STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **7**, December 2019, pp 826–839.
Published online in International Academic Press (www.IAPress.org)

Development of Oil Production Forecasting Method based on Deep Learning

Fargana Abdullayeva*, Yadigar Imamverdiyev

Institute of Information Technology, Azerbaijan National Academy of Sciences, Baku, Azerbaijan

Abstract Identification of the quick declines of the desirable production fluids and rapid increases of the undesirable fluids are the production problems of oil wells. The main purpose of this work is to develop a method that can forecast oil production with high accuracy, using Deep neural networks based on the debt data of wells. In this paper, a hybrid model based on a combination of the CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) networks, called CNN-LSTM is proposed for the forecasting of oil production time series. The architecture of the proposed CNN-LSTM model is hierarchical. Here, at first the CNN layer of the model is applied to the current time window, then the relationship between the time windows is predicted by applying the LSTM. The challenges of time series prediction often come from the continuity duration of every state. In order to overcome this problem, we try to predict temporal dependency in the certain time window. This issue is solved by the application of the CNN algorithm. Evaluation efficiency of the proposed model is performed on the QRI dataset. The prediction accuracy of the method is tested by RMSLE loss function and the best results are obtained using our proposed in the testing process.

Keywords Petroleum well data, Production data, Deep Learning, Production forecasting, Debt data of the wells, CNN, LSTM.

AMS 2010 subject classifications 60G25, 62M20.

DOI: 10.19139/soic-2310-5070-651

1. Introduction

The existing oil production industry is built on outdated technologies. Such systems can lead to decreased debt of oil production wells, increasing costs of the oil production process and so on. Under these conditions, an effective development of oil fields can increase the oil production and the oil transfer capability of reservoirs, extend the life cycle of oil fields, and have a great importance on economic efficiency.

Identification of the quick declines of the desirable production fluids, and rapid increases in the undesirable fluids are the production problems of the oil wells.

The application of Deep neural networks in the solution of such urgent issues can lead to successful results. Application of these methods to the oil production control and management field can result in high efficiency in various issues, such as the prevention of the inefficient use of energy, optimization of oil extraction time, control of equipment condition, collection, storage and processing of current and history data. Recently, there has been a revival in the application of Deep neural networks in the oil industry by the world's leading oil companies. Chevron is one of the several companies that point out that Deep learning is the most appropriate technology for reservoir characterization. In 2016 researchers at Chevron led an effort to employ Deep neural networks and fuzzy kriging to analyse the viability of a reservoir in Californias San Joaquin Valley [1] .

*Correspondence to: Fargana Abdullayeva (Email: a_farqana@mail.ru). Institute of Information Technology, Azerbaijan National Academy of Sciences, 9A, B. Vahabzade Street, AZ1141, Baku, Azerbaijan.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

From another point of a view, researchers from the SAS Institute have found that merging traditional seismic analysis methods and Deep learning methods has been a more effective tool in efficient discovery of resources in an upstream.

In [2] a supervised and unsupervised approach is proposed to characterize the reservoir on the seismic profiles. Here the seismic images are used as the input data of the proposed Deep Learning method. During the preprocessing phase, patches are created from larger image sets that can reduce the number of features required to represent an image and can decrease the training time needed by algorithms to learn from the images.

In recent years, as Deep learning methods have gained great popularity in the various research areas as a new approach, the researchers have started using CNNs in their work to recognize high-level samples from multidimensional time series data. CNNs have two key features: weights sharing and spatial pooling. This feature has turned the CNN method into a very useful tool for computer vision applications. In this type of issues, the input data of CNN is 2 dimensional (2D) data. However, the CNN model is also applied to natural language processing and speech recognition issues. In this case, the input data of the CNN is 1 dimensional (1D) data [3, 4]. CNN with one layer performs the feature extraction function from the input signals by applyying the convolution operation. Usually, 1D kernel function is used when the input contains one-dimensional time series data. In other words, the convolution operation is applied separately to each size of the data [5]. In addition, some hybrid models are proposed as a result of the combination of CNN and LSTM models for forecasting and classifying time series by learning complicated features [6].

Recently an researchers have conducted extensive investigation in the application of Deep learning methods to the oil production forecasting field. In [7], the method for the forecasting of the oil production time series based on multilayer neural networks with Multi-Valued Neurons (MLMVN) is proposed. In this work, the forecasting capability of the MLMVN model for the prediction of the dynamics of the reservoir is demonstrated. A dataset consisting of monthly production data from 14 wells of oil fields located on the shores of the Gulf of Mexico is used to test the proposed forecasting model. In [8], the application of high order neural networks (HONN) to water, oil and gas production forecasting is considered. In [9] natural gas prediction method based on neural networks is proposed. In [10] the application of the multiple artificial neural networks (MNN) to the evaluation of the future production performance of oil wells based on the time series of monthly production data is considered. MNN is a group of the single artificial neural networks (ANN) which cooperate with each other to solve a specific problem. Each MNN makes predictions in the different time periods. The results obtained here show that the MNN model has shown better results in making long-term predictions compared to the single ANN model. In this work, the data for the experiments are taken from the Saskatchewan Energy and Field center. The activation function of the neural network is sigmoid. Since the value of the sigmoid function changes in the interval [0, 1], the data here is normalized using the min-max algorithm. Deep learning methods such as CNN, LSTM, DBN and others are also applied in the oil production forecasting field.

Since CNN is designated for short-term forecasting, it cannot model the long-term time series sequences. This is related to inability to record time series dynamics efficiently by CNN models that are limited by convolution layer. To resolve this problem the LSTM networks with memory cells are used by researchers. LSTM is using the concept of gates in the memory cells. This function allows the network to learn what hidden states it needs to forget and what states it needs to refresh [11]. LSTM can learn long-term temporal dynamics of the data in the form of sequences and accepts the unprocessed data as an input data. When applying LSTM to the features extracted from unprocessed data by the other Deep learning methods, it allows achieving higher efficiency in the forecasting [12].

In some approaches, LSTM acts as a predictive block of the proposed models [13]. In [14] combination of the conventional and recurrent layers is applied to the time series classification task and high results are achieved.

To ensure accurate forecasting of oil production in oil wells, it would be expedient to use the combination [15] of the above mentioned models.

In this paper, a hybrid model based on a combination of the CNN and LSTM networks is proposed for the forecasting of oil production time series. Here, at first the CNN layer of the model is applied to the current time window and by this, the features are extracted, and then the relationship between the time windows is predicted by applying the LSTM.

In the proposed model, the objective of the CNN layer is to extract features, and LSTM layer is to perform prediction. The main contributions of this work are:

1. The effectiveness of the Deep neural networks in the oil production forecasting is investigated;
2. The new architecture containing an improved Deep CNN and LSTM blocks is proposed for efficient forecasting of the oil production in the wells;
3. Proposed model achieved high results in the forecasting process.

This paper consists of the following sections: Section 2 summarizes some of the methods used in the oil production time series prediction. In section 3 an architecture of the proposed CNN-LSTM model is provided. In section 4 dataset description is provided. In section 5 the results of the comparative analysis of the proposed method with existing methods is described. In section 6 conclusion of this work is provided.

2. Related works

There are many methods involved in Deep Learning, [16, 17] which are divided into three groups: generative, discriminative, and hybrid. Generative architectural methods include autoencoders [18], recurrent neural networks (RNN) [19], Boltzmann machines.

Each layer of the deep network learns independently, bypassing previous pre-training procedure. It then allows checking a good initial approach to run backpropagation algorithm. Depending on the selected model, each layer may be RBM or CNN (Convolutional Neural Network) [20].

Boltzmann Machine (BM) is a network of symmetrically connected stochastic binary units. The units are divided into two groups, describing visible and hidden states (analogy with hidden Markov models). The states of the visible and the hidden neurons vary according to the probability activation functions.

Restricted BM (RBM) is a BM that has no relation between the hidden layer neurons. Due to the special bipartite graph structure, it is possible to clearly find the probabilities of the hidden layers neurons. If a sufficient number of neurons is used in the hidden layer, RBM can generate any discrete distribution.

RBM is a key structural unit for constructing the Deep Belief Network (DBN). DBN is a multilayer network [21], in which the lower layers are sigmoid Belief network, and the upper layer is RBM.

Deep BM (DBM) is sometimes used in the pre-training step instead of the autoencoder. Multilayer architecture of the DBM is the main difference from RBM.

CNN is a multilayered neural network with a special architecture to detect complex features in data. CNNs have been used in image recognition, powering vision in robots, and for self-driving vehicles.

LSTM recurrent neural networks are capable of learning and remembering over long sequences of inputs. LSTMs work very well if the problem has one output for every input, like time series forecasting. But LSTMs can be challenging to use when the problem has very long input sequences and only one or a lot of outputs.

Hybrid DL architecture integrates the generative and discriminative architectures. Deep Neural Network (DNN) can be given as an example of hybrid architecture. In [22] DNN is a cascade of fully-connected hidden layers and often uses the RBM stack as a pre-training stage.

The main purpose of this work is to develop a method that can predict the oil production with high accuracy using the Deep neural networks based on the debt data of wells.

The oil and gas supply chain consists of three streams: upstream, which covers the exploration, development, and production of oil and gas; midstream, which includes the transportation of oil and gas by tanker; and downstream, which includes the refinement and sales processes.

This paper researches the upstream level of the oil industry. The oil production processes are modeled on the basis of hydrodynamic numerical evaluation of the processes in the reservoir, a dataset containing history data on the creation of the oil fields, equipment characteristics, the time-varying geological characteristics of the reservoir, well operation modes, well operation and break time.

The data required to predict oil production are divided into the following groups:

1. **Time and periodicity of information recording.** Determined by the recording time of the measurements.

2. **Characteristics of injection wells.** This group includes the load volumes, the acceleration, the well operation time, the coordinates and numbers of the wells and so on.
3. **Characteristics of the production well.** This group includes the volume of produced water and oil, liquids separation, debt of the wells by oil and gas, total debt, total production, operating time, coordinates of wells, number of production wells and so on.

Depending on the input data a number of Deep learning architectures are proposed. One of the special research directions of the oil and gas industry is the reservoir characterization. In [23] Deep neural networks are used to predict the properties of the oil reservoir, such as porosity, permeability, pressure-volume-temperature (PVT), depth, drive mechanism, structure and seal, diagenesis, well spacing and well-bore stability. Some of these properties of the reservoir are used to detect drilling problems, to determine reservoir quality, to optimize reservoir architecture, to the identification of lithofacies, and to measure reservoir volume. Here varies methods that perform petroleum reservoir characterization, based on the hybridization of different algorithms with neural networks are also reviewed. In [24] reservoir characterization issue based on neural networks is taken into consideration.

In [25] by applying kernel method to Arps decline model, - a new nonlinear multidimensional forecasting model, titled as the nonlinear extension of Arps decline model (NEA) is proposed. The base structure of the NEA model is the Arps exponential decline equation and nonlinear combinations of time series in the input are created by applying the kernel method to the model. It can effectively determine the nonlinear relation between the input time series and the oil production. In order to evaluate the effectiveness of the NEA model, in this work, the experiments are conducted on the data taken from oil fields in China and India. In order to improve the ability of the model, the combination of the decline curve methods with intellectual methods is provided.

In [26] oil well production model based on the MLP method is proposed using production data. In [27, 13] LSTM type recurrent neural network is used in the recognition of top-level templates and value forecasting issues to study the temporal and sequential features of the time series. While the above-mentioned methods provide good results in the recognition of templates, these methods encounter great difficulties in recognition of the temporal features as a sequence.

3. Proposed method

In recent years, the combination of CNN and LSTM layers has gained more attention [28, 29, 30]. Two types of combinations exist. The first group combines by applying separate layers of convolutions or LSTMs one after another. The second group of combinations includes convolution into LSTMs or general RNNs.

In this section, we introduce a CNN+LSTM Deep network for time series prediction.

An architecture of the proposed CNN+LSTM Deep Learning model is described in Figure 1.

Our model has two major components: the CNN layer and the LSTM layer. These layers are stacked from bottom to top, and statement of these layers individually refers to capture features from the sensor sequence in sliding windows and from the sequence of states.

The algorithm of the proposed CNN-LSTM model for the prediction oil production properties is as follows:

Step 1. The oil well characteristics determination. For a time series problem, here the observation from the last time step (t-1) is used as the input (a sequence of history values) and the observation at the current time step (t) is used as the output (the value on next timestamp).

Step 2. Building training samples based on CNN. In our case, 48 samples are used for the neural network training and 12 samples are used for forecasting. Note that, there is no minimum or maximum for training and testing sample size. In the proposed model the number of samples can be taken in any size. Generally, higher number of samples for training can ensure better performance.

Step 3. Building the network of the proposed hybrid Deep Learning architecture. Input the oil production debt data to the constructed network and train the neural network based on these data.

Step 4. After learning the neural network, implement the testing phase and find the required solution.

Step 5. Calculation of the loss caused by prediction.

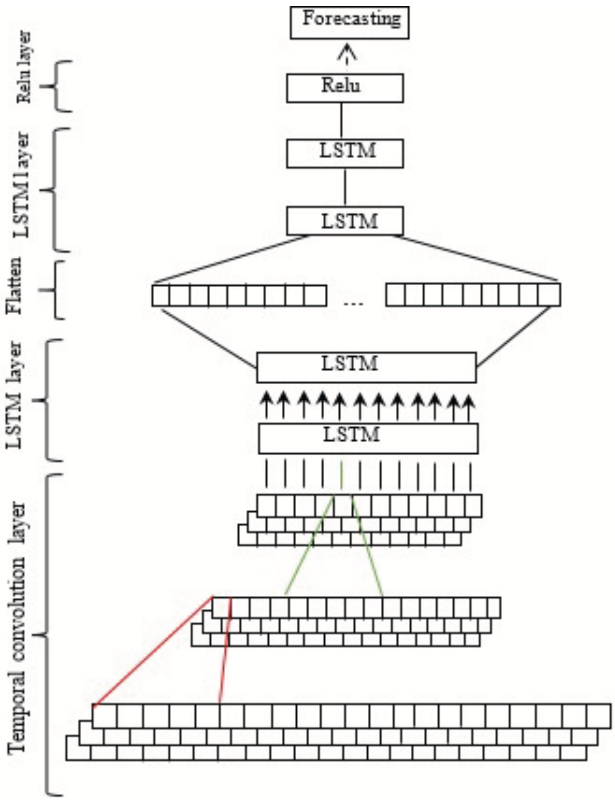

Figure 1. The architecture of the CNN+LSTM model

Here, the time series are used to perform the forecasting.

A time series is a sequence of real-valued data points with timestamps generated by D different sensor channels. The raw data x_{ti} at any timestamp i is a multidimensional vector that can be described as a tuple $\left(t_i,\ \vec{V}_i\right)$, where t_i is a timestamp and $\vec{V_i} \in R^D$ is a D dimensional vector of measurements.

The challenges of time series prediction often come from the continuity in a time of every state. In order to overcome this problem, we try to predict temporal dependency in the certain time window. This issue is solved by the application of the CNN algorithm.

Convolutional neural networks

Filtering time series data is an important tool to improve prediction performance. CNN allows the automatic creation of good filters. Although the success of CNN emerged from the vision domain [31], they also demonstrated potential in time series applications, for example, in activity classification [32]. One main difference from the most classical filters is that convolutional filters are multivariate and thus combine inputs. We define the $1D$ convolution operator at some time step t as:

$$\hat{y}_{\bar{t}}^{conv} = conv(x_{\bar{t}}) = \Phi_{conv}\left(b + \sum_{i=1}^{|k|} x_{\bar{t}-\frac{|k|}{2}+(i-1)} k_i\right) \tag{1}$$

where k is a kernel vector that depicts the filter. Successively applied operations lengthen the amount of time in a filter and lead to potentially more meaningful, but abstract features. A CNN then consists of multiple conv layers and a dense layer at the end.

The number of filters or neurons of the conv layer corresponds to the number of output channels. All filters are applied to the input series $X_t^\delta = \left(x_{t-(\delta+1)},\ ...,\ x_t\right)$ with time frame length δ at the time t:

$$conv\left(X_t^\delta\right) = \begin{pmatrix} conv_1(x_{t-(\delta+1)}) & conv_1(x_{t-(\delta+1)+s}) & \cdots \\ . & & \\ . & & \\ . & & \\ conv_n(x_{t-(\delta+1)}) & & conv_n(x_t) \end{pmatrix} \tag{2}$$

where s is a stride, skips intermediate steps. This layer uses less trainable weights than a dense layer who would learn on X_t^δinstead of just x_t, as the kernel applies to all time steps.

In the proposed architecture the structures of the individual convolutional subnet for the input p_i at different time are the same. Assume that input $p_i = \{x_{t_1},\ x_{t_2},\ ...,\ x_{t_l}\}$ is a $L_0 \times D_0$ tensor, L_0 equals to the sequence length of sliding window l, where D_0 equals to the number of sensor channels D. For each time interval l, the matrix p_i will be put in into a CNN architecture. To learn temporal dynamics from the input p_i the $1D$ filters with shape $(k,\ 1)$ is applied. In this paper, the size of filters in every convolutional layer is the same, and the convolution is only computed where the input and the filter fully overlap. For each convolutional layer, the model learns f filters, through which the model got more nonlinear functions and learned more global information of the current sequence, and use ReLU as the activity function. The convolutional layer is not followed by a pooling operation, as the next LSTM layer requires a data sequence to process. The shape of feature maps output by the m convolutional layers is $L_m \times D_o$, where $L_m = L_0 - m(k-1)$ is the new sequence length of the time window. The shape of the output of CNN has form $(f,\ L_m,\ D_0)$. For putting this output to the input of the LSTM model it needs to change to form $(L_m,\ f \times D_0)$. So the length of input of the LSTM is L_m, the dimension of input is $D1$, where $D_1 = f \times D_0$. Here, we flattened the output matrix of layers into vector V_i, which is considered as the feature representation of a high-level pattern.

LSTM networks

Unlike dense layers, LSTM utilizes the temporal information of a series by sequentially processing the data. This differs from conv layers because its kernel weights are trained to find features throughout the time series. RNNs have a memory for previous inputs and outputs because they introduce the previous output with a hidden weight back to themselves. In other words, the input series X_t^δ is sequentially processed vector for vector in increasing amount of time steps δ. LSTM is one of the types of RNN models. In RNN with an increasing amount of time steps δ the gradients begin to vanish exponentially, which impedes the learning process. Unlike RNN in LSTM the applied gating mechanisms prevent the loss of far-away information. LSTM is excellent at time series prediction tasks [33]. In an LSTM layer, the hidden weights h get adjusted with every time step $\overset{\vee}{t} \in (t-(\delta+1)\,,\ ...,\ t)$ by taking the elementwise product $(\circ)$ of a output gate o and the activation of a cell c. This cell determines how much of the previous cell is retained with a forget gate f and adds it to the product of an input gate i and an input modulation j. All gates consider the hidden weight from the preceding time step. We give a formal representation as follows:

$$\begin{gathered} \hat{y}{}^{LSTM}_{\overset{\vee}{t}} = b_y + w_{hy} h_{\overset{\vee}{t}} \\ h_{\overset{\vee}{t}} = o_{\overset{\vee}{t}} \circ \Phi_h(c_{\overset{\vee}{t}}) \\ c_{\overset{\vee}{t}} = f_{\overset{\vee}{t}} \circ c_{\overset{\vee}{t}-1} + i_{\overset{\vee}{t}} \circ j_{\overset{\vee}{t}} \\ i_{\overset{\vee}{t}} = \Phi_i\left(b_i + W_{xi} x_{\overset{\vee}{t}} + W_{hi} h_{\overset{\vee}{t}-1}\right) \\ f_{\overset{\vee}{t}} = \Phi_f\left(b_f + W_{xf} x_{\overset{\vee}{t}} + W_{hf} h_{\overset{\vee}{t}-1}\right) \\ o_{\overset{\vee}{t}} = \Phi_o\left(b_o + W_{xo} x_{\overset{\vee}{t}} + W_{ho} h_{\overset{\vee}{t}-1}\right) \end{gathered} \tag{3}$$

$$j_{\overset{\vee}{t}} = \Phi_j\left(b_j + W_{xj}x_{\overset{\vee}{t}} + W_{hj}h_{\overset{\vee}{t-1}}\right)$$

with non-linear activation functions φ (tanh, sigmoid, relu), weight matrices W , and bias vectors b, whereas the relations to the gates are visible in the subscripts.

As a result of the operation of the convolutional layer, the sensor data $p_i = \{x_{t_1},\ x_{t_2},\ ...,\ x_{t_l}\}$ in a sliding window has been processed to a feature vector V_i. Concatenating all n vectors $\{V_1,\ V_2,\ ...,\ V_n\}$ into a n-row matrix V, which is the input of the LSTM layer. Then the output of LSTM at every time step is passed into a relu output layer which yields the prediction outcome y_i. So the input of this part of LSTM is V, the output is Y.

Dropout

Dropout established itself as new and reliable regularizer that helps to avoid overfitting [34]. Essentially, dropout deactivates a random part of the input for every neuron with probability p. Here, we define it formally as follows. Every weight matrix W is now interpreted as a random matrix that contains the weight matrix $\hat{W}$ and initializes a different vector z for every neuron:

$$W = W \bullet diag\left([z_j]_{j=1}^{n}\right) \tag{4}$$

This dropout vector z has its values randomly set to one instead of zero with a probability p. Further, the dropout vector changes for every training step.

For LSTM layers not only the input should be partially dropped, but also the recurrent units. To that end, equation (3) is reparametrized to:

$$\begin{pmatrix} i_{\overset{\vee}{t}} \\ f_{\overset{\vee}{t}} \\ o_{\overset{\vee}{t}} \\ j_{\overset{\vee}{t}} \end{pmatrix} = \begin{pmatrix} \Phi_i \\ \Phi_f \\ \Phi_o \\ \Phi_j \end{pmatrix}\left(b + \begin{pmatrix} x_{\overset{\vee}{t}} \\ h_{\overset{\vee}{t-1}} \end{pmatrix} \cdot W\right) \tag{5}$$

4. Dataset description

The Deep Learning method proposed in this paper is tested on the dataset presented by QRI (Quantum Reservoir Impact). This dataset consists of data from seven drilling reservoirs. Field name, formation type, well name, oil production, water production, gas production is a feature of this dataset.

Each of the seven reservoirs consists of several wells. Each well has a name. Data on oil, water and gas production from each well are recorded in each month. In this paper, only oil production is considered. Here, each well is divided into smaller chunks (parts). Each of these chunks contains 60 points (these points represent the number of months). Here 48 points of these months are input data and expected 12 months (the points to be predicted) are outputs of the proposed system.

To standardize the data of each chunk, they are normalized. In addition, according to the oil well index the separation of data into training, validation and test data is conducted. Where training data consist of 5041 chunks, validation data 1026 chunks, and the test data 1144 chunks.

Efficiency assessment metric of the model (RMSLE)

In this paper for the evaluation accuracy of the proposed method, The Root Mean Squared Logarithmic Error (RMSLE) is used. RMSLE is to compare the predictive value with the true value, and is calculated as the square root of the squared bias plus squared standard error:

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log\left(a_i+1\right) - \log\left(b_i+1\right)\right)^2} \tag{6}$$

Method	RMSLE metric
LSTM	0.193466
CNN	0.187198
Hybrid CNN+LSTM	0.186891

Table 1. Prediction loss of 12 points on QRI dataset

where n is the total number of observations in the testing dataset, a_i is predicted value, and b_i is the actual value. RMSLE is a method to measure the error rate. According to this measure, a low RMSLE value indicates a better forecasting solution.

5. Experiments

The dataset is divided into three subsets, training, validation and testing. The training and testing data are consists of 5041 and 1026 monthly production taken from several wells. The validation data is consist of 1144 productions. Here the training data is used to train the neural network. Validation data is used to determine the effectiveness of the neural network on samples, which are not used in the training process. Training and validation are done simultaneously. These two data are used to configure the neural network parameters. The test data is used to evaluate the overall effectiveness of the model after determination the neural network parameters values.

Comparative analysis of the proposed CNN+LSTM method with CNN and LSTM methods is conducted. Here the main purpose of the CNN+LSTM approach is the forecasting the monthly production in various periods starting from 1 month to 48 months.

As a result of this analysis, loss values of the LSTM, CNN and proposed hybrid CNN+LSTM models are calculated based on the RMSLE metric and added to Table 1.

As seen from Table 1, the results of the proposed hybrid CNN+LSTM model outperforms the results of other models. Thus, during prediction of the 12 points in the test data the RMSLE value of the CNN+LSTM model is obtained 0.186891, but in the CNN model this value reached to 0.187198 and in the LSTM model to 0.193466. RMSLE is a method to measure the degree of loss (error rate), so smaller RMSLE value indicates a more accurate model. As seen from here, in the assessment of oil productivity of the wells, the effectiveness of the CNN+LSTM method is better than other algorithms.

To better illustrate the results of Table 1, the loss dynamics of each model are depicted in Figure 2.

As seen from the figure 2, by training the CNN+LSTM model the loss value starting from the first iteration to the last step is slow down by changing very smooth, compared to other methods at the end it achieves the lowest loss value. However, the loss values of the CNN and LSTM models separately varies not smooth. In some cases, there is a situation where the loss value of the next iteration is bigger than the previous one. As the loss was less than other methods in proposed hybrid CNN+LSTM model during the forecasting, this model is considered to be more effective.

To evaluate the importance of methods, in the scientific papers statistical significance tests are used. For this purpose, the methods are launched 70 times on the QRI dataset (Table 2).

As seen from Table 2, better results are achieved by the hybrid CNN+LSTM model. For the verification the robustness of the prediction methods, the boxplot representation is created based on values in Table 2 (Figure 3).

As seen from Figure 3, the hybrid CNN+LSTM model has achieved better results than other methods. Here, because of the increase and decrease of values are changing with the small leap over every iteration, the boxplot representation of this method becomes tighter.

The proposed hybrid CNN+LSTM model has predicted the future oil production data for the next 12 months very accurately. Forecasting of the hybrid CNN+LSTM model on the data, taken from several chunks is depicted in Figure 4.

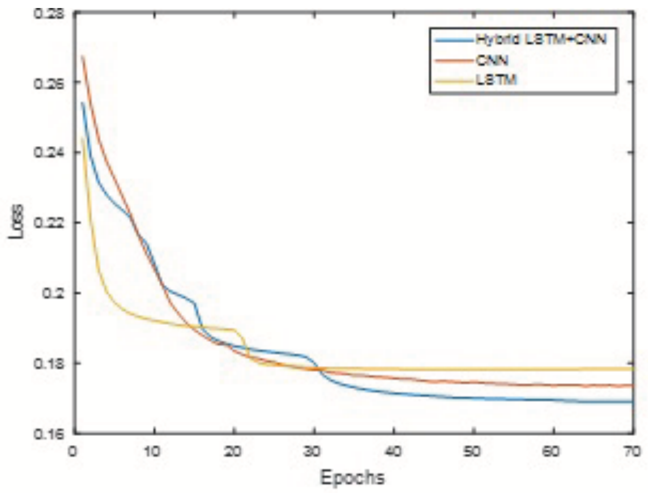

Figure 2. Loss dynamics by number of epochs

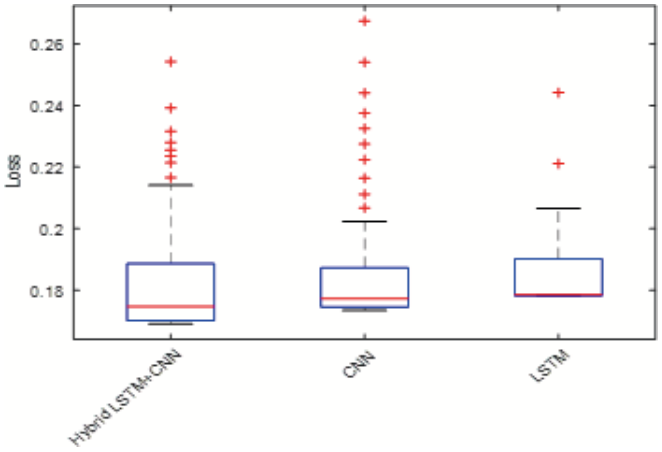

Figure 3. The boxplot representation of the prediction methods based on loss metric

In Figure 4 the production dynamics of the oil well by time function is depicted.

In general, oil production is changing for a variety of reasons. For example, well-stimulation operations cause changes in the wellbore area, which leads to an increase in production. The decrease in production is usually caused by reduced pressure in the reservoir or the degradation of the mechanical condition of the production well. An effective way to slow down the degradation is to carry out additional recovery operations, such as water flooding. Another way to restore the pressure is the shutdown the well for a certain period of time. If the production is down to zero during the closure of the well, then it usually begins to rise. But over time, the decline is repeated.

As seen from Figure 4, constructed neural network predicts downward slopes, flat lines and sudden upward jumps with high precision. Most approaches cannot accurately predict sudden upward jumps in time series [35]. As seen in Figure 4, the oil production rate is increased considerably over a certain period of time and it started to decline later.

In order to provide optimized training of the proposed multilayer neural network, the numerous experiments at various values of the batch size parameters are conducted. Batch size is the number of training data used in one iteration. Here, in high values of the batch size parameter, the effectiveness of the model falls (Table 3 and Figure

Table 2. The loss dynamics of Deep Learning methods by number of epochs

Epochs	**Hybrid CNN+ LSTM**	**CNN**	**LSTM**	**Epochs**	**Hybrid CNN+ LSTM**	**CNN**	**LSTM**
1	0.2542	0.2674	0.2441	36	0.1727	0.1766	0.1785
2	0.2391	0.2540	0.2211	37	0.1723	0.1763	0.1784
3	0.2316	0.2440	0.2066	38	0.1720	0.1761	0.1785
4	0.2280	0.2375	0.2004	39	0.1717	0.1760	0.1784
5	0.2255	0.2325	0.1973	40	0.1715	0.1757	0.1784
6	0.2236	0.2275	0.1954	41	0.1713	0.1755	0.1783
7	0.2214	0.2224	0.1941	42	0.1711	0.1756	0.1783
8	0.2166	0.2164	0.1933	43	0.1709	0.1752	0.1783
9	0.2141	0.2112	0.1926	44	0.1708	0.1750	0.1783
10	0.2083	0.2068	0.1922	45	0.1706	0.1747	0.1783
11	0.2023	0.2023	0.1917	46	0.1705	0.1749	0.1783
12	0.2005	0.1972	0.1913	47	0.1703	0.1748	0.1783
13	0.1995	0.1942	0.1909	48	0.1702	0.1746	0.1783
14	0.1985	0.1915	0.1906	49	0.1702	0.1744	0.1783
15	0.1971	0.1895	0.1904	50	0.1701	0.1747	0.1783
16	0.1897	0.1880	0.1903	51	0.1700	0.1744	0.1783
17	0.1876	0.1866	0.1900	52	0.1699	0.1743	0.1783
18	0.1864	0.1854	0.1899	53	0.1698	0.1742	0.1783
19	0.1856	0.1853	0.1897	54	0.1698	0.1741	0.1783
20	0.1849	0.1835	0.1895	55	0.1697	0.1740	0.1783
21	0.1845	0.1825	0.1872	56	0.1697	0.1738	0.1783
22	0.1840	0.1817	0.1815	57	0.1696	0.1739	0.1783
23	0.1837	0.1814	0.1802	58	0.1695	0.1740	0.1783
24	0.1834	0.1807	0.1797	59	0.1695	0.1739	0.1783
25	0.1831	0.1803	0.1795	60	0.1695	0.1737	0.1783
26	0.1828	0.1796	0.1793	61	0.1694	0.1739	0.1783
27	0.1826	0.1792	0.1791	62	0.1693	0.1738	0.1783
28	0.1823	0.1787	0.1790	63	0.1692	0.1739	0.1783
29	0.1818	0.1784	0.1789	64	0.1691	0.1735	0.1784
30	0.1801	0.1781	0.1788	65	0.1691	0.1738	0.1784
31	0.1769	0.1780	0.1787	66	0.1691	0.1736	0.1784
32	0.1753	0.1774	0.1786	67	0.1691	0.1738	0.1784
33	0.1743	0.1772	0.1786	68	0.1691	0.1735	0.1784
34	0.1737	0.1770	0.1785	69	0.1691	0.1737	0.1784
35	0.1731	0.1765	0.1785	70	0.1691	0.1736	0.1784

5). Thus, the results of the experiments performed with 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000 batch size parameters are shown in Table 3 and Figure 5.

In our work the experiments are conducted by the following parameters: a number of neurons 50, the batch size 20, the activation function relu, the optimization function sgd, the number of iterations 1000.

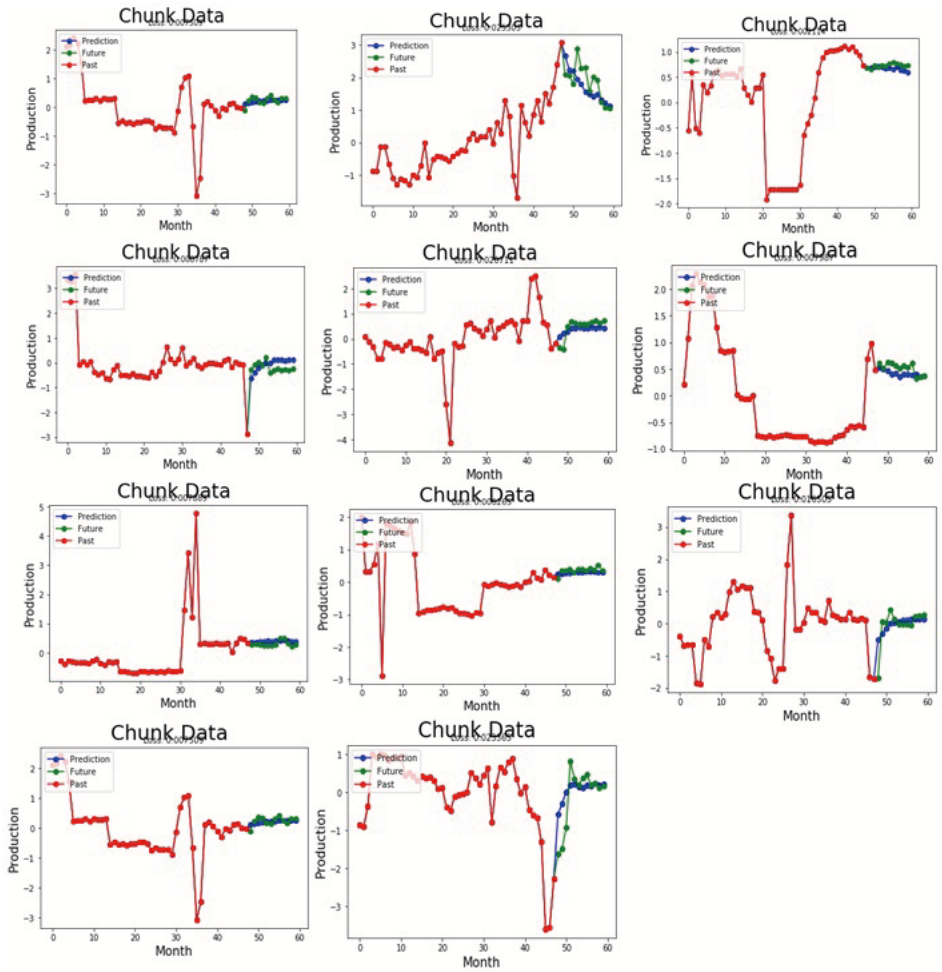

Figure 4. The boxplot representation of the prediction methods based on loss metric

Table 3. Comparison Deep Learning methods by batch size

Batch size	LSTM	Deep LSTM (12 layers)	RNN	Deep RNN (3 layers)	CNN	Fully con-nected model	CNN+ LSTM
1000	0.187881	0.187614	0.229302	0.200905	0.188205	0.188684	0.198972
2000	0.188097	0.187203	0.240377	0.201013	0.187753	0.190146	0.214238
3000	0.189732	0.188513	0.248435	0.207155	0.186916	0.194668	0.226295
4000	0.189789	0.188502	0.248436	0.207165	0.187848	0.194710	0.226333
5000	0.189985	0.189151	0.248429	0.208660	0.190121	0.194667	0.226439
6000	0.203144	0.190879	0.260842	0.208144	0.190747	0.214714	0.232704
7000	0.203144	0.190879	0.260842	0.208144	0.190750	0.214692	0.232704
8000	0.203144	0.190879	0.260842	0.208144	0.190750	0.214695	0.232704
9000	0.203144	0.190879	0.260842	0.208144	0.190747	0.214693	0.232704

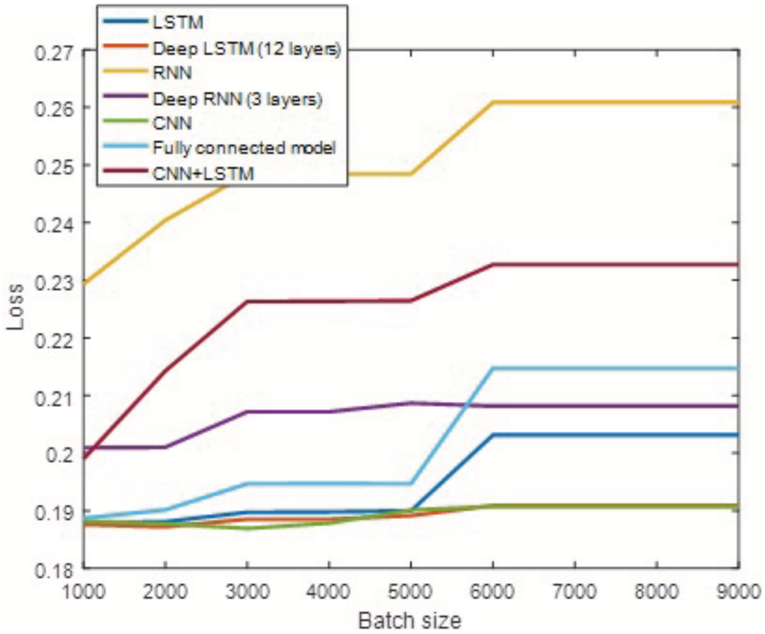

Figure 5. Effectiveness of the deep learning methods by batch size

6. Conclusion

In this paper, we proposed a novel prediction model for the oil well production forecasting based on the hybridization of the CNN and LSTM models, which is called the CNN+LSTM model. The hybridization of these methods is improved forecasting capability of the above mentioned Deep Learning methods.

It has been shown in the case studies that the CNN outperforms the LSTM accuracy, which indicated that the LSTM is eligible to be applied in the nonlinear forecasting problems in the petroleum engineering.

The impact of the batch size parameter and the regularized parameter is also tested in this study. The results show that the smaller batch size makes the CNN+LSTM model be more accurate, and the larger batch size makes the CNN+LSTM unsuitable. And the CNN+LSTM model produced smoother points with a smaller batch size, and it was represented to be more accurate in training with smaller batch size. These results all indicated that in parameters such as the number of neurons 50, the batch size 20, the activation function relu, the optimization function sgd, the number of iterations 1000 the optimal solution would be obtained.

The limitations of this study should also be noticed that we are not proposed an available algorithm to compute the optimal parameters of the proposed CNN+LSTM method. In fact, selection of these parameters is an open problem in the Deep Learning researches, and there is no available algorithm to select the optimal parameters up to now. But it will be interesting and might be possible to find an optimal interval which contains the optimal parameters, and once this interval is found, it will be more computationally efficient to select the optimal parameters for the CNN+LSTM model.

Acknowledgement

This work was supported by the Science Foundation of the State Oil Company of the Azerbaijan Republic (SOCAR).

REFERENCES

1. M.M. Korjani, A.S. Popa, E. Grijalva, S.Cassidy, I. Ershaghi *Reservoir characterization using fuzzy kriging and deep learning neural networks*, SPE Annual Technical Conference and Exhibition, 2016, pp. 1-15.
2. K. Holdaway, P. Hall, J. Silva and I. Kabul, *Surface reservoir characteristics from subsurface seismic images with deep learning methodologies*, In proc. of the International Conference and Exhibition, pp. 239-239, 2016, https://doi.org/10.1190/ice2016-6319324.1
3. O. Abdel-Hamid, A.R. Mohamed, H. Jiang, G. Penn, *Applying convolutional neural networks concepts to hybrid NNHMM model for speech recognition*, In proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4277-4280, 2012.
4. G.E. Dahl, D. Yu, L. Deng, A. Acero, *Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition*, IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 30-42, 2012.
5. M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, J. Zhang, *Convolutional neural networks for human activity recognition using mobile sensors*, In proc. of the IEEE 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), pp. 197-205, 2014.
6. T.N. Sainath, O. Vinyals, A. Senior, H. Sak, *Convolutional, long short-term memory, fully connected deep neural networks*, In proc. of the IEEE International Conference on In Acoustics, Speech and Signal Processing (ICASSP), pp. 4580-4584, 2015.
7. I. Aizenberg, L. Sheremetov, L. Villa-Vargas, J. Martinez-Munoz, *Multilayer Neural Network with Multi-Valued Neurons in time series forecasting of oil production*, Neurocomputing, vol. 175, Part B, pp. 980-989, 2016.
8. N.C. Chakra, K.Y. Song, D.N. Saraf, M.M. Gupta, *Production Forecasting of Petroleum Reservoir applying Higher-Order Neural Networks (HONN) with Limited Reservoir Data*, International Journal of Computer Applications, vol. 72, no.2, pp. 23-35, 2013, 10.5120/12466-8834
9. S.M. Al-Fattah, R.A. Startzman, *Predicting Natural Gas Production Using Artificial Neural Network*, In proc. of the SPE Hydrocarbon Economics and Evaluation Symposium, pp. 1-11, 2001, https://doi.org/10.2118/68593-MS
10. H.H. Nguyen, C.W. Chan, M. Wilson, *Prediction of oil well production using multi-neural networks*, In proc. of the Canadian Conference on Electrical and Computer Engineering, pp. 798-802, 2002.
11. K. Greff, R.K. Srivastava, J. Koutnék, B.R. Steunebrink, and J. Schmidhuber, *Lstm: A earch space Odyssey*, IEEE transactions on neural networks and learning systems, vol. 28, no. 10, pp. 2222-2232, 2017.
12. D. Palaz, M. Magimai-Doss, and R. Collobert, *Analysis of cnn-based speech recognition system using raw speech as inputℓ_1 function approximation*, In proc. of the Interspeech, pp. 11-15, 2015.
13. P. Malhotra, L. Vig, G. Shroff, P. Agarwal, *Long short term memory networks for anomaly detection in time series*, In proc. of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, pp. 89-94, 2015.
14. T.N. Sainath, O. Vinyals, A. Senior, H. Sak, *Convolutional, long short-term memory, fully connected deep neural networks*, In proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4580-4584, 2015.
15. R.M. Alguliyev, R.M. Aliguliyev, F.J. Abdullayeva, *PSO+K-means algorithm for anomaly detection in Big Data*, Statistics, Optimization And Information Computing, vol. 7, pp 348-359, 2019.
16. L. Deng, *A tutorial survey of architectures, algorithms, and applications for deep learning*, APSIPA Transactions on Signal and Information Processing, vol. 3, pp. 1-29, 2014.
17. Y. Imamverdiyev, F. Abdullayeva, *Deep Learning Method for Denial of Service Attack Detection Based on Restricted Boltzmann Machine*, Big Data, vol., 6, no. 2, pp. 159C169, 2018.
18. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol, *Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion* Journal of Machine Learning Research, vol. 11, no. Dec, pp. 3371-3408, 2010.
19. S., Hochreiter, J. Schmidhuber, *Long short-term memory*, Neural Computing, vol. 9, No. 8, pp. 1735-1780, 1997.
20. A. Krizhevsky, I. Sutskever, G.E. Hinton, *ImageNet classification with deep convolutional neural networks*, Advances in neural information processing systems, vol. 25, no. 2, pp. 1097-1105, 2012.
21. P. Zhong , Z. Gong, *A hybrid DBN and CRF model for spectral-spatial classification of hyperspectral images*, Statistics, optimization and information computing, vol. 5, pp 75-98, 2017.
22. L. Deng, D. Yu, *Deep learning: methods and applications*, Foundations and Trends in Signal Processing, vol. 7, no. 3-4, pp. 197-387, 2014.
23. A. Fatai, L. Jane, A. Abdulazeez, *Hybrid intelligent systems in petroleum reservoir characterization and modeling: the journey so far and the challenges ahead*, Journal of Petroleum Exploration and Production Technology, vol. 7, no 1, pp. 251-263, 2017.
24. S.D. Mohaghegh, *Recent Developments in Application of Artificial Intelligence in Petroleum Engineering*, Journal of Petroleum Technology, vol. 57, no. 04, pp. 86-91, 2005.
25. X. Ma, Z. Liu, *Predicting the oil production using the novel multivariate nonlinear model based on Arps decline model and kernel method*, Neural Computing and Applications, vol. 29, no 2, pp. 579-591, 2018, https://doi.org/10.1007/s00521-016-2721-x
26. G.J. Velazquez, C.J. Quintero, E.R. Gimenez, *Production monitoring using artificial intelligence*, SPE Intelligent Energy International, vol. 1, pp. 80-94, 2012.
27. A. Graves, A.R. Mohamed, G. Hinton, *Speech recognition with deep recurrent neural networks*, In proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645-6649, 2013.
28. L. Ding, W. Fang, H. Luo, P.E. Love, B. Zhong, X. Ouyang, *A deep hybrid learning model to detect unsafe behavior: Integrating convolution neural networks and long short-term memory*, Automation in construction, vol. 86, pp. 118-124, 2018.
29. G. Swapna, Kp. Soman, R. Vinayakumar, *Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals*, Procedia computer science, vol. 132, pp. 1253-1262, 2018.
30. S.L. Oh, E.Y. Ng, R.S. Tan, U.R. Acharya, *Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats*, Computers in biology and medicine, 2018, In press.
31. Krizhevsky A., Sutskever I., Hinton G.E., *Imagenet classification with deep convolutional neural networks*, Proc. of the 25th International Conference on Neural Information Processing Systems, pp. 1097-1105, 2012.

32. F.J. Morales, D. Roggen, *Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition*, Sensors, pp. 115-140, 2016, https://doi.org/10.3390/s16010115

33. P. Filonov, A. Lavrentyev, A. Vorontsov, *Multivariate industrial time series with cyber-attack simulation: fault detection using an LSTM-based predictive data model*, pp. 1-8, 2016, https://arxiv.org/pdf/1612.06676.pdf

34. N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*, Journal of Machine Learning Research, vol. 15, no 1, pp. 1929-1958, 2014.

35. J. Garcia, A. Levy, A. Tung, R.M. Yang, F.V.Kaynig, *Applying Deep Learning to Petroleum Well Data*, QRI project at TRiCAM, pp.1-9, https://www.academia.edu