Conference, Monterey, California, 19–25 April. SPE-165346-MS. https://doi.org/10.2118/165346-MS.

Liu, Y., Horne, R. N. [2013]. Interpreting Pressure and Flow Rate Data From Permanent Downhole Gauges With Convolution-Kernel-Based DataMining Approaches. Presented at the SPE Annual Technical Conference and Exhibition, New Orleans, 30 September–2 October. SPE-166440-MS. https://doi.org/10.2118/166440-MS.

Mirzaei-Paiaman, A., Salavati, S. [2012]. The Application of Artificial Neural Networks for the Prediction of Oil Production Flow Rate. Energy Sources. 34. 10.1080/15567036.2010.492386.

Ristanto, T., Horne, R. [2018]. Machine Learning Applied to Multiphase Production Problems.

Shehata, A. M., El-banbi, A. H., Sayyouh, H. [2012]. Guidelines to Optimize CO2 EOR in Heterogeneous Reservoirs. Society of Petroleum Engineers. Society of Petroleum Engineers. doi:10.2118/151871-MS.

Shengnan, C. [2019]. Application of Machine Learning Methods to Predict Well Productivity in Montney and Duvernay. Calgary Petroleum Club.

Tian, Ch., Horne, R. [2019]. Applying Machine-Learning Techniques To Interpret Flow-Rate, Pressure, and Temperature Data From Permanent Downhole Gauges. SPE Reservoir Evaluation & Engineering. 22. 386-401. 10.2118/174034-PA.