Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

UDC 50.10

On manuscript rights

Aman Zhumekeshov

Forecasting Oil Production using DCA and LSTM Networks

THESIS

Presented in Partial Fulfillment for the
Degree of Master of Science in Computing Systems and Software
(degree code: 6M070400)
Department of Computer Sciences
Faculty of Engineering and Natural Sciences

Supervisor: **Assist. Prof. PhD Bogdanchikov Andrey**

Kaskelen, 2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Aman Zhumekeshov

Abstract

Natural resources are limited and very important in our industrial life and development. Oil is considered as the black gold and it is included in hundreds of industrial fields. Therefore, forecasting future oil production performance is an important aspect for oil industry. In this study, we proposed improvements to the existing deep learning model in order to overcome limitations associated with the original model. For evaluation purpose, proposed and original deep learning models were applied on a real case oil production data. The empirical results show that the proposed adjustments to the existing deep learning model achieves better forecasting accuracy.

Аңдатпа

Табиғи ресурстар шектеулі және біздің өндірістік өміріміздe және дамуымызда өте маңызды. Мұнай қара алтын болып саналады және ол жүздеген өнеркәсіпте пайдаланады. Сондықтан мұнай өндірісінің болашақтағы көрсеткіштерін болжау мұнай саласы үшін маңызды аспект болып табылады. Осы зерттеуде біз түпнұсқа модельмен байланысты шектеулерді еңсеру үшін қолданыстағы терең оқыту моделін жетілдіруді ұсындық. Мұнай өндірісінің нақты деректерінде бағалау мақсатында ұсынылған және түпнұсқалық терең оқыту модельдері қолданылды. Эмпирикалық нәтижелер қолданыстағы терең оқыту моделіне ұсынылған түзетулер болжамның дәлдігіне жететінін көрсетеді.

Аннотация

Природные ресурсы ограничены и очень важны в нашей промышленной жизни и развитии. Нефть считается черным золотом и используется в сотнях промышленных областях. Поэтому прогнозирование будущих показателей добычи нефти является важным аспектом для нефтяной промышленности. В этом исследовании мы предложили усовершенствования существующей модели глубокого обучения, чтобы преодолеть ограничения, связанные с исходной моделью. Для целей оценки предложенные и оригинальные модели глубокого обучения были применены на реальных данных добычи нефти. Эмпирические результаты показывают, что предлагаемые корректировки существующей модели глубокого обучения обеспечивают лучшую точность прогнозирования.

Acknowledgements

Going through this Master degree was a challenging experience for me, which included significant and constant guidance from many people. First of all, I am greatly thankfull to all members of Computer Sciences Department, especially to the Dean of faculty and my supervisor Dr. Andrey Bogdanchikov for his support and encouragement. I would also like to thank Prof. Abay Nussipbekov, whose enthusiasm in knowledge sharing helped me to build solid understanding of a forecasting approaches using Deep Learning techniques. And special thanks to program advisor Nazerke Sultanova, for her constant support and patient. Secondly, many thanks to my wife, without her persuasion I would have stopped these studies a long time ago. To conclude, I cannot forget to thank my friends in Almaty for unconditional support during these two academic years.

To my family

Contents

List of Figures

Nomenclature

ANN Artificial Neural Networks

$ARIMA$ Autoregressive Integrated Moving Average

DCA Decline curve analysis

$DLSTM$ Deep Long-Short Term Memory

GC Gamma Classifier

RNN Recurrent Neural Networks

SVM Support Vector Machines

TS Time Series

TSF Time Series Forecasting

1. Introduction

1.1 Motivation

Since ancient times, energy has been the most important factor determining human life and the development of civilization. The history of energy from the centuries when a man took possession of fire, the energy of rivers, wind, and to this day reflects a constant search, great discoveries, accumulation and transfer of knowledge from generation to generation, the most important achievements in the field of knowledge of the laws of nature.

Until the XIX century, the main energy resource on the planet was wood. At the turn of the XIX – XX centuries. internal combustion engines were invented. Their introduction into practice led to the rapid development of the oil industry.

Oil and natural gases have been known to mankind since ancient times. Scientists have found that more than 500 thousand years ago, oil was already discovered on the shores of the Caspian Sea, and 6 thousand years BC there was an exit to the surface of the natural gas in the Caucasus and Central Asia.

Archaeological excavations have shown that on the banks of the Euphrates, oil was extracted 6–4 thousand years BC. It has been used for various purposes, including medicine. The ancient Egyptians used asphalt for embalming. They mined it, according to the ancient Greek historian and geographer Strabo (63 BC), mainly off the coast of the Dead Sea. Oil bitumen was used for the preparation of mortars and as a lubricant. Oil was an integral part of the incendiary, which went down in history under the name "Greek fire". Among the peoples inhabiting the southern shores of the Caspian Sea, oil has long been used to illuminate homes. This is evidenced by the ancient Roman historian Plutarch, who described the campaigns of Alexander the Great.

Until the middle of the XIX century oil was produced in small quantities, mainly from shallow wells near its natural outlets to the surface of the earth. Since the second half of the XIX century demand for oil began to increase due to the widespread use of steam engines and the development of industry, which required large quantities of light sources. The introduction of well drilling in the late 60s of the XIX century is considered the beginning of the emergence of the oil and gas industry.

Now oil and natural gas are the basis of the global fuel and energy balance. Petroleum products and natural gases are widely used in all sectors of industry, agriculture, transport and in everyday life. It is difficult to list all the areas of application of oil and natural gas, so their role in the life of modern society is multifaceted. For a long time, mankind has considered gold as an enduring value. All material goods of special significance are compared with it. For example, cotton called "white gold", oil called "black gold", and natural gas called "blue gold". But life itself shows that oil and natural gases in the modern world mean much more than gold of the highest standard.

At school:	rulers, crayons, ink and cartridges, glue, coverings on books, binders...
For your health:	coatings for pills, binding agent for creams, disposable syringes...
In the home:	contact lenses, cosmetics, clothing, fabrics, nail polish, deodorants, shampoo, paint, upholstery and carpets, detergents for washing up and laundry, dry-cleaning fluid...
Out shopping:	shopping bags, credit cards, egg cartons, plastic milk bottles...
While cooking:	non-stick pans, cling film, storage containers...
For building:	roofing tiles, pipes, insulating material, paint...
On the move:	petrol and diesel for cars and lorries, emergency services and trains, asphalt road surfaces...
In the office:	computer hardware, phones and faxes, diskettes, pens, chairs, printing ink...
At your leisure:	CDs, videos, cassette tapes, camera film, artists' paint, bicycle handlebar grips, tyres, crash helmets, football boots, trainers, shin pads, windsurfers, roller blades...
Garden:	fertilizers, pesticides, garden furniture...

Table 1.1: Some examples of what we owe to oil, every day of our lives

Similarly, oil and gas industry play extremely important role in the economy Kazakhstan. The budget of the state is largely dependent on extraction taxes and revenues coming from the export of oil and gas. As can be seen from the table below, all together, in 2018 oil and gas companies in Kazakhstan paid 87% of all taxes from the country's top 30 taxplayers [1].

Sectors	Paid Taxes, Billion Tenge	Share in total tax revenue, %
Oil & Gas Production	4195.8	87.0%
Metallurgy industry	211.3	4.4%
Tobacco manufacturing	159.4	3.3%
Transportation	72.5	1.5%
Financial sector	51.9	1.1%
Communication	40.2	0.8%
Mining industry	31.6	0.7%
Construction	30.6	0.6%
Oil refining	30.1	0.6%
TOTAL	**4823.7**	**100%**

Table 1.2: Tax revenues to the consolidated budget of the Republic of Kazakhstan by industry in 2017

All these dependencies on oil and its finite nature, pose complex problem of future oil production patterns. Historical and most common approach in solving this problem is Decline curve analysis (DCA). By identifying decline curves, DCA extrapolates past production in order to estimate expected production. Extrapolation of production history has been considered the most accurate method to estimate remaining reserves. The basic assumption in this method is that whatever causes controlled the trend of a curve in the past will continue to govern its trend in the future in a uniform manner. Therefore, if the conditions affecting the rate of production are changed, the curve will not be regular and will not exhibit exponential behavior during depletion. Similarly, in absence of stabilized production trends the technique cannot be expected to give reliable results.

1.2 Aims and Objectives

The aim of present research study is:
Analyzing and comparing different methods of regression models to fit historical oil production data for further forecasting into the future.
In order to achieve the aim:

- Real case data-set from oil field located in western part of Kazakhstan is analyzed.
- Existing reported models on Time Series Forecasting are investigated for advantages and limitations.
- Widely used technique Decline Curve Analysis (DCA) is explained and set as reference method.
- Proposed model is explained and applied on our data-set using the Keras, Python Deep Learning library.

1.3 Thesis Outline

The first chapter is Introduction chapter. It is this one that you are currently reading. It gives insight into the problem statement. In Chapter 2, dataset considered in this work is explained. In Chapter 3 we review related work, and in Chapter 4 we formulate application of one of the existing methods to solve the problem. Chapter 5 outlines another widely known mathematical method Decline Curve Analysis and Chapter 6 describe proposed model DCA-DLSTM. In Conclusion chapter we provide our conclusion.

2. Dataset Description

The Dataset gives details about the crude oil production from the oil field located in the western part of Kazakhstan from May 2013 to September 2019, with 2320 points (these points represent the number of days). It is a univariate time series data which entails the average daily production of crude oil per each well. The oil production is measured in barrels per day.

The trend in the oil production in the mentioned time period is given below.

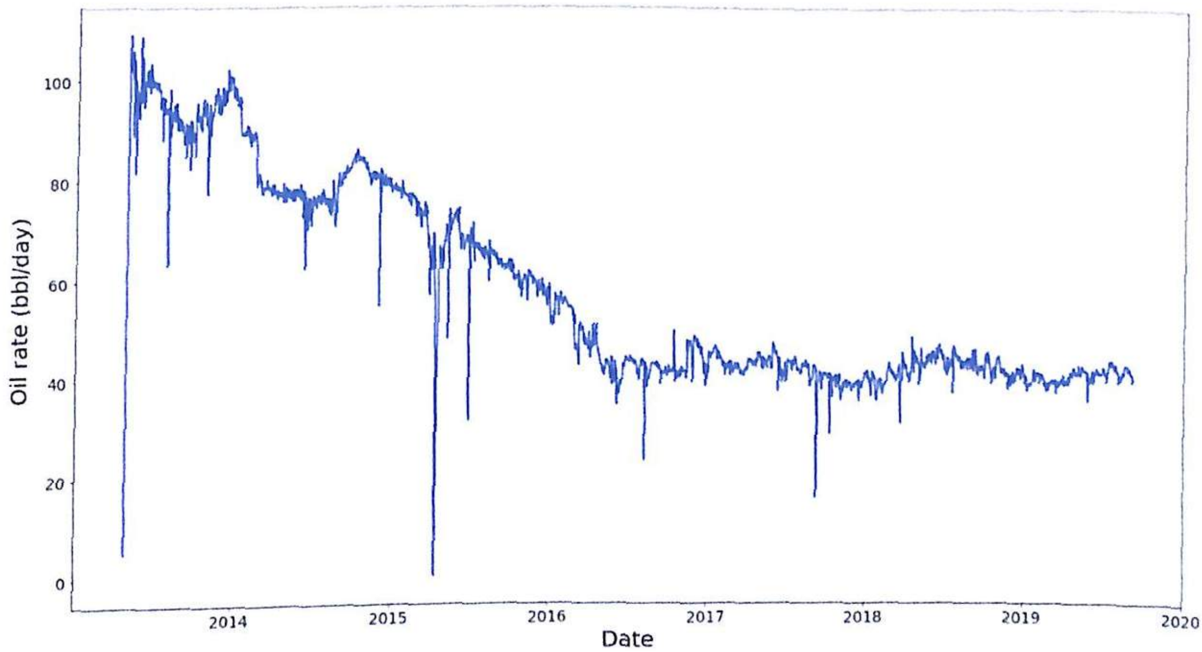

Figure 2.1: Timeseries data on oil production

Note that the curve contains some noise, defects and anomalies. It is also non-stationary and exhibits declining trend over production period. This implies that the statistical characteristics of the data, such as frequency, variance, and mean, undergo alternation over the time.

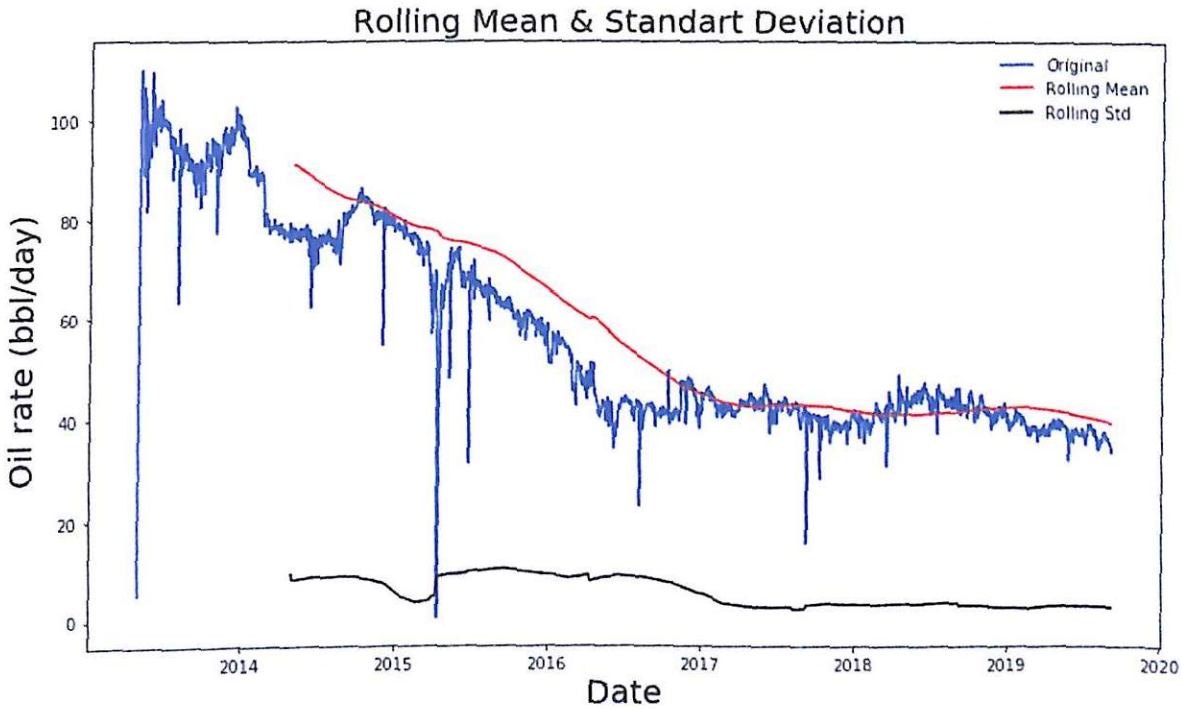

Figure 2.2: Rolling Mean & Standard Deviation of the data

Since, oil companies and countries are interested in long-term oil production, data series were grouped into the monthly data frequency with average oil production within each month. This in turn will allow for more complex network architecture, while still keeping optimal running time for training the model.

3. Related Work

Time series data is a sequentially measured data at some (often equal) time intervals. The most valuable information that can be extracted from TS is prediction of its future values. A forecast is always a probability, but at the same time it is a scientific judgment about the possible future states of trend or phenomenon; there is always a fraction of error and miss-match. In the best case, this error is random, in the worst - systematic. In order to build a forecast, a mathematical model is needed that sets the rule for its calculation, and this model will be stochastic, since it will contains uncertainty in the form of an error. The main feature of forecasting time series in comparison with other tasks of machine learning is that the prediction of the target variable is usually based not on other features, but on the values of itself for the previous time period. Thus, for forecasting time series, almost exactly the same models are used as for any other prediction tasks, but with some modifications designed to take into account their coherence. It is obvious that a successful time series forecasting depends on an appropriate model fitting. A lot of efforts have been done by researchers over many years for the development of efficient models to improve the forecasting accuracy. As a result, various important time series forecasting models have been evolved, which can be grouped into three major groups.

First group involves Statistical Models such as: Moving Average, Exponential Smoothing, or derivative from these models [2]. An inherent advantage of these models is that a few model parameters are required for describing time series, therefore, less effort and timing.

The second group of models includes using different types of Artificial Neural Networks. Application of ANN does not require specification of the model form or any prior assumption about the statistical distribution of the data. Instead, the final model is generated adaptively based on the features presented the historical section of the data.

The last group's models are based on the concept of using Support Vector Machines. Important characteristic of Support Vector Machines is the training process which is exactly the same as solving a linearly constrained quadratic programming problem. So, contrary to other neural networks' training processes, solution's obtained from Support Vector Machines are always unique and globally optimal. However a major drawback of Support Vector Machines is that when the training dataset size is too large, it will have an enormous amount of computation which leads to increased time complexity of the solution [3].

Below is the list of few studies that researches have conducted in the application of oil production forecasting:

- Ayeni and Pilat [4] described application of ARIMA model for monthly time series oil production data. ARIMA is an extension of AutoregRessive Moving-Average Model for non-stationary time series data, which can be converted into the stationary by taking the difference of some order from the original time series. It is by far one of the most popular forecasting tool in many areas of finance and science. It uses linear combination of past observations and some random errors while forecasting future behavior of time series data. Authors applied this tool on historical oil production data coming from twelve wells. Different ARIMA model's parameters were obtained for each well while fitting to the original data by minimizing residual error. According to the authors *"ARIMA models provided better results than decline curve analysis. Moreover, ARIMA obtained good results when decline curve analysis failed (e.g. for waterdrive reservoirs)"*.

- Holdaway [5] presented application of exponential smoothing techniques to oil production forecasting in the Oilfield Production Forecasting toolkit developed for Saudi ARAMCO National Oil Company. As in the case of regression-based techniques, smoothing shows good short-term accuracy but is only valid if the error terms are assumed to be random.

- Alaa and Mostafa [6] proposed a deep learning model which can overcome the limitations of traditional forecasting approaches and show accurate predictions. The proposed approach is a deep long-short term memory (DLSTM) architecture, as an extension of the traditional recurrent neural network. Genetic algorithm is applied in order to optimally configure DLSTM's optimum architecture. For evaluation purpose, two case studies from the petroleum industry domain are carried out using the production data of two actual oilfields. Toward a reasonable assessment, results of the proposed method is compared with several standard methods, either statistical or soft computing. Using different measurement criteria, the empirical results show that the proposed DLSTM model outperforms other standard approaches.

- Aizenberg et al. [7] examined application of Multilayer Neural Network with Multi-Valued Neurons (MLMVN) to long-term monthly TS production data from a real oilfield example. As usual, predictive modeling was divided into regression and pattern classification. While regression models are based on the analysis of relationships between variables and trends in order to make predictions about continuous variables, the task of pattern classification is to assign discrete class labels to particular observations as outcomes of a prediction. The distinctive feature of the forecasting method, is that MLMVN-based prediction combines the features of the both approaches. While externally the neural network constructs a regression-like model of the time series, internally classification techniques are used. Both multivariate and univariate models were developed. The advantage of the proposed forecasting models as compared to conventional forecasting methods is that prediction allows local variation rather than smooth curve projection.As the experimental results showed, both models can be used efficiently and achieve results that are significantly better in terms of the anticipated interval and allow for long-term predictions. Choosing embedding dimensions from multivariate time series data, however, is a difficult and yet open problem that requires additional research effort.

- Frausto-Solís et al. [8] proposed a new method "SAM-oil" to forecast monthly oil production data. This work can be considered as an alternative or supplement to traditional techniques of petroleum engineering. In particular, SAM-oil can be considered as a method which supplements decline curve analysis, since the obtained models include an exponential decline curve as that obtained with traditional DCA. Accuracy of SAM-oil method was determined in the sense of monthly oil production, since its good performance on forecasting time series with different levels of difficulty. Authors compared the accuracy of "SAM-oil" with ARIMA, neural networks, and some models from the exponential smoothing. They claim that "SAM-oil" is reliable method in oil production forecasting.

- Fargana and Yadigar [9] 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.

- Leonid et al. [10] compared traditional ANN forecasting to the Gamma Classifier (GC). The main conclusion was that the GC can be successfully applied to the task of time series prediction, though it is not the kind of tasks envisioned while designing and developing the GC model.

- Sheremetov et al. [11] analyzed several aspects of oil production forecasting with NARX networks: i) the applicability to the univariate and multivariate forecasting, ii) different topologies of the networks, and iii) the application of clustering techniques to improve forecasting results. The described case studies come from the naturally fractured reservoir with primary production and under gas injection for pressure maintenance.

- Chithra et al. [12] reported use of another type of ANN, higher-order neural networks (HONN), to forecast production of water, oil and gas. That type of neural networks has a good capacity to capture nonlinear relationships between inputs and weights of the network. Two case studies illustrate the behavior of the proposed model with i) only one dynamic parameter data, oil production data, and ii) three dynamic parameter data, oil, gas and water production data used for forecasting. The authors report similar accuracy results but it should be noted that HONN was used as a one-step ahead predictor, which is impractical in most of the oil forecasting applications.

In general, workflow of all above proposed models is similar in a way that, all of them use predict one-step ahead forecast, which is given input Xt at time t, the time series value at the next time step (t+1) is predicted, $\hat{Y}$t. Finally, loss function, which is $\sum$ F($\hat{Y}$t-Yt), is optimized. In the next chapter, method proposed by Alaa and Mostafa will be reduplicated and applied on our dataset for further comparison of Deep learning method to other methods.

4. DLSTM Model

This section, shows application of a deep long-short term memory (DLSTM) architecture proposed by Alaa and Mostafa, who shared the codes used in their experiment in github (`https://github.com/DeepWolf90/DLSTM2`). Their proposed model will be applied both on original dataset they used and our data for further comparison and discussion.

Note: original data was collected from Cambay Basin oil field in India, with real production data through six years from 2004 to 2009, i.e. about 63 months.

4.1 Experimental Setting

As it was shown in Data Description section, these time series data on oil production is also non-stationary data, in fact, it exhibits a specific decline trend. Stationary data is easier to model and will very likely result in more skillful forecasts. In the current prepossessing step, we removed the trend property in the data, whether it is increasing or decreasing trend. Later, it will be added back to forecasts in order to return the forecasting problem into the original scale and calculate a comparable error score.

There are two standard ways to remove the trend; differencing (taking the difference Y_t - Y_{t-1}) and detrending (removing the trend). If the non-stationary process is a random walk with or without a drift, it is transformed to stationary process by differencing. On the other hand, if the time series data analyzed exhibits a deterministic trend, the spurious results can be avoided by detrending [13].

Note: In this work,author suggested to use differencing, however on the later sections both methods will be discussed and compared.

```
series = read_csv('cumulative_oil.csv')
raw_values = series.values
diff = []
for i in range(1, len(raw_values)):
    value = raw_values[i] - raw_values[i-1]
    diff.append(value)
diff = Series(diff)
```

Listing 4.1: Transform data to be stationary

Then, the time series were divided into input (x) and output (y) using lag time of 12 months (1-year period).

```
lag_time= 12
dataset = diff.values
dataset = np.insert(dataset, [0]*lag_time, 0)
batches = []
for i in range(len(dataset) - lag_time):
    a = dataset[i:(i+lag_time+1)]
    batches.append(a)
dataset = np.array(batches)
```

Listing 4.2: Convert an array of values into a dataset matrix

Additionally, data is split into training (80%) / testing (20%) sets and transformed into the scale of the activation function used by the network, which is hyperbolic tangent (tanh) with output values between -1 and 1. Later, the scaled data transformed back in order to return the forecasting problem into the original scale.

```
scaler = MinMaxScaler(feature_range=(-1, 1))
scaler = scaler.fit(dataset)
dataset_scaled = scaler.transform(dataset)
train_scaled = scaler.transform(train)
test_scaled = scaler.transform(test)
```

Listing 4.3: Transform the scale of the data

In the proposed method, three LSTM [14] blocks were stacked one after another in a deep recurrent network fashion to combine the advantage of a single LSTM layer, as shown in the Figure 4.1.

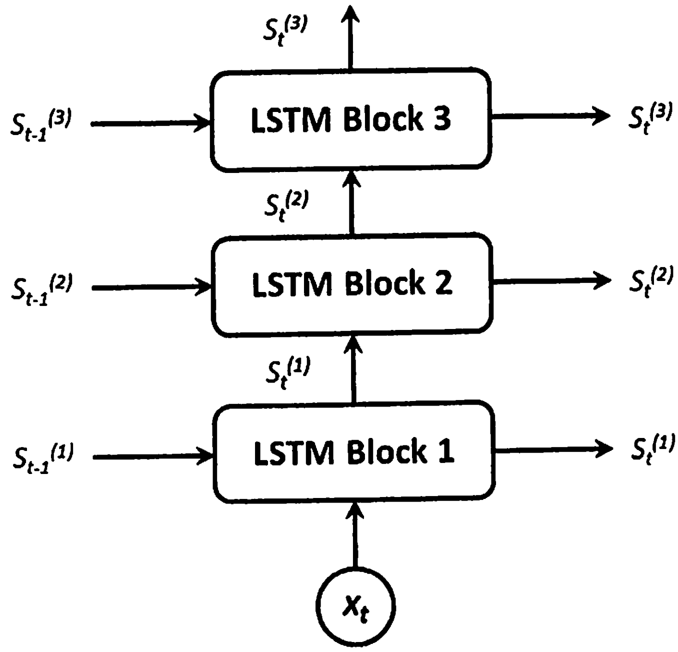

Figure 4.1: The architecture of DLSTM recurrent network

```
model = Sequential()
model.add(LSTM(32, batch_input_shape=(1, lag_time, 1), stateful=True,
    return_sequences = True))
model.add(Dropout(0.1))
model.add(LSTM(16, batch_input_shape=(1, lag_time, 1), stateful=True,
    return_sequences = True))
model.add(Dropout(0.1))
model.add(LSTM(64, batch_input_shape=(1, lag_time, 1), stateful=True))
model.add(Dropout(0.1))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
```

Listing 4.4: Deep Gated Recurrent Unit

4.2 Experimental Results

As can be seen on the Figure 4.2, predicted oil production values modelled by the proposed DLSTM method match quite well to original data both in train data set and test data set. Although, authors tried to evaluate performance of the model on unseen data (test dataset), it still uses test dataset with lag time gap while predicting next value. For fair evaluation, predictions should have been done on last data range (with lag time lenght) of train data, and further values for input should have been taken from newly predicted values.

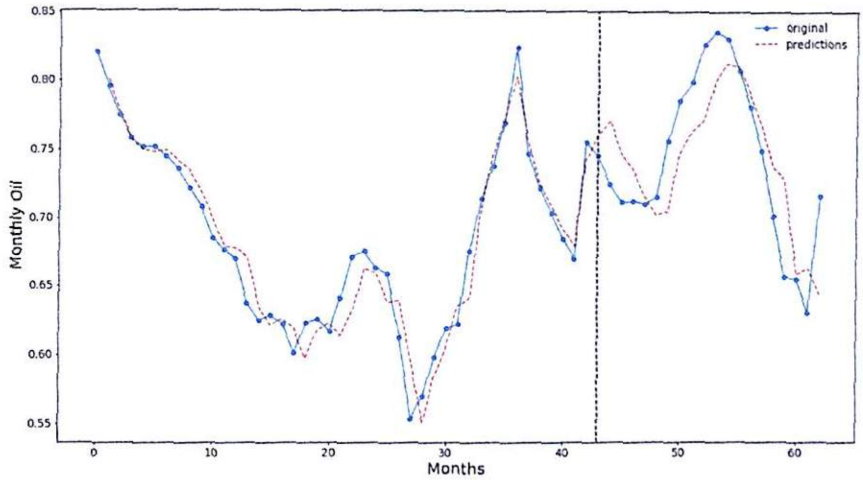

Figure 4.2: Production data v.s. prediction using DLSTM (paper data)

Authors of this work also compared the proposed DLSTM model with different reference models that vary from statistical methods, machine learning methods, and hybrid (statistical and machine learning) methods; ARIMA, RNN, DGRU, NEA, HONN.

Forecasting Model	RMSE (root mean square error)	RMSPE (root mean square percentage error)
ARIMA	0.027	3.773
NEA	-	4.221
DLSTM	**0.025**	**3.496**
RNN	0.027	3.731
DGRU	0.028	3.991
HONN	0.035	3.459

Table 4.1: Comparison of proposed to reference models

Final comparison Table 4.1, shows that proposed DLSTM model outperformed its counterparts.

Similarly, if the same model applied to the data-set considered in this work, very nice match can be achieved (Figure 4.3). This is, indeed, what should be expected as the proposed model use one-step ahead forecast (Figure 4.4). Estimation of model performance using this approach would be optimistic and might lead to biased decisions. In applied deep learning, whole data-set is usually split into a train and test sets: the training set used to prepare the model and the test set used to evaluate it. However, these method cannot be directly used with time series data, because assumption that there is no relationship between the observations is not valid anymore [15].

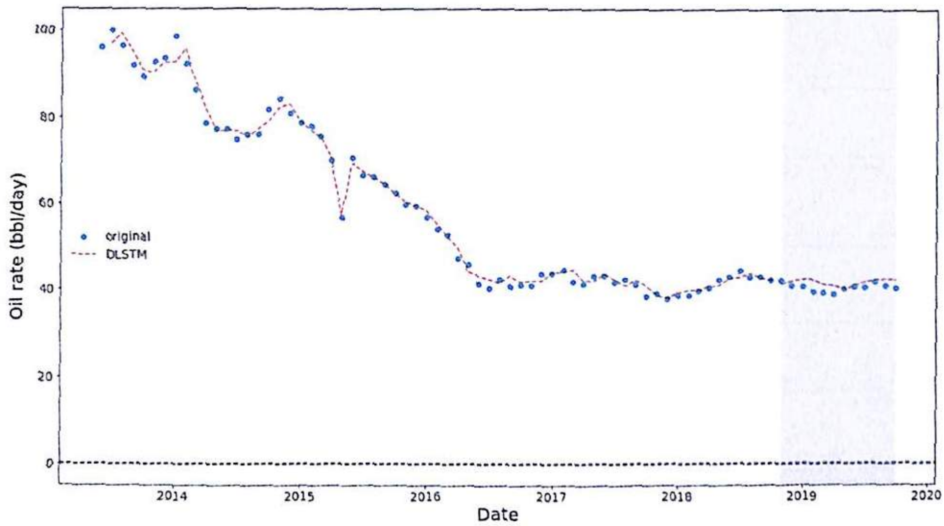

Figure 4.3: Production data v.s. prediction using DLSTM (case data)

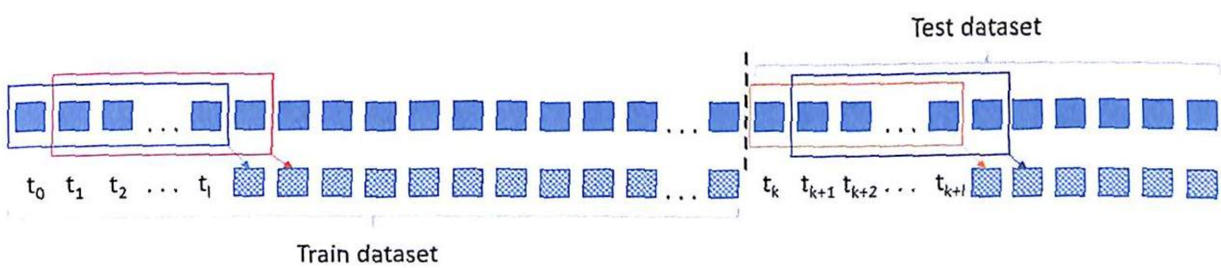

Figure 4.4: Proposed evaluation by Alaa and Mostafa

Alternatively, in order to make fair evaluation, it is possible to split whole data-set into a train and test sets. Use train set to prepare the model and ask the model to make future predictions for test set period (Figure 4.5). The evaluation of these predictions will provide a good proxy for the model performance.

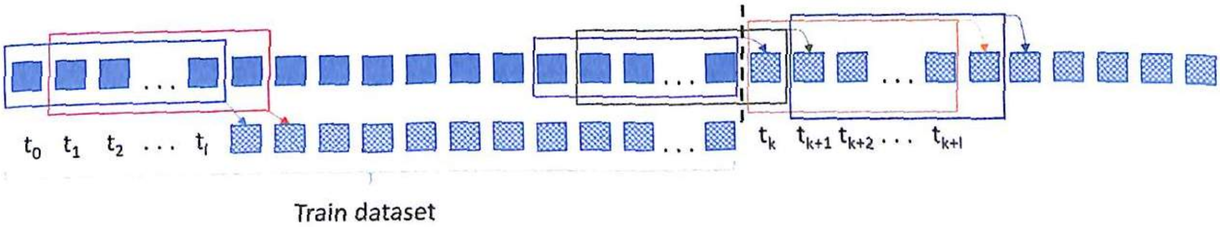

Figure 4.5: Suggested fair evaluation

5. DCA Model

Decline curve analysis (DCA) is a empirical technique used for analyzing trends in the production data from the oil and gas wells in order to forecast future performance of the wells. Oil and gas production flow rates decline as a function of time; due to the loss of reservoir pressure, and/or changing relative volumes of the produced fluids. Fitting a line through the performance history and assuming that similar trend will continue in future are the main concept and assumption in the DCA procedure. The technique is not derived from fundamental theory of rock mechanics but is based on empirical observation of production decline [16].

5.1 DCA History

J.J. Arps was an American geologist who collected these ideas into a comprehensive set of equations defining the exponential, hyperbolic and harmonic declines. Later his work was further investigated and modified to include special cases by many other researches. Following section gives a historical perspective of work done on the subject.

- Arps 1945 and 1956.
- Brons 1963 and Fetkovitch 1983 applied constant pressure solution to diffusivity equation and demonstrated that exponential decline curve actually reflects single phase, incompressible fluid production from a closed reservoir. DCA is more than a empirical curve fit.
- Fetkovitch 1980 and 1983 developed set of type curves to enhance application of DCA.
- Doublet and Blasingame 1995 developed theoretical basis for combining

transient and boundary dominated flow for the pressure transient solution to the diffusivity equation.

5.2 DCA Today

The major application of DCA in the industry today is still based on equations and curves described by Arps. Arps applied the equation of Hyperbola to define three general equations to model production declines.

In order to locate a hyperbola in space one must know the following three variables:

1. The starting point on Y axis, q_i, initial rate.

2. Initial decline rate, D_i

3. The degree of curvature of the line, b.

Arps did not provide physical reasons for the three types of decline. He only indicated that exponential decline (b=0) is most common and that the coefficient b generally ranges from 0 to 0.5.

$$q_t = \frac{q_i}{(1 + b \times d \times t)^{1/b}} \tag{5.1}$$

Arp's Equation 5.1 for General Decline in a Well.

<u>Definition of b</u>

Variable 'b' has no units and is generally defined as hyperbolic constant. Usually, 'b' range between 0 to 1 in the context of DCA for oil and gas wells. From Equation 5.1 it is clear that a high value of b (close to 1) has a dominant impact on curve form q vs. t when t is high, which causes and preserves the form of the curve to be essentially flat throughout this period. The short-term shape for the curve is not largely affected by the value of b for a given set of values for q and b, but the long-term shape is. This means that in short term all decline curves; exponential, hyperbolic and harmonic will generate very close results. Due to this reason, it is extremely difficult to determine the value of b at early stages of production. The situation is further impaired if the data quality is noisy (which is often the case), making it possible to fit the same dataset with a wide range of b values. Since the value of b has a large impact in the late period, this will lead to different estimates of ultimate recovery. Reliability in estimating b rise with longer production period. There is a large number of physical events and processes integrated into the b value, which were analyzed and described in many papers.

5.3 Application of DCA

As it was described above, general Decline Curve Analysis (DCA) will be set up fitting the historical oil production data and forecasting future values.

Notice that, in case of hyperbolic decline constant (b) approaching zero, according to Taylor series, Equation 5.1 can be rewritten as:

$$q_t = \frac{q_i}{e^{D \times t}} \tag{5.2}$$

Equation 5.2 is known as Exponential decline, which is a special case of general DCA.

```
def forecast(time, args):
    qi = args[0]; Di = args[1]; b = args[2]
    if b > 1e-6:
        q = qi / (1 + Di * b * time) ** (1 / b)
    else:
        q = qi / np.exp(Di * time)
    q = np.array(q, dtype='float')
    return q
```

Listing 5.1: Fit function applied to the time array

For the optimization purposes, optimize.least_squares package from SciPy library is used, which is the most commonly used optimization algorithms. Given the residuals f(x) (an m-dimensional real function of n real variables) and the loss function rho (a scalar function), least_squares finds a local minimum of the cost function F(x).

Five loss functions are available within the package:

- 'linear' rho(z) = z. Gives a standard least-squares problem.
- 'soft_l1' rho(z) = $2 \cdot ((1 + z)^{0.5} - 1)$. The smooth approximation of absolute loss.

- 'huber' rho(z) = z if z <= 1 else $2 \cdot z^{0.5}$ - 1. Works similarly to 'soft_l1'.
- 'cauchy' rho(z) = ln(1 + z). Severely weakens outliers influence.
- 'arctan' rho(z) = arctan(z). Limits a maximum loss on a single residual.

Four of these loss functions (linear, huber, cauchy, arctan) will be applied to the dataset and compared results.

Also, boundaries must be set on our parameters, as per definition of Decline Curve Analysis initial nominal decline rate (d) cannot be negative and hyperbolic decline constant (b) must be within the range of [0; 1]. Initial production rate (q_i) also bounded to the reasonable range.

```
Linear_dca = sc.optimize.least_squares(
                fun=lambda parms, time, rate: rate - forecast(time, parms),
                bounds=([10, 1e-6, 0], [1000, 1e6, 1]),
                x0=np.asarray([100, .001, 1.]),
                args=(fit_time, fit_rate),
                loss='linear')
```

Listing 5.2: Example of optimization using linear loss function

	q_i	D	b
Linear	105.64	23.81	0.82
Huber	104.44	23.99	1.00
Cauchy	101.54	22.41	1.00
Arctan	99.32	20.66	0.36

Table 5.1: Fit Parameters

As can be seen from the Figure 5.1, all models provided similar results except for arctan loss function. But still, all of them break down in fitting the last year historical production data, which is the most crucial in forecasting future values. This is due to changing conditions affecting the rate of production. Plots colored by point-by-point costs illustrate the sections fitted most by each model.

5.4 Limitations of DCA

The basic assumption in decline curve analysis is that whatever controls the trend of a production curve in the past will continue to govern its future trend in a similar

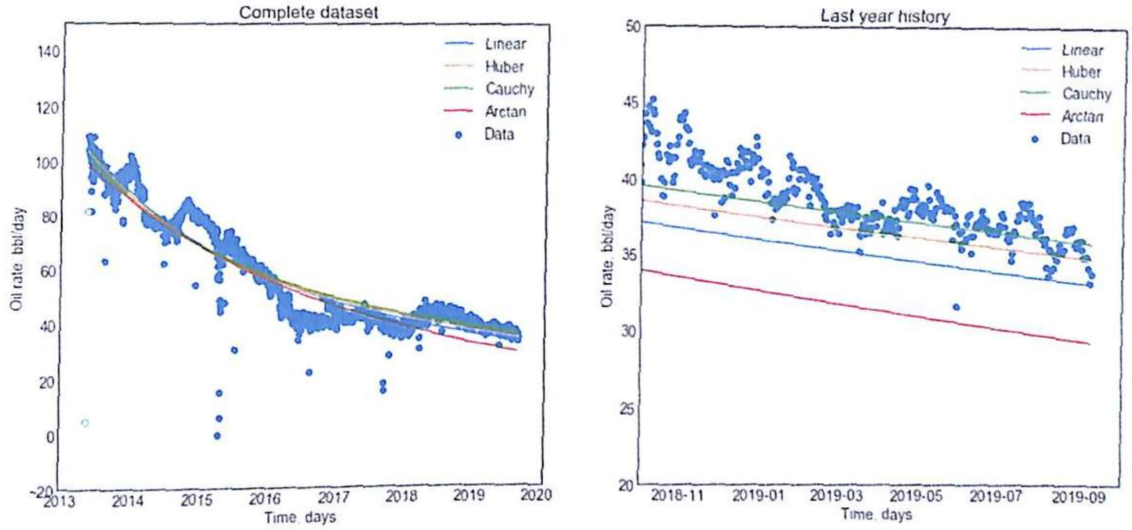

Figure 5.1: Fitted Models

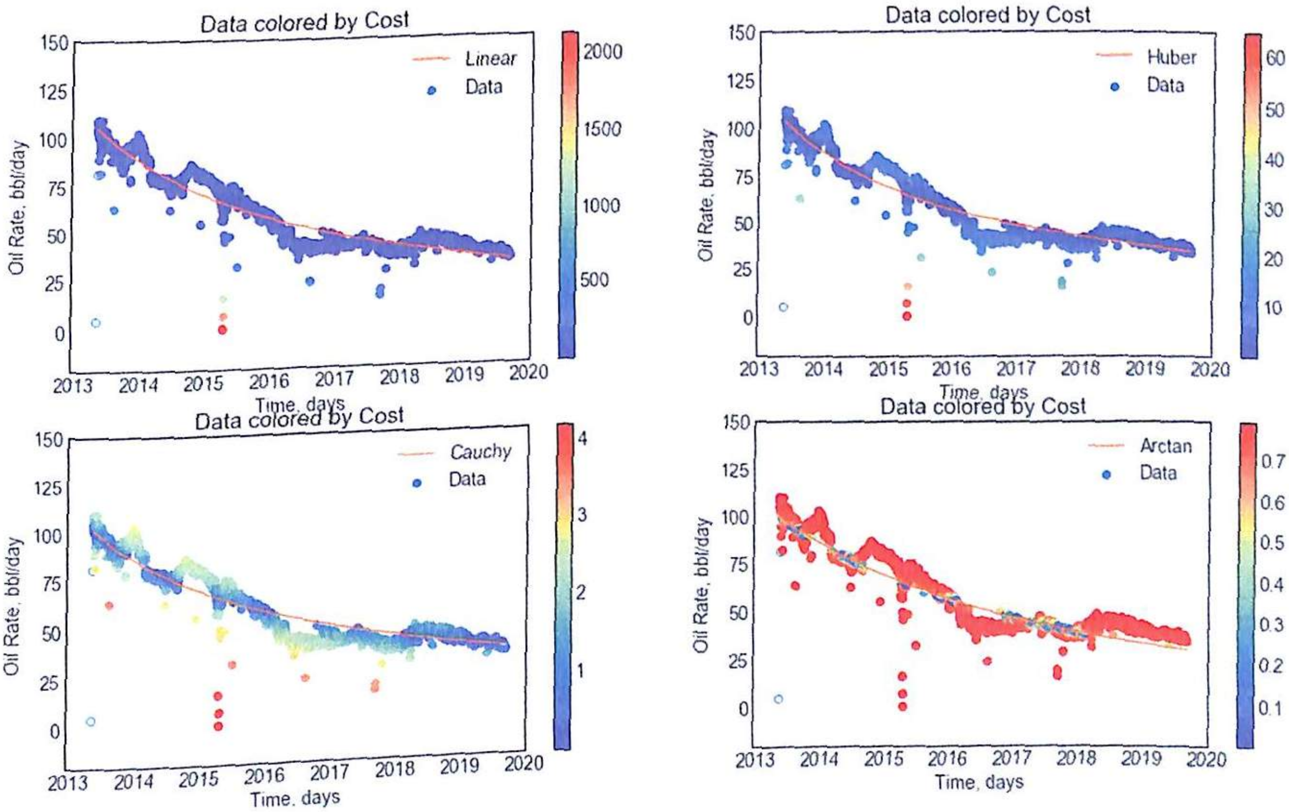

Figure 5.2: Point-by-point Costs

and uniform manner. Therefore, if the conditions affecting the rate of production are changed, the curve will *not* be regular and will not exhibit exponential behavior during depletion. Similarly, if there is no stabilized production period, extrapolated trend from this technique cannot give reliable results.

6. DCA-DLSTM Model

Let's compare results from DLSTM and DCA seperatly. In order to make fair comparison on models evaluation, both models were trained on train dataset. However, future predictions using DLSTM model were performed using last lag time values of train set and already predicted values, while DCA model were just extented into the same future period length, which is as long as train dataset lenght.

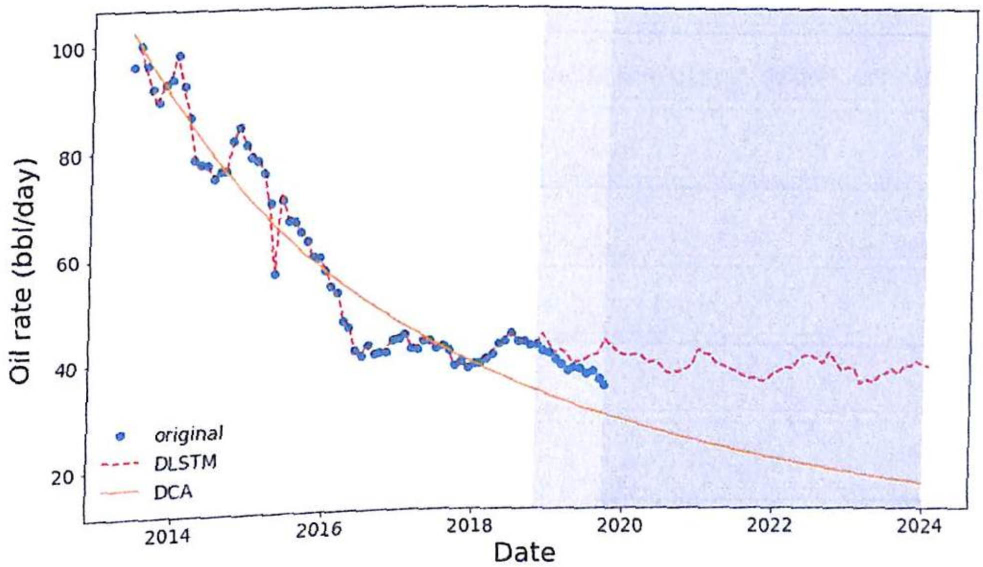

Figure 6.1: DLSTM v.s. DCA model's results

As can be seen from the Figure 6.1, DLSTM model matched historical production data much better than DCA. However, future predictions from DLSM model does not represent general trend of conventional decline curve, which suggest that oil production with time should decline over the time, mainly due to the pressure drop in the reservoir.

6.1 The Proposed Model

The main reason why DLSTM Model fails in following decline trend is the fact that during removing the trend of historical production curve difference of consecutive data was applied. This approach does not constrain the general curve to decline over the time.

In the proposed DCA-DLSTM Model constrain to decline is achieved by taking the difference between production data and general exponential decline, and only then apply DLSTM model. Therefore, deviation of historical production data from overall decline is actually modelled (Figure 6.2).

```
parm_init = np.asarray([90, .015, .9])
bounds=([10, 1e-6, 0], [1000, 1e6, 1])
raw_time = np.linspace(1, len(raw_values), len(raw_values))
popt, pcov = sc.optimize.curve_fit(forecast, raw_time, raw_values, p0=
    parm_init, bounds=bounds)
yfit = forecast(raw_time, *popt)
dataset = raw_values - yfit
```

Listing 6.1: Transform data to be stationary using decline

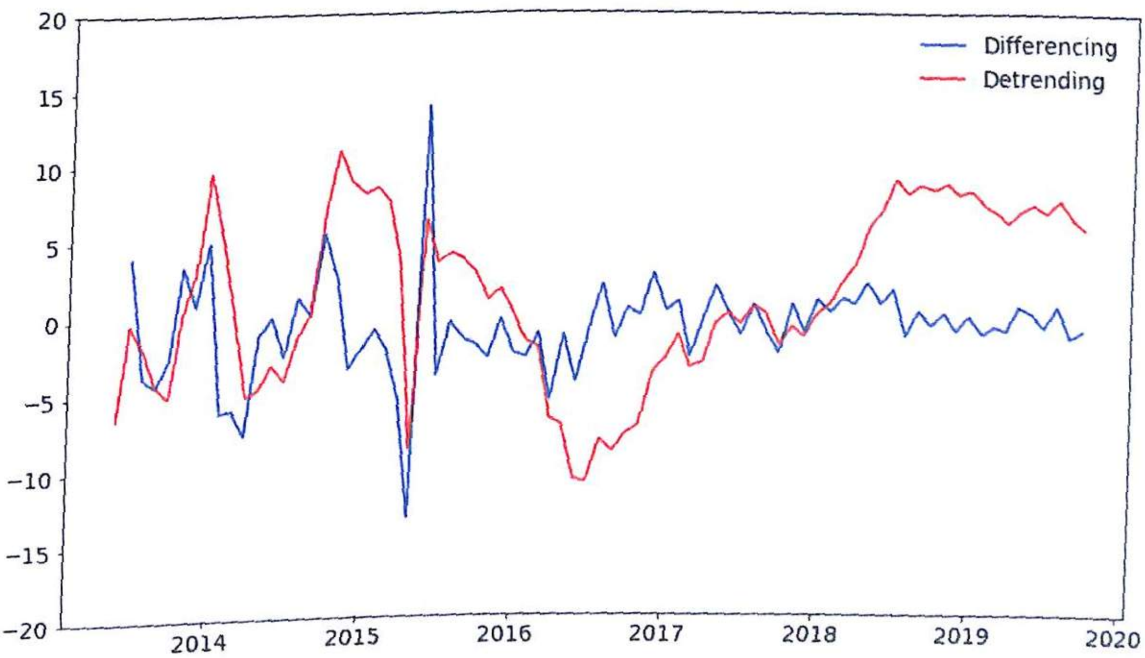

Figure 6.2: Stationary Oil Production Data

After de-trending historical production data and repeating steps in discussed DLSTM Model above (skipping Listing 4.1), it is possible to achieve both good match on train and test data-sets. At the same time, future predictions from DCA-DLSM model follow general trend of conventional decline (Figure 6.3).

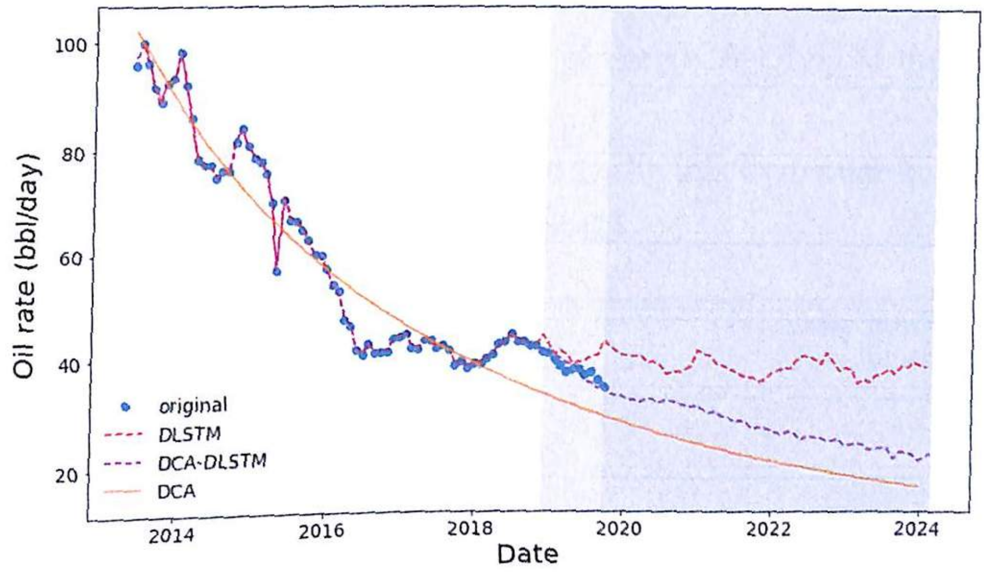

Figure 6.3: Proposed DCA-DLSTM model result

6.2 Experimental Results

To control the learning process and evaluation accuracy of the proposed method, the root mean square error (RMSE) was used. The RMSE is a frequently used measure of the differences between values (sample or population values) predicted by a model and the values observed [17]. It represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences (Equation 6.1).

$$E_{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2} \tag{6.1}$$

As seen from Table 6.1, the results of the proposed DCA-DLSTM model outperforms its counterparts. Thus, during prediction the RMSE value of the DCA-DLSTM model is obtained 0.53, but in the DLSTM model this value reached to 3.65 and in the DCA model to 5.80. As seen from here, in the assessment of

Forecasting Model	RMSE
DCA	5.80
DLSTM	3.65
DCA-DLSTM	**1.29**

Table 6.1: Forecasting accuracy

oil production forecasting, the effectiveness of the DCA-DLSTM model is better than other algorithms.

To better illustrate the results of Table 6.1, the loss dynamics of DLSTM and DCA-DLSTM models are depicted in Figure 6.4.

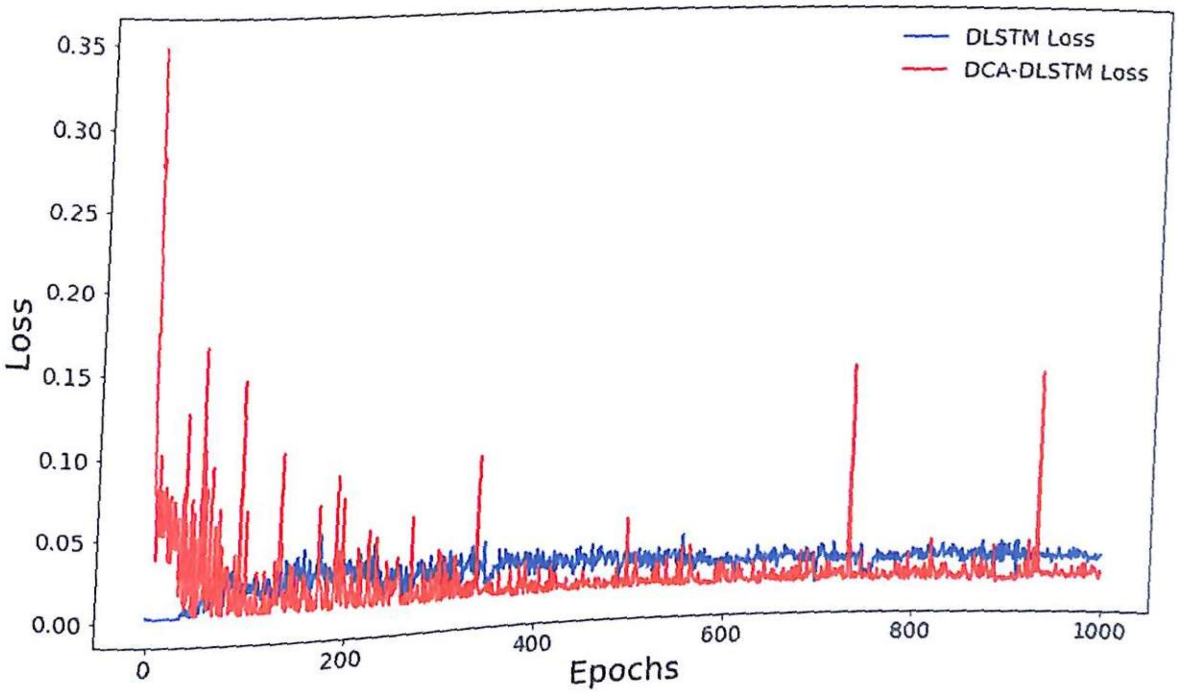

Figure 6.4: Loss dynamics by number of epochs

7. Conclusion

In this work we discussed traditional techniques of oil production forecasting methods (DCA), together with existing experiments conducted using recurrent neural networks (LSTM). While DCA model can provide and predict general trend of the time series data, it fails in capturing any abnormalities and deviations. On the other hand, Recurrent Neural Network method – DLSTM can capture any deviation but limited in controlling overall decline trend of time series data. Based on the limitations of both methods above, we proposed combined model DCA-DLSTM, which includes detrending of monthly oil production data using decline curve analysis and stacked three LSTM layers that learns certain properties from historical dataset. The results show that the accurate prediction and learning performance of proposed model outperformed its counterparts. Further enhancements include development of the Multivariate forecasting model, when a TS member to predict depends on the members of another TS.

A. Appendix A

A.1 Complete Code of the Proposed Model

```
# Import Libraries

import numpy as np
import pandas as pd
from pandas import Series
import scipy as sc
import scipy.optimize
import matplotlib.pyplot as plt
%matplotlib inline
import matplotlib as mpl
mpl.rcParams['font.size'] = 14
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from keras.models import Sequential, load_model
from keras.layers import Dense, LSTM, Dropout

# Set Hyperparameters

look_back= 12
r = 0.8
epochs=1000
length = 1
neurons=[32, 16, 64]
dropout = 0.1

# Load Data

df = pd.read_csv('path/*csv')
df.Date = pd.to_datetime(df.Date, format='%d-%m-%y')
df = df.set_index('Date')
df_mean = df['AvgOil_total'].resample('M').mean()
```

```
raw_values = df_mean.values
raw_time = np.linspace(1, len(raw_values), len(raw_values))

# Define DCA

parm_init = np.asarray([90, .015, .9])
bounds=([10, 1e-6, 0], [1000, 1e6, 1])
def forecast(time, qi, D, b):
    if b > 1e-6:
        q = qi / (1 + D * b * time) ** (1 / b)
    else:
        q = qi / np.exp(D * time)
    return q
popt1, pcov1 = sc.optimize.curve_fit(forecast, raw_time, raw_values, p0=
    parm_init, bounds=bounds)
yfit1 = forecast(raw_time, *popt1)

# Transform Data to be Stationary using DCA and Create Batches

dataset1 = raw_values - yfit1
dataset1 = np.insert(dataset1, [0]*look_back, 0)
batches1 = []
for i in range(len(dataset1) - look_back):
    a = dataset1[i:(i+look_back+1)]
    batches1.append(a)
dataset1 = np.array(batches1)

# Split into Train and Test Sets

train1_size = int(dataset1.shape[0] * r)
test1_size = dataset1.shape[0] - train1_size
train1, test1 = dataset1[0:train1_size], dataset1[train1_size:]

# Transform the Scale of the Data

scaler1 = MinMaxScaler(feature_range=(-1, 1))
scaler1 = scaler1.fit(dataset1)
dataset1_scaled = scaler1.transform(dataset1)
train1_scaled = scaler1.transform(train1)
test1_scaled = scaler1.transform(test1)

# Recurrent Neural Network

model1 = Sequential()
model1.add(LSTM(neurons[0], batch_input_shape=(1, look_back, 1),
    return_sequences = True))
model1.add(Dropout(dropout))
```

```
model1.add(LSTM(neurons[1], batch_input_shape=(1, look_back, 1),
    return_sequences = True))
model1.add(Dropout(dropout))
model1.add(LSTM(neurons[2], batch_input_shape=(1, look_back, 1)))
model1.add(Dropout(dropout))
model1.add(Dense(1))
model1.compile(loss='mean_squared_error', optimizer='adam')

# Train Model

X_train1, y_train1 = train1_scaled[:, 0:-1], train1_scaled[:, -1]
X_train1 = np.expand_dims(X_train1, axis=2)
X_test1, y_test1 = test1_scaled[:, 0:-1], test1_scaled[:, -1]
X_test1 = np.expand_dims(X_test1, axis=2)
history1 = model1.fit(X_train1, y_train1,
                      validation_data=(X_test1, y_test1),
                      epochs=epochs,
                      batch_size=1,
                      verbose=0,
                      shuffle=False)


# Plot Loss Function Progress

mpl.rcParams['font.size'] = 14
fig = plt.figure(figsize=(12, 7))
plt.plot(history1.history['loss'])
plt.plot(history1.history['val_loss'])
plt.legend(['Train Loss','Validation Loss'], loc='best')
plt.xlabel('Epochs', fontsize = 20)
plt.ylabel('Loss', fontsize = 20)
plt.show

# Forecast

train1_match = list()
for i in range(len(train1_scaled)):
    X, y = train1_scaled[i, 0:-1], train1_scaled[i, -1]
    X = X.reshape(1, look_back, 1)
    yhat = model1.predict(X, batch_size=1)[0, 0]
    X = X.reshape(look_back)
    new_row = [x for x in X] + [yhat]
    array = np.array(new_row)
    array = array.reshape(1, look_back+1)
    array = scaler1.inverse_transform(array)[0, -1]
    yhat = array
    yhat = yhat + yfit1[i]
    train1_match.append(yhat)
n1 = length*train1_size
time1_BGA = np.linspace(1, train1_size+n1, train1_size+n1)
```

```
time1_DCA_date = pd.date_range(df_mean.index[0], periods=train1_size+n1,
    freq='M')
yfit2 = forecast(time1_DCA, *popt1)
predictions1 = raw_values[:train1_size][-look_back-1:]
for i in range(n1):
    diff1_ = list()
    for j in range(0, look_back+1):
        value_ = predictions1[-look_back-1+j] - yfit2[train1_size-look_back
    -1+i+j]
        diff1_.append(value_)
    diff1_ = Series(diff1_)
    dataset_ = diff1_.values
    dataset_ = dataset_.reshape(1, look_back+1)
    X_scaled = scaler1.transform(dataset_)
    X_scaled = X_scaled[0, 1:]
    X_scaled = X_scaled.reshape(1, look_back, 1)
    yhat = model1.predict(X_scaled, batch_size=1)[0, 0]
    X_scaled = X_scaled.reshape(look_back)
    new_row = [x for x in X_scaled] + [yhat]
    array = np.array(new_row)
    array = array.reshape(1, look_back+1)
    yhat = scaler1.inverse_transform(array)[0, -1]
    yhat = yhat + yfit2[train1_size+i]
    predictions1 = np.append(predictions1, [yhat])
predictions1 = predictions1[look_back+1:]
lastrec1 = df_mean.index[train1_size]
index1 = pd.date_range(lastrec1, periods=n1, freq='M')

# Plot Results

mpl.rcParams['font.size'] = 14
fig = plt.figure(figsize=(12, 7))
plt.plot(df_mean.index, raw_values, 'o', label='original', )
plt.plot(df_mean.index[:train1_size], train1_match, 'm--', label='DCA-DLSTM'
    )
plt.plot(index1, predictions1, 'm--')
plt.axvspan(df_mean.index[len(train1_scaled)+1], df_mean.index[-1],
    facecolor='blue', alpha=0.1)
plt.axvspan(df_mean.index[-1], time1_DCA_date[-1], facecolor='blue', alpha
    =0.2)
plt.legend(loc='best')
plt.xlabel('Date', fontsize = 20)
plt.ylabel('Oil rate (bbl/day)', fontsize = 20)
plt.show()
```

References

[1] A. Musapirova. (2019). Breadwinners of kazakhstan: 30 largest taxpayers in 2018, [Online]. Available: https://kursiv.kz/news/tendencii-i-issledovaniya/2019-03/kormilcy-kazakhstana-30-krupneyshikh-nalogoplatelschikov-v.

[2] D. Burba. (2019). An overview of time series forecasting models, [Online]. Available: https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb.

[3] L. Cao and F. E. Tay, "Support vector machine with adaptive parameters in financial time series forecasting", *IEEE Transaction on Neural Networks*, vol. 14, pp. 1506–1518, 2003.

[4] B. Ayeni and R. Pilat, "Crude oil reserve estimation: An application of the autoregressive integrated moving average (arima) model", *J. Pet. Sci. Eng.*, vol. 8, pp. 13–28, 1992.

[5] K. Holdaway, "Let oil and gas talk to you: Predicting production performance", *Proceedings of the SAS Global Forum 2012 Conference*, 2012.

[6] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep lstm recurrent networks", *Neurocomputing*, 2018. DOI: https://doi.org/10.1016/j.neucom.2018.09.082.

[7] L. S. Igor Aizenberg and L. Villa-Vargas, "Multilayer neural network with multi valued neurons in time series forecasting of oil production", *Neurocomputing*, vol. 175, pp. 980–989, 2016. DOI: https://doi.org/10.1016/j.neucom.2015.06.092.

[8] J. C.-C. Frausto-Solis and L. Sheremetov, "Forecasting oil production time series with a population-based simulated annealing method", *Russian Ge-*

ology and Geophysics, vol. 40(4), pp. 1081–1096, 2015. DOI: https://doi.org/10.1007/s13369-015-1587-z.

[9] F. Abdullayeva and Y. Imamverdiyev, "Development of oil production forecasting method based on deep learning", *International Academic Press*, vol. 7, pp. 826–839, 2019. DOI: 10.19139/soic-2310-5070-651.

[10] A. G.-S. Leonid B. Sheremetov and I. López-Yáñez, "Time series forecasting: Applications to the upstream oil and gas supply chain", *International Federation of Automatic Control*, 2013. DOI: 10.3182/20130619-3-RU-3018.00526.

[11] . G.-S. A. Sheremetov L. Cosultchi A., "Data-driven forecasting of naturally fractured reservoirs based on nonlinear autoregressive neural networks with exogenous input", *J. Pet. Sci. Eng.*, vol. 123, pp. 106–119, 2014.

[12] . G. M. Chithra Chakra N. Song K.Y., "Production forecasting of petroleum reservoir applying higher-order neural networks (honn) with limited reservoir data", *Int. J. Comput. Appl.*, vol. 72, pp. 23–35, 2013.

[13] T. Iordanova. (2020). An introduction to stationary and non-stationary processes, [Online]. Available: https://www.investopedia.com/articles/trading/07/stationary.asp.

[14] R. Orac. (2019). Lstm for time series prediction, [Online]. Available: https://towardsdatascience.com/lstm-for-time-series-prediction-de8aeb26f2ca.

[15] J. Brownlee. (2019). How to backtest machine learning models for time series forecasting, [Online]. Available: https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/.

[16] (2018). Production forecasting decline curve analysis, [Online]. Available: https://petrowiki.org/Production_forecasting_decline_curve_analysis.

[17] (). Root-mean-square error, [Online]. Available: https://en.wikipedia.org/wiki/Root-mean-square_deviation.