Machine Learning with Applications 3 (2021) 100013

Contents lists available at ScienceDirect

Machine Learning with Applications

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

A deep gated recurrent neural network for petroleum production forecasting

Raghad Al-Shabandar [a,*], Ali Jaddoa [b], Panos Liatsis [c], Abir Jaafar Hussain [a]

[a] Applied Computing Research Group, School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool, L3 3AF, UK
[b] Computing and Mathematical Sciences, University of Greenwich, London, SE10 9LS, UK
[c] Department of Electrical Engineering and Computer Science, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates

ARTICLE INFO

Keywords:
Deep gated recurrent unit networks
Long short-term memory networks
Recurrent Neural Networks

ABSTRACT

Forecasting of oil production plays a vital role in petroleum engineering and contributes to supporting engineers in the management of petroleum reservoirs. However, reliable production forecasting is difficult to achieve, particularly in view of the increase in digital oil big data. Although a significant amount of work has been reported in the literature in relation to the use of machine learning in the oil and gas domain, traditional forecasting approaches have limited potential in terms of representing the complex features of time series data. More specifically, in a high-dimensional nonlinear multivariate time series dataset, a shallow machine is incapable of inferring the dependencies between past and future values. In this context, a novel forecasting model for petroleum production is proposed in this work. The model is a deep-gated recurrent neural network consisting of multiple hidden layers, where each layer has a number of nodes. The proposed model has a low-complexity architecture and the capacity to track long-interval time-series datasets. To evaluate the robustness of our model, the proposed technique was benchmarked with various standard approaches. The extensive empirical results demonstrate that the proposed model outperforms existing approaches.

1. Introduction

Forecasting of oil production has become an increasingly important task in the oil and gas industry and can assist petroleum engineers in the analysis of the characteristics of crude oil wells. However, accurate prediction is hard to achieve and considerable amounts of finance and technology are required in order to assess the performance of a reservoir (Hammerton, Osborne, Armstrong, & Daelemans, 2002; Moreno, 2011).

In the past, several approaches have been utilised to forecast oil production, which can be broadly divided into statistical and soft computing methods. One of the most popular traditional statistical models is Decline Curve Analysis (DCA). The key limitation of this approach is that it considers only interpretations of the subject data, in such way that the decision-maker cannot gain deep insights into the factors influencing the oil production rate and reservoir performance (Fetkovich, Vienot, Bradley, & Kiesow, 1987). The most common traditional application is Autoregressive Integrated Moving Average (ARIMA) model, which has been successfully applied to diverse petroleum-related tasks such as forecasting oil prices, estimation of crude oil reserves and oil production. Despite the considerable potential of ARIMA, drawbacks related to capturing temporal sequence data represent significant shortcomings of this model (George, Gwilym, & Gregory, 2015).

In the last two decades, soft computing methods have attracted the attention of researchers, and the use of artificial intelligence has become more prominent in the oil and gas industry. Several machine learning algorithms have been extensively applied such as Fuzzy Logic (FL), Genetic Algorithms (GA), Support Vector Machines (SVM) and intelligent optimisation (Braswell, 2013; Mohaghegh, 2005). A neural network is one of the most popular machine learning models and is capable of handling nonlinear datasets (Zhang, Patuwo, & Hu, 1998). Although this approach has shown good prediction outcomes, it has several disadvantages in terms of solving time-series tasks. Since the information transfer in this model is only in the forward direction, it cannot track historical data (Moreno, 2011), and hence is incapable of inferring the dependency between past and future events.

Recurrent Neural Networks (RNNs) have been widely applied to perform forecasting activities in various areas of petroleum production. Still, they are inadequate to cope with long-interval time-series data (Hochreiter, Bengio, & Frasconi, 2011; Moreno, 2011). In order to overcome this limitation, Long Short-Term Memory (LSTM) was proposed as an extension of RNN. Empirical studies have demonstrated that LSTM can handle dynamic petroleum temporal data efficiently (Hochreiter et al., 2011). However, since massive amounts of data can be generated from upstream and downstream petroleum operations,

* Corresponding author.
E-mail addresses: ragednazar@yahoo.com (R. Al-Shabandar), a.h.jaddoa@gre.ac.uk (A. Jaddoa), panos.liatsis@ku.ac.ae (P. Liatsis), A.hussain@ljmu.ac.uk (A.J. Hussain).

https://doi.org/10.1016/j.mlwa.2020.100013
Received 2 August 2020; Received in revised form 18 November 2020; Accepted 18 November 2020
Available online 23 November 2020

this type of shallow machine learning may be insufficient to represent complex temporal structures, particularly for large, heterogeneous datasets (Najafabadi et al., 2015).

Deep learning is a promising solution that can be applied to solving problems in the oil and gas field. The ability to learn various levels of data representation that match the hierarchical elements of the relational architecture is one of the distinctive features of deep learning. This approach can automatically learn the temporal dependence of heterogeneous data, and can also extract patterns from long input sequences without explicit, upfront human effort (Goodfellow, Bengio, & Courville, 2016). Although deep learning is a sophisticated approach for long-interval time series that can dramatically reduce the human involvement needed to extract features, it has limited potential due to its high costs and complex architecture (Huang, Song, Hong, & Xie, 2014).

In this paper, a Deep Gated Recurrent Unit (GRU) model is proposed for petroleum forecasting that can support learning using complex temporal datasets with low-complexity structures. The model involves a stack of several GRU layers, each of which contains several neural networks. In our framework, the hierarchical feature representation of time series data can be described in a more sophisticated way by selecting only a few parameters.

The research provides major novel contributions for oil gas industry. To the best of our knowledge, none of the current work has applied GRU model to the oil and gas domain. Compared with state of the art, all existing production models rely on RNNs and LSTM, which have high memory requirements (Sagheer & Kotb, 2019). Our framework can capture the long-term dependencies of time series data without the need for large amounts of memory.

The remainder of this paper is organised as follows. Section 2 will give detailed information about prior works. Section 3 shows the overview of big data followed by detail explanation of recurrent neural network. In Section 4 the real application including data description, exploratory data analysis, Data Pre-Processing are described. The detailed explanation of proposed Stacked Gated Recurrent Unit Network, experiment setup and experiment results are provided. Simulation results are shown in Section 5 while conclusion and future works are shown in Section 6.

2. Literature review

A number of researchers have successfully applied neural networks in the oil industry, such as in reservoir history-matching forecasts for natural gas production, pattern recognition in well test analyses and crude oil price forecasting (Hammerton et al., 2002).

The authors of Ferreira da Silva, Portella, Emerick, and Favilla Ebecken (2007) used a neural network to learn a time series data; it was employed to perform fluid production forecasts for two onshore fields (Bonsucesso and Carmopolis). Since the production data were collected from very small oilfields, a synthetic reservoir was used to capture long-term historical production data. The results demonstrated that the neural network performed better in both the long and short terms than DCA. Although the neural network was successfully applied as a time series, it suffers from several limitations including the inability to handle discontinuities and high-order nonlinearities in the dataset.

A feed-forward neural network inspired by the Imperialist Competitive Algorithm (ICA) was proposed by Berneti and Shahbazian (2011) to predict oil flow rates in oil fields in the northern Persian Gulf. ICA is capable of optimising the initial weights of a neural network by combining the local searching ability of BP with the global searching ability of ICA. The results showed that ICA-Artificial Neural Network (ANN) achieved Mean Squared Error (MSE) of 0.0123, while the traditional ANN obtained a value of 0.04702. The main disadvantage of the ANN and ICA-ANN networks is that the topology of the networks needs to be determined manually, meaning that optimisation of the structure of the neural network may take a long time (Berneti & Shahbazian, 2011).

The wireline formation-testing tool was widely used to capture oil production data. The tool can be defined as an electric logging cable that is used to measure the formation pressures at a particular location in the timestamp. This measurement can be useful to determine the variations of pressure across different formations; The tool can help the drillers to determine if there is enough amount of hydrocarbons which can guide them moving to next phase of completing well production (Masoudi, Arbab, & Rezaei, 2014).

Researchers have found that feeding production data from a wireline formation-testing tool into a neural network can give rise to large errors, and various analysis methods have therefore been used. For instance, the researchers in Aizenberg, Sheremetov, Villa-vargas, and Martinez-mu noz (2016) employed wavelet analysis to extract information from a Modular Dynamics Testing (MDT) application and then fed it to a neural network. The results revealed that combining wavelet analysis with neural network could significantly refine the performance of a predictive model.

A Higher-Order Neural Network (HONN) is a promising technique for forecasting oil production (Frausto-Solís, Chi-Chim, & Sheremetov, 2015). The most distinctive feature of a HONN is its ability to map nonlinearities with few features, and it has been shown that a HONN has significant advantages over traditional methods, such as faster training with lower computational cost (Frausto-Solís et al., 2015).

The authors of Song, Saraf, and Gupta (2013) applied a HONN to forecast the oil production of a petroleum reservoir, and oil, gas and water data were used to verify the performance of the network. The results showed that HONNs could be efficiently applied in oil production forecasting, achieving a value of 13.86 for Mean Absolute Percentage Error (MAPE). Although the HONN model shows potential in terms of delivering reliable oil production forecasts, it can only be applied to short time intervals (i.e., six to 18 months). The use of a more advanced ANN is presented in Aizenberg et al. (2016), where a multilayer neural network with multi-valued neurons (MLMVN) was employed to forecast oil production. The most critical feature of the MLMVN was that training did not require the derivative of the activation function, making the learning process quicker and simpler than the ANN.

The ensemble method of Gradient Boosting Machines (GBM) was used in Bikmukhametov and Jäschke (2019) for estimation of the flow rate of an oil production system under various conditions. The findings revealed that GBM gave an accurate flow rate prediction and could be used as a virtual flow metre. The computations involved in this virtual flow metre were relatively low compared with an actual multiphase flow metre. Recently, the authors of Ma and Liu (2018) proposed a hybrid model called the Nonlinear Extension of Arps (NEA) decline model, which integrated the kernel method with the Arps decline model. The kernel trick was used to improve the functionality of the Arps linear model into a nonlinear multivariate framework. Data were collected from real oil fields in China and India; the findings showed that NEA model is a robust model capable of discovering the nonlinear relationships between features and oil production.

With the growing success of AI in oil and gas industry, machine learning was utilised to monitor oil production rate, the author in Khan, Alnuaim, Tariq, and Abdulraheem (2019) used Artificial Neural Fuzzy Inference systems (ANFIS) alongside with neural network to predict the oil rate on gas lift wells. The result shows that the artificial neural network model achieves the highest accuracy flow rate with a value of 99% compared with other non-linear regression models.

With fast of development of technology, machine learning was applied to solve the problem related to offshore oil field predation facilities; the researchers in Chen, Gang, Junfei, Zheng, and Jinyuan (2019) proposed the IA-SVM model to examine the factors that impact the corrosion of submarine oil and gas pipelines. The model is combined Support Vector Machine (SWM) with by Immune Algorithm (IA). The predicated MAR and RMSE of IA-SVM model are shown relatively small with the values of 1.45, 0.015, respectively.

3. Recurrent neural networks and big data

In this section, the concept of big data for oil data prediction will be presented as well as various recurrent neural network architectures.

3.1. Big data

With rapid advances in technology, massive amounts of data can be generated from upstream and downstream oil and gas operations. Big data analytics has become an essential tool for many oil and gas applications (Crockett & Kurrey, 2014; Mohammadpoor & Torabi, 2019). In drilling operations, big data has the potential to increase drilling safety and minimise drilling time. In addition, the analysis of big data has been shown to significantly improve oilfield production while enhancing safety (Crockett & Kurrey, 2014; Feblowitz, 2013).

In 2018, the authors of Ockree, Brown, Frantz, Deasy, and John (2018) utilised big data to build a predictive production model. Data generated from the machine-learning model were combined with economic analysis to provide insights and guidance for field development. The results demonstrated that the integration of big data analytics into the oil and gas industry could increase revenue and provide added value.

Although the use of big data has been shown to significantly improve several aspects of the oil and gas industry, the handling and processing of these data are primary concerns (Crockett & Kurrey, 2014; Sivarajah, Kamal, Irani, & Weerakkody, 2017). The most important technical challenge in big data applications is the cost of data storage. Emerging trends in new technologies such as cloud computing and the Internet of Things (IoT) have been successfully applied to address computational cost concerns (Anderson, 2017).

However, utilising big data in the oil and gas domain showed that the volume of data is challenging problem since huge amounts of information is often generated from exploration, drilling and production operations (Azzedin & Ghaleb, 2019; Feblowitz, 2013). There are various shallow machine learning methods that have been reported in the literature in solving problems in the oil and gas field. However, a well-known limitation of these approaches is that they are unable to infer nonlinear relationships between future and past values with respect to long intervals in high-dimensional data (Frausto-Solís et al., 2015; Song et al., 2013).

Furthermore, the potential for extracting information from big data is relatively narrow. Since shallow machine learning is insufficient to help engineers extracting information from reservoir model. Petroleum engineers should spend more time in understanding the characterisation of the reservoir (Frausto-Solís et al., 2015). To overcome this issue, researchers have adopted more advanced machine learning methods to reduce the need for human intervention. For example, in Korjani, Popa, Grijalva, Cassidy, and Ershaghi (2016), the authors built a deep neural network to predict the petrophysical characteristics of reservoirs. The findings reveal that deep learning is a promising solution that could be used to identify reservoir characterisation.

Investigation of the status of pump units in oil fields is a complex task, and shallow machine learning has been shown to be inadequate in inferring the condition of such equipment. The authors of Korovin (2017) applied a Deep Recurrent Neural Network (RNN) in conjunction with a Convolutional Neural Network (CNN) in the prediction of equipment status for oil and gas production. They suggest the use of additional noise to increase the size of the RNN; however, this noise would affect the decision boundaries. The author did not include results from deep machine learning (Korovin, 2017).

According to the existing literature, a variety of shallow machine learning methods have been used by researchers to estimate the price of crude oil, although these models cannot accurately identify the entire range of factors that impact crude oil markets (Yu, Dai, & Tang, 2016; Zhao, Zeng, Wang, & Zhang, 2019). As such authors in An, Mikhaylov, and Moiseev (2019) employed linear regression to forecast oil price. The authors find various factors could influence oil price including the US Federal Reserve rate US dollar index, conflicts in the Middle East and terrorist attacks. The finding reveals that predicated oil price was close to the real oil price. However, the shallow machine learning model fails to estimate the sharp fall in prices caused by changes in the market condition of oil futures. Deep learning is a promising method for forecasting oil prices, especially using big data. A deep belief neural network, in conjunction with LSTM, was used to forecast crude oil price in Yu et al. (2016). To capture the nonlinear dynamics of crude oil price movement, a hybrid model was constructed that combined deep learning models with the Adaptive Risk Analysis and Management model (ARAM). The findings demonstrated that deep learning improved the performance of the predictive model for the crude oil price. Deep learning also has the potential to detect all of the factors that influence crude oil prices (Yu et al., 2016).

There are various tools that can be used to capture digital oil field data in real-time, and sensors have been widely applied in the oil and gas industry for the purpose of capturing huge amounts of real-time data (Mohammadpoor & Torabi, 2019). In a recent study, industrial IoT and sensors were utilised to collect hourly data from a real, active field over 11 days (Singh & Pateriya, 2019). ARIMA was combined with deep learning to forecast future oil production over a short period. The findings revealed that incorporating deep learning into ARIMA rapidly enhanced the predictive performance of the model. The authors also found that a deep learning model could detect well production going upwards and downwards in a more precise way (Singh & Pateriya, 2019).

A deep LSTM model called “DLSTM” was proposed in Sagheer and Kotb (2019) for oilfield production problems, and a genetic algorithm was utilised to select the optimal parameters. The results revealed that the proposed model performed better than ARIMA, particularly for complex, long-sequence time series data. The results also demonstrated that DLSTM could effectively handle nonlinear relationships in time series data. The authors evaluated DLSTM only with a univariate dataset (Sagheer & Kotb, 2019).

3.2. Recurrent Neural Network

Recurrent Neural Network (RNN) is an extension of an artificial neural network that is capable of handling temporal patterns and has an internal memory to process arbitrary temporal sequences. This type of network has a feedback loop that can store information about past behaviour (Connor, Martin, Atlas, & Ieee, 1994; Giles & Lawrence, 2001). RNN turns input sequence data in vectors and processes the current observation in the context of the internal (hidden) state of the network to generate the output.

RNN could be inadequate in processing long-term sequential data as it suffers from the vanishing gradient problem. With a large time-series dataset, network errors are replicated from the output layer into the input layer in the training phase (Karim, Majumdar, Darabi, & Chen, 2018; Tian & Pan, 2015). This propagated error can cause the weight decay of the gradient descent to become smaller and to stall in the lower layers over time. When the vanishing gradient problem occurs in a dataset, it can prevent further training of RNN (Hochreiter, 1998).

A Short-Term Long Memory (LSTM) is a type of RNN that can represent long-term dependencies and is capable of resolving the vanishing gradient problem. It is composed of two units, a standard and a special unit. The special unit can store information in memory for a long time by using multiplicative unit gates. For each step, constant error set in a special unit, the unit gates control a constant error flow. Hence, LSTM can modify the parameters of weight decay over time (Weninger, Bergmann, & Schuller, 2015).

LSTM network consists of three gates (i.e., a forget gate, an input gate and an output gate) in addition to the cell state. The input at the current time step is combined with information about the hidden state

in the previous time step, and is passed through an activation function to the input gate, as follows (Weninger et al., 2015):

$$f_{t=}\sigma(X_t W^f + S_{t-1}U^f + b_f) \tag{1}$$

where f_t is forget gate. X_t is the input at time step t and S_{t-1} represents hidden state at the previous time step t − 1. W^f is the weight of the input layer and U^f is recurrent weight of the hidden state. The b_f is the bias of the input layer. The input gate decides which information will be stored in the cell state. The input gate has two tasks: in the first task, the input gate layer decides which value is updated, while in the second task, tanh layer regulates the network by creating a vector of all new candidate values. The equations for the two tasks are as follows (Salehinejad, Sankar, Barfett, Colak, & Valaee, 2017; Weninger et al., 2015):

$$i_{t=}\sigma(X_t W^i + S_{t-1}U^i + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(X_t W^c + S_{t-1}U^c + b_c) \tag{3}$$

$$C_t = C_{t-1} * f_t + i_t * \tilde{C}_t \tag{4}$$

The output gate decides which hidden states are used for prediction via a sigmoid activation function. The new modified cell state is passed to the tanh function and multiplied to produce the output, as follows (Salehinejad et al., 2017):

$$O_{t=}\sigma(X_t W^o + S_{t-1}U^o + b_o) \tag{5}$$

$$S_{t=}O_t * \tanh(C_t) \tag{6}$$

The weights can be represented by two matrixes, {W, U}. The matrix $W = \{W^i, W^c, W^o\}$ represents the input weights associated with the input layer, hidden layer and output layer, respectively, while $U = \{U^i, U^c, U^o\}$ denotes the weights of the previous hidden state (i.e., recurrent weights) in the input layer, hidden layer and output layer. The collection of bias vectors can be given as a matrix $B = \{b^i, b^c, b^o\}$, with the indices corresponding to the same layers as described previously (Salehinejad et al., 2017; Weninger et al., 2015).

The Gated Recurrent Unit (GRU) network is a new generation of RNN that was introduced in 2014 by Cho et al. (2014). The GRU is similar to LSTM but has a less complex structure, and can control the flow of information without the need to use a separate memory cell. GRUs have been successfully applied in speech recognition tasks, such as speech enhancement, speech activity detection and text recommendation (Cho et al., 2014).

A GRU comprises two gates, i.e., the update and the reset gates, to track the state of sequences. The update gate is responsible for controlling how much past information is maintained and how much new information is added.

The reset gate, on the other hand, controls how much the past state contributes to the current state. A distinctive feature of the GRU is its ability to capture dependencies of long time series without requiring internal memory cells, in contrast to LSTM. This assists in solving the vanishing gradient problem efficiently (Cho et al., 2014).

The update gate operates in a similar way to the forget gate and the input gate in LSTM. The new input is multiplied by its weight, while the hidden state at the previous time step is multiplied by the recurrent weight. The contributions of both candidates are added together, and the sigmoid function is used to map the resulting values between zero and one:

$$Z_{t=}\sigma(X_t W^z + h_{t-1}U^z + b_z) \tag{7}$$

where Z_t is represented the update gate, the X_t the input vector at time step t while h_{t-1} is the previous output from previous units. The W^z is the weight of the input layer, and U^z is the recurrent weight. The b_z is the bias of the input layer. The reset gate decides how much past information should be forgotten. Its equation is similar to that of the update gate. The new input unit and hidden state are multiplied by their corresponding weights $\{W^r, U^r\}$, and the result is summed and passed through a sigmoid function. The output of the reset gate is as follows:

$$R_{t=}\sigma(X_t W^r + S_{t-1}U^r + b_r) \tag{8}$$

The new memory contact is used reset gate to store the information of the previous state. The input gate multiplied by its weight, element-wise multiplication Hadamard ($\odot$) should be applied in the rest gate (R_t) and previous out (h_{t-1}). This will allow the network to select only relevant past information. Current memory contact is calculated as follow.

$$\tilde{h}_t = \tanh(X_t W + U(R_t \odot h_{t-1})) \tag{9}$$

$$h_{t=} = Z_t \odot h_{t-1} + (1 - Z_t) \odot \sigma(\tilde{h}_t) + b_h \tag{10}$$

The update gate decides what information should be collected from the current memory content and previous steps. The new hidden state should be store at final memory as output for current time step and pass via sigmoid (σ) activation function.

4. Real data application

4.1. Data description

The US Distribution and Production of Oil and Gas Wells dataset is available from the US Energy Information Administration (EIA) website (Eia, 2018). Drillinginfo, a leading Software as s Service (SaaS) and data analytics company for energy exploration decision support, collected annual production data from various US state agencies between 1919 and 2009 (U.S. Energy Information Administration, 2019). Some states did not have any productive wells, and thus, the records have missing values for the relevant feature set entries. Two sets of features are considered in the dataset, categorical attributes and numerical features. The numerical features are extracted based on the two factors of the state for which the data was collected and the rate class. Overall, nine input features are captured in the database (Eia, 2018).

The "rate class" variable describes the class of crude oil price based on the customer group; there were 22 rate classes of oil provided in the database. The "num_oil_wells" variable is defined as the number of oil wells in each state, while the "oil_wells_dayson" variable is the number of unique days that wells remained open to complete the process of producing oil. The "'conden_prod_BBL" feature represents a mixture of light liquid hydrocarbons and condensate measured in barrels.

An estimation of the annual gas production is also reported in the data. The raw gas fields include {"num_gas_wells", "NAgas_prod_MCF", "gas_prod_MCF" and "gas_wells_dayson"}. The "num_gas_wells" variable represents the number of gas wells, while the "gas_wells_dayson" variable represents the number of days that a gas well was open to accomplish the gas production task. The "NAgas_prod_MCF" and "ADgas_prod_MCF" variables define the quantity of natural gas, measured in thousand cubic feet (MCF) (Eia, 2018; U.S. Energy Information Administration, 2019).

The "oil_prod_BBL" output variable represents the production data from the first month to the last year of production, across each state and rate class. The volume of oil production is measured in barrels. In the USA, a barrel is equivalent to 42 gallons (U.S. Energy Information Administration, 2019).

4.2. Exploratory data analysis

Exploratory Data Analysis (EDA) is applied in this study in order to gain insight into the rate of oil production from wells over time. Oil production data were collected from 31 states in the US and were aggregated based on the state and year. The primary step in our time series analysis is to examine the stationarity of the data and to achieve this; we conduct an Augmented Dickey–Fuller test (ADF).

ADF is a statistical hypothesis test that is used to evaluate the stationarity of a time series dataset (Lee & Chang, 2008). The results

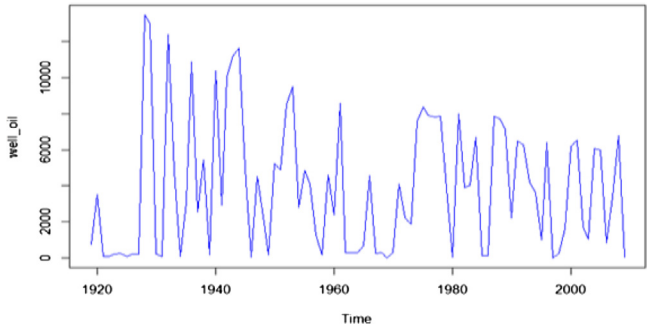

Fig. 1. Number of oil-producing wells in the USA over time.

show that the *p*-value is (0.01) greater than 0.05, which allows us to accept the null hypothesis. Thus, these data are strongly classified as a stationary series.

The time series plot was used to visualise the difference in US oil production across various wells over time. As it can be shown from Fig. 1, there are random fluctuations in the data over time, particularly in the period between 1920 and 1930; however, a constant variation in the oil production rate is shown for the last eight years. Since the random fluctuations in the data are approximately constant over a period of 45 years, the additive model can describe the US oil production data.

The residual plot in Fig. 2 can be used to evaluate whether or not the linear model fits the data. The Generalised linear model (GLM) model is utilised to examine the linearity of the time series data. The residual obtained from a forecast of oil production shows that the linear model does not fit the data well, and as a consequence, a more advanced forecasting method should be considered. The histogram shows that the distribution of the data does not follow the normal distribution and seems to be skewed to the left of the curve. We normalised the data with aim to obtain a more accurate result with respect to correction between features across time.

The relationship between the observations from the current time series with observations from the previous time step is examined in this study using the Autocorrelation Function (ACF), in conjunction with the Partial Autocorrelation Function (PACF). As it can be seen from Fig. 2, the ACF plot shows that the autocorrelation is significant for a lag interval of 1–10. Since autocorrelation corresponds to the confidence interval, the significance of autocorrelation is determined by two factors, namely the length of time series and standard deviations of the confidence interval. The observed correlation showed normally distributed, hence 95% confidence level was observed.

The highest correlation is found for lag 1, and we can see that there is little correlation for lags 12–18. A weak negative correlation can be seen between lags of 18 and 40. The PACF plot shows the first, second and seventh lag values are significant, whereas the PACF for other lags is not significant. A negative correlation is found for lags of 10–20 and 30–40.

4.3. Data pre-processing

The data pre-processing phase is divided into two steps, i.e., noise reduction and data transformation. Data cleaning was employed to handle the noise in the data.

As previously mentioned, US Distribution and Production of Oil and Gas Wells dataset is annual data covering the period from 1919 to 2009 with a total of 27,290 observation. On inspection, around 17% of the oil wells in some US states were unproductive during the period, and thus, 6290 records were excluded from the dataset. The data was split into three sets, forming the training, validation and testing sets. The training and validation sets were normalised by transforming the input data and normalised with a zero mean and standard deviation of one.

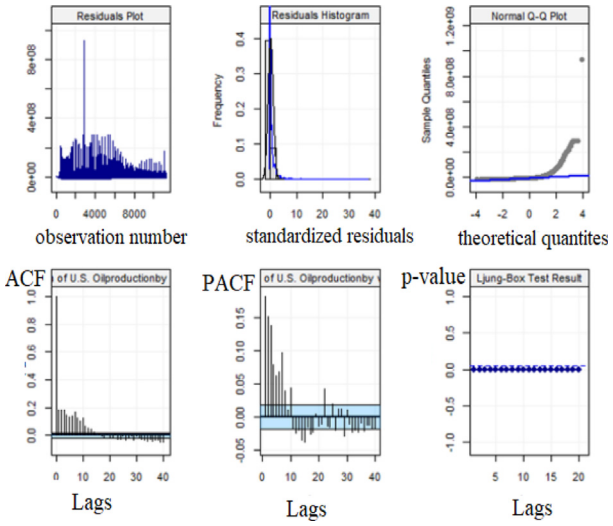

Fig. 2. Diagnostic plots.

Since oil production in the US is measured in barrels (bbl), oil production values were large and heterogeneous. It can be problematic to feed a neural network with large values, and the proposed model may therefore suffer from high generalisation errors and poor performance (Erhan, Szegedy, Toshev, & Anguelov, 2014). To overcome the issue, the output variables for oil production were scaled based on the calculated maximum value for oil production in the training dataset. The output values for the training, validation and test sets were divided by this maximum value. Using this approach, a smoother estimation of oil production could be generated (Erhan et al., 2014).

4.4. Stacked gated recurrent unit network framework

A deep GRU was proposed in this study with the aim of improving the performance of the predictive model for petroleum production, particularly over long time intervals. The proposed model is capable of handling the temporal dependences of complex time-series data at a deep level. It consists of stacks of several layers, where each layer solves part of the task and passes the results to the next layer. Since each layer combines the learned representations of the previous layer and feeds them to a higher layer, better representations of the data can be achieved in the model.

Fig. 3 shows the architecture of the Deep Gated Recurrent Unit Network (DGRU). The DGRR consists of n blocks, where each block has one GRU layer in addition to a dropout layer. The dropout layer is used as a regularisation method to reduce overfitting by the GRU stack. To provide a more powerful recurrent neural network, all intermediate layers are customised.

As can be shown from Fig. 3, the GRU layers are stacked on top of one another across several blocks. The input data is fed in with the previous hidden state to the first GRU block. The dropout layer randomly drops the output unit of the GRU layer based on the dropout rate, which is set to 0.5. The output values of the GRU layer below the dropout rate are removed, while the remaining values from the GRU layer in the first block are passed forward to the next GRU block. The previous hidden state is combined with the current state and fed to the next block until all blocks are fully stacked. The output from the stack GRU layer is transferred to the feed-forward neural network via a dense layer to produce the final output.

The main advantage of the stack GRU is its structural simplicity; for instance, entire layers can learn a complex representation of long interval time series data jointly. In this scenario, when adjusting the single features of the model, all feature dependencies can be automatically

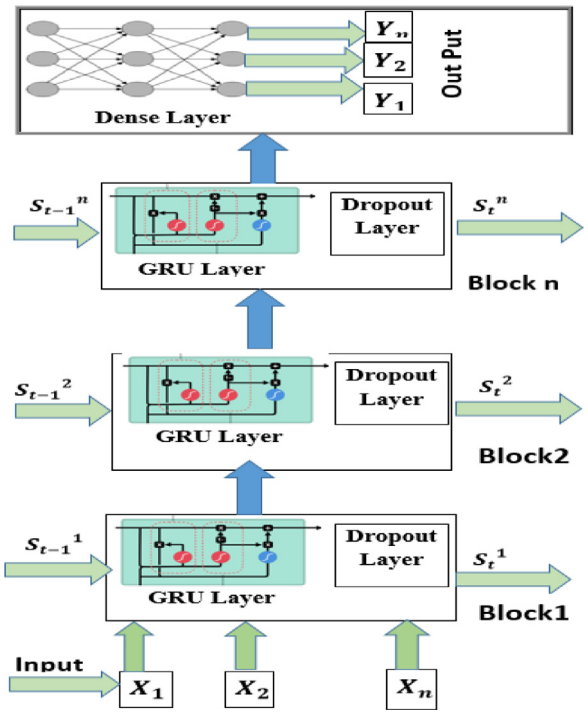

Fig. 3. DGRU recurrent network architecture.

updated without requiring human intervention. Another benefit of this model is that it can run at different time scales by breaking layers into intermediate blocks, where each block is able to capture information over the entire period.

4.5. Model construction and validation

The dataset contains 21,000 observations of US oil Production data over a period of 90 years. The original dataset is divided into three subsets, i.e. the training (40%), validation (20%) and testing sets (40%). This method can be used to evaluate the performance of the trained model as well as to assess the stacked deep GRU model using unseen data.

A 3D tensor is used to store the oil production data. As shown in Fig. 4, the tensor has three axes, with dimensions (sample, time step, features). 'Sample' refers to the sample size of the training set and validation set, which contains 13,000 observations. The training data are divided into small sample batches of different sizes and then fed into the model. Since the aim of this study is to forecast the annual oil production for each US state, we set the time step to one. This study uses various types of neural networks, including a multi-RNN, single and multi-GRUs, an Artificial Neural Networks (ANN), multi-LSTM and the proposed DGRU, with different values of hyperparameter.

4.6. Forecasting algorithms for time series data inspired by stacked deep learning

Artificial neural network with various topologies and RNN showed sufficient capacity to predict oil production only for short term forecasting intervals. These machine learning models have difficulties on longer interval and high nonlinearity properties of time series data (Cho et al., 2014). To overcome this drawback, the LSTM is used to forecast petroleum production forecasting. Despite the fact that it achieves more accurate forecasting rate, its performance to handle high dimensional time series data is not satisfactory (Karim et al., 2018). To this end, an algorithm for Oil and Gas Production is proposed in this work. The algorithm can be used in a time series regression

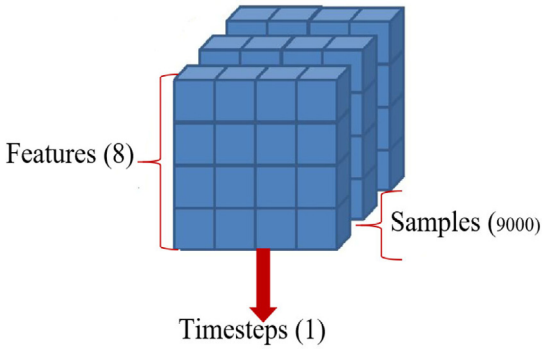

Fig. 4. 3D tensor for US oil and gas production time series dataset.

setting practically for long-term dependencies tasks. The DGRU consists of stacking multiple GRU layers of nonlinear operation with several dropout layers. It showed potential of solving any time series task with low computational cost. Nevertheless, the deep GRU capable of training high dimension dataset faster than LSTM. Thus, an algorithm can be applied in high dimensional dataset effectively. The detailed of a proposed algorithm is described as follows.

In our algorithm, we assume dataset D consists of a sequence of samples $\{s_{1,}\ s_{2,}\ldots s_{n,}\}$, each sample denotes a series of data points listed in time order $\langle X_{i,}T_i\rangle$ where $x_i\in$ X, X is the set of attributes $\{X_{1,}\ X_{2,}\ldots,X_{m,}\}$ at different time t and $t_i\in$T is a unique time interval. The D has been split into smaller subsets {tr, vl, ts}. The train (tr) and Val (vl) are employed for the learning process, while the ts (test) is used to evaluate the performance of the model. The three sets are selected sequentially. The users can dynamically determine the number of instances in a sample (S) and allocate this to each subsets tr, vl, ts in database D.

In the first stage, noisy data is cleaned with the aim of better oil production in future. For each subset, the fluctuation from noisy data is minimised by normalised input to $(\mu,\ \sigma)\,(0,1)$. Three new subsets are derived for database D = { Tr1, Vl1, Ts1}. Each new subset represents the transformation set the means of data to 0 and standard deviation to 1 for subsets tr, vl, ts, respectively. The output of oil production are large values, learning neural networks with large weights could cause large error gradients values. A model with large error can result in the instability in the learning process and high generalisation error. To solve the problem, the target output variable is normalised by applied the max normalisation (range transformation method). As can be seen in tAlgorithm 1, the target variable for each subset tr, vl, ts is divided by the maximum oil production value, we defined as (Tr_mx) variable, the new subsets of scaled output are generated for database D = {Tr_o, VL_o Ts_o}.

The second stage of the proposed algorithm is to train the regression models; four regression neural networks are employed to forecast oil and gas production in addition to our proposed DGRU model. We define the variable (π) as a training counter. Each model (rce_r) training the scaled data (Tr1) that are obtained from the previous stage. The learning procedure for each (rce_r) is iterated 100 times until the required threshold is achieved. Replicate the training process over time could help $(rce)$to minimise the error at every time and achieved more accurate predicted values.

In the third stage, the model makes the prediction. The trained model is used to generate the predicted output $(\hat{Y}$ on unseen data and compared with scaled output values (Ts_o) of the test dataset. In the last stage, regression matrix was generated to evaluate the performance of the oil production models these including loss, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Relative Square Error (RSE), and R-squared (R^2). Each matrix is obtained by computing the difference between (Y_i) that corresponding to output values of the scaled set subset (Ts_o) with the predicted output $(\hat{Y}_i)$.

The last stage, the regression performance matrix report is generated. The performance set is defined as P = {Loss, MAE, MSE, RMSE, R2} calculated as shown in Eqs. (11)–(15), respectively.

$$\text{Loss} = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \tag{11}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| \left(\hat{Y}_i - Y_i\right)^2 \right| \tag{12}$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\hat{Y}_i - Y_i\right)^2} \tag{13}$$

$$\text{RSE} = \frac{\sum_{i=1}^{n} \left(\widehat{Y}_i - Y_i\right)^2}{\sum_{i=1}^{n} \left(\overline{Y_i} - Y_i\right)^2} \tag{14}$$

$$R^2 = 1 - \text{RSE} \tag{15}$$

Algrthim1. Petroleum Production Model

Input: S ⟶set of n samples where S={$(x_i, y_i), \ldots x_m, y_n)$}
X ⟶set of m-dimensional features where $x_i = \{x_{i1}, \ldots, x_m\}$
Y ⟶set of real oil production Y = {$y_1, \ldots y_n$}

Output: $Y^{\wedge}$ ⟶Set of predicted values where $Y^{\wedge} = y_1{}^{\wedge}, \ldots y_n{}^{\wedge}$}

Train = { tr ∈ S ⇒ tr ⋉ S}
Val = { vl ∈ S ⇒ vl ⋉ S}
Test = { ts ∈ S ⇒ ts ⋉ S & ts ∉ Train ^ ts ∉ Val }
i → index to number of samples

For ∀ (x_i, y_i) in tr, **do**
Tr1=$\frac{x_i - \mu}{\sigma}$ where $\mu = 0 \wedge \sigma = 1$
Tr_mx= $calcuate(Max)$ per y_i
Tr_o=y_i / Tr_mx
End for

Tr_mx ⟶ Maximum output value in training set.

For ∀(x_i, y_i) in vl , **do**
Vl1 =$\frac{x_i - \mu}{\sigma}$ where $x_i \notin$ tr, $\mu = 0 \wedge \sigma = 1$
VL_o=y_i / Tr_mx
End for

For ∀ x_i in vl, **do**
Ts1=$\frac{x_i - \mu}{\sigma}$ where $\mu = \text{Tr1}_{\mu} \wedge \sigma = \text{Tr1}\_\partial$
Ts_o=y_i / Tr_mx
End for

Rcu ⟶set of recurent neural networks, Rcu = {multi-RNN, single GRU, multi-GRU/ ANN, multi-LSTM, DGRU} where $rce_r \in$ Rcu.

P ⟶set of performance matrix, P = {Loss, MAE, RMSE, RSE,R^2} where $p_a \in$ P.

π → Training counter threshold

For ∀ rce_r in Rcu, **Do**
While π <100,**do**
$\frac{1}{m}\sum_{i=1}^{m} \boldsymbol{Train}(y_i, f(x_i, rce_r)$
Stop training when a stop-criterion met
End While

For ∀ p_a in P, **Do**
$\widehat{y_i} = f(x_i, rce_r)$, $x_i \in$ Ts1
Compute performance as in Eq.n11,12,13,14,15

End for
End for

5. Simulation results

The simulation results for the oil production models are presented in this section. The analysis was run ten steps ahead for the multi-RNN, single GRU, multi-GRU, ANN, multi-LSTM and the proposed DGRU. The average values for each model were computed to acquire the final forecasting result. Hyperparameters were used to obtain promising network architectures. Two approaches were used to configure these, namely optimised hyperparameters and model-specific hyperparameters. The results of hyperparameter tuning for each model are shown in Table 1.

Since batch size plays an important role in preventing the learning process from getting stuck at local minima, the training and validation sets were divided into a set of small batches, which were passed to the neural networks. As can be seen from Table 1, the dataset is split into various batch sizes (20, 32, 64, and 128).

The overall number of epochs for the network architectures was generally set to large values, with the aim of reducing the training error and improving the performance of the models. The learning rate was also tuned by considering a variety of learning rate values. The number of hidden units and the number of layers were experimentally tuned.

Fig. 5 presents the MAE metrics for the training and validation sets. As can be seen from Fig. 5, the values for the validation and training sets are constant over all epochs for the multi-RNN model. The model over fitted the data after the first epoch, and therefore does not perform well on the US oil production data.

With regards to the single GRU model, the MAE decreases linearly over time, reaching nearly 0.001 for the validation set, while the value for the training set decreases after 100 epochs and then stalls. The MAE is slightly decreased for the validation sets, reaching a minimum after 200 epochs, while the value for the training set continues to decrease after reaching 0.001. Again, the model is overfitting the data. Although both the LSTM and DGRU models are explicit, with three dropout layers, the LSTM is overfitting after 5 epochs. The number of epochs was reduced to 5 to avoid overfitting, but the results show that the neural network does not converge for a small number of epochs.

The MAE shows improvement in the DGRU model. Overfitting is mitigated in this model by adding three dropout layers, and the MAE drops linearly, achieving its lowest values of 0.0027 and 0.0024 for the validation and training sets, respectively, at epoch 100. Hence, the DGRU model achieves the best results.

The results are presented in Tables 2–6. The findings show that the loss metric showed slightly better performance for the training and validations datasets than the test sets, for all recurrent neural network models. However, the DGRU model achieved the best loss value, reaching 0.00009 for the test dataset.

Although multi-GRU/ANN model achieved the best values for the RSME, it did not generalise well on the test data, as there was a large difference in the RSME error between the training and test data. The best model with regard to RSME metric was again the proposed DGRU. The RMSE was increased by 1% in the DGRU model, an average value of 0.0078 was achieved.

The R squared metric is used to evaluate the goodness of fit predictive model into the regression line. This is called the coefficient of determination. The best R^2 was obtained by the multi-GRU/ANN and DGRU methods, which gave values of 0.71923 and 0.70020, respectively.

The relationship between the original production data and the predicted data for US oil production using 10 steps ahead prediction is shown in Fig. 6.

Large amount of data to train neural networks in replicating the analysis of simulations for ten times which could make the simulations process slower. Our purposes DGRU capable of learning long term dependencies fast. It can act significant improvements in prohibitively of the slowing learning process.

DGRU model can train fast emulators for high dimensional data while offering significant enhancement in respect to accuracy compared with other deep recurrent neural network techniques. In order to

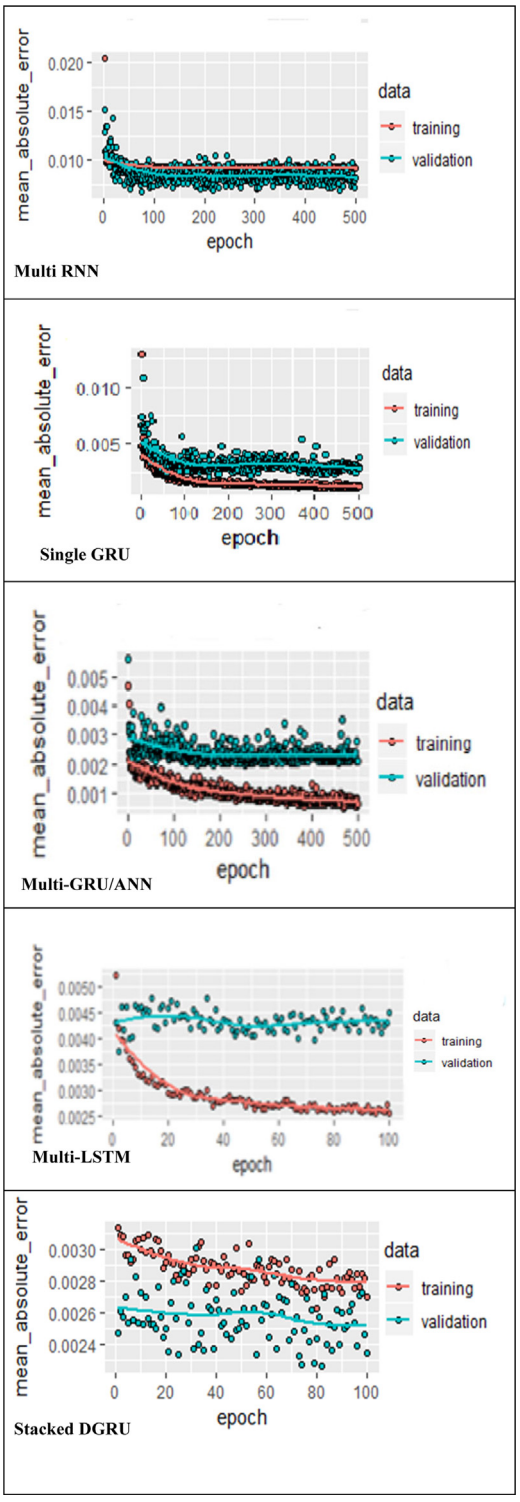

Fig. 5. Training and validation MAE metric.

Table 1
Experimental setup parameters.

Model	Method	Tuning parameters
Multi-RNN	Optimised hyperparameters	Epochs (500) Batch size (128) Learning _rate (0.01)
	Specific hyperparameters	No. of hidden layers [3] No. of units in hidden layers [32,32,32] No. of units in input layer [64]
Single GRU	Optimised hyperparameters	Epochs (500) Batch size (128) Learning _rate (0.001)
	Specific hyperparameters	No. of hidden layers [1] No. of units in hidden layer [64] No. of units in input layer [64]
Multi-G RU/ANN	Optimised hyperparameters	Epochs (500) Batch size (64) Learning _rate (0.0 1)
	Specific hyperparameters	No. of hidden GRU layers [2] No. of units in GRU layer [64,32] No. of hidden ANN layers [1] No. of units in ANN layer [32] No. of units in GRU input layer [60]
Multi-LSTM	Optimised hyperparameters	Epochs (100) Batch size (20) Learning _rate (0.001)
	Specific hyperparameters	No. of hidden LSTM layers [2] No. of units LSTM layer [32,8] No of dropout layers [3] Dropout rate [0.5,0.2,0.5] No. of units in LSTM input layer [64]
DGRU	Optimised hyperparameters	Epochs (100) Batch size (32) Learning _rate(0.0001)
	Specific hyperparameters	No. of hidden GRU layers [2] No. of units in GRU layer [32,20] No. of dropout layers [3] Dropout rate [0.5,0.2,0.5] No. of units in GRU input layer [64]

Table 2
Multi-RNN results for 10 steps prediction.

Subset	loss	RMSE	MAE	RSE	R^2
Train	0.00060	0.02458	0.009143	1.00016	0.00005
Val	0.00041	0.02028	0.009111	1.00022	0.00002
Test	0.00015	0.01292	0.00699	1.22908	0.05523

Table 3
Single GRU results for 10 steps prediction.

Subset	loss	RMSE	MAE	RSE	R^2
Train	0.010	0.00441	0.00181	0.03220	0.96878
Val	0.012	0.00829	0.00511	0.16727	0.83272
Test	0.00012	0.01098	0.00454	0.88785	0.55078

Table 4
Multi-GRU/ANN results for 10 steps prediction.

Subset	loss	RMSE	MAE	RSE	R^2
Train	0.000001	0.00120	0.00037	0.00239	0.99780
Val	0.000050	0.00709	0.00210	0.12227	0.88451
Test	0.000060	**0.00780**	0.00407	0.44747	0.71923

evaluate the feasibility of proposed models, the algorithms learning run time was considered. We trained the machine learning algorithms taking into account two values of hyperparameter epochs (100, 500).

Fig. 7 shows the simulations run time measured in seconds for each learning algorithm. The faster model was DGRU that takes 6, 17 s for epochs (100, 500), respectively. The Multi-LSTM model requires a longer time to train a neural network for both sets of epochs (100, 500) compared with the other algorithms. A number of reasons could affect the speed of the Multi-LSTM. The most significant one is that three gates are used in Multi-LSTM to decide which information in a memory cell should be carried forward. Thus, network converged are relatively slow, particularly when capturing the dependences of long-interval. The optimal epoch's values that give the highest accuracy only provided in the final result as the list in Table 1.

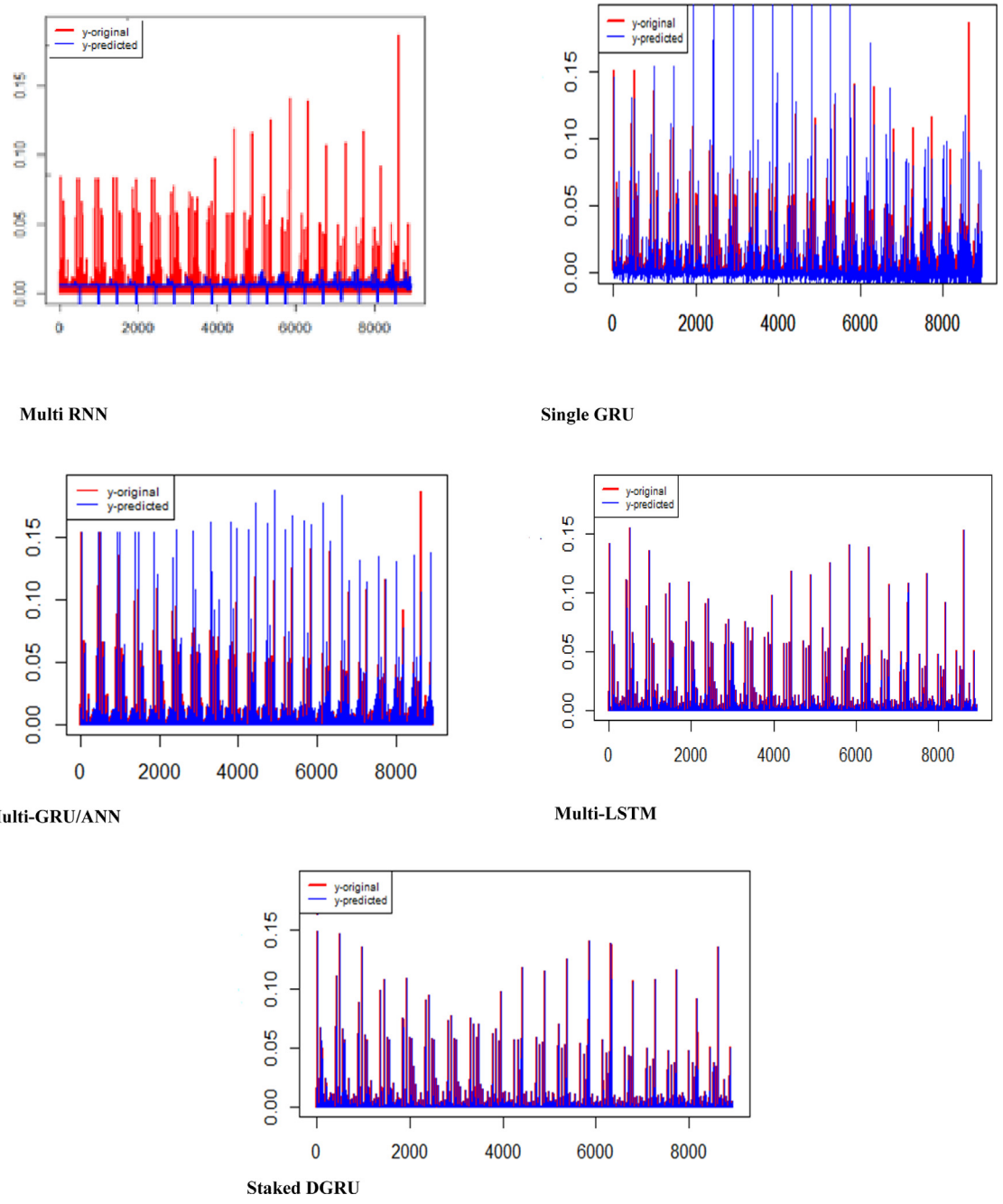

Fig. 6. Actual oil production value vs predicted values.

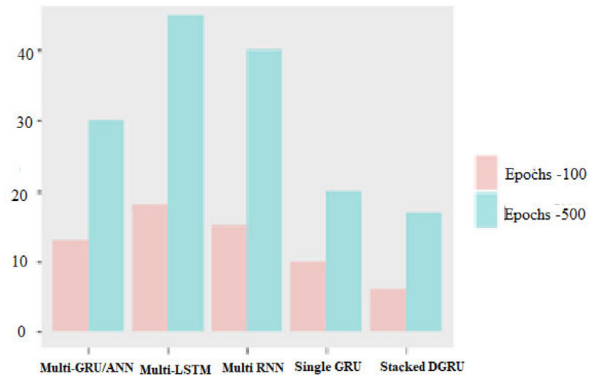

Fig. 7. Computational training time.

Table 5
Multi LSTM results for 10 steps prediction.

Subset	loss	RMSE	MAE	RSE	R^2
Train	0.00009	0.00959	0.00216	0.13299	0.89719
Val	0.00014	0.01219	0.00449	0.35074	0.74595
Test	0.00007	0.00869	0.00332	0.55643	0.51005

Table 6
Stacked DGRU results for 10 steps prediction.

Subset	loss	RMSE	MAE	RSE	R^2
Train	0.00011	0.01088	0.00262	0.15481	0.55031
Val	0.00095	0.03088	0.00872	2.24835	0.03088
Test	**0.00006**	0.00951	**0.00313**	0.66611	**0.70020**

5.1. Discussion

In this work, we aim to evaluate the robustness of the stacked DGRU model against various alternative time series models. The findings demonstrate that the proposed DGRU is a promising model that can be used to solve time series problems, and specifically for long-term forecasting tasks, as it can train long-interval time-series datasets faster than other standard methods.

Our simulation results indicated that regression metric performs poor in test and validation set for single GRU model. This is expected since the structure of the network is quite simple; it has a single hidden layer that consists of (64 neurons). The researchers demonstrate that increasing the number of hidden layers could significantly improve the performance of the network (Ebraheem, Thirumuruganathan, Joty, Ouzzani, & Tang, 2017). Our data is complex and require stacks of hidden layers in order to create more features representation of input; the number of hidden neurons could be another factor that leads to poor performance. The inferring nonlinearity relationship between the oil production, and features could be obtained by increase the number of hidden neurons.

The MAE and RSE in Multi-GRU/ANN model show lower performance for testing; again, it could be relevant to the structure of the network. Although the model is stacked of three hidden layers, these are two GRU layers and single ANN layer. It is incapable of handling change in the underlying time trend for our data. The hidden neurons in GRU layer hold the information from previous time step and feed to ANN layer as input; the hidden nodes in ANN layer transmit the flow of information in forward and pass it to the output layer. Transmitting information in one direction could reduce the efficiency of the network in discovering the entire autocorrelation particularly that are based on time.

With respect to Multi-LSTM model, the MAE yields low performance for the test dataset. It is representing the absolute value of residual often computed by finding the difference between values of model prediction error with the test dataset. Since the MAE considers the absolute value of residual to evaluate error, it contributes partially to the total error. Thus, it is not indicated under-performance or over-performance of the model. Hence, other metrics considered to measure the accuracy of the model. Our empirical results illustrate that the test set gives better performance compared with the trained test. For instance, the model obtains higher RMSE in the test than train and validation set achieving values of 0.00869. The RMSE metric measures the difference between the predicted values and the actual observations. The lowest RSME value demonstrates the better performance of the predictive model.

The DGRU model is based on a stacked deep learning mode of the GRU recurrent neural network. One of the important advantages of using GRU in the oil and gas industry is its capacity to handle the vanishing gradient problem without the need to use a memory unit, meaning that the computational cost is lower than LSTM. Another advantage of GRU is that its structure is less complex than LSTM, since the input and forget gates of LSTM are combined into a single update gate. This gate has mechanisms for deciding how much information should be used to make a prediction. For instance, we have approximately 13,000 instances. If the observation is not relevant to the target, the network will disregard these instances.

Nevertheless, stacking more recurrent layers in the proposed model may be a way to improve the representation of the model, although the computational costs for this could be high. Using GRU layers in a stacked deep learning model gives a significant reduction in the cost of the model.

The results illustrate that the multi-RNN is a simplistic method for forecasting US oil production rates; more specifically, this model is unable to learn the long-term dependencies that cause the vanishing gradient issue, thus preventing the algorithm from learning. Single GRU model is not suitable for this dataset, as the model is overfitting the data from the very beginning. In addition, there was no significant improvement in the MAE for the training and validation sets.

The multi-GRU/ANN system gives the best RSME result. However, this model has limited capacity to capture the states between the inputs as a consequence, such densely connected network would not be able to infer the effectiveness of prior productive oil well in the context of the current well.

LSTM has a complex structure compared to the GRU and the entire sequence of the data should be stored in memory. Moreover, adding the dropout layers does not prevent the overfitting issue, and alternative strategies should be investigated to deal with this.

6. Conclusion

In this study, a novel deep recurrent neural network was constructed. The proposed model can assist petroleum engineers in monitoring the characteristics of productive oil wells. The model is based on the deep structure of a GRU recurrent neural network. We consider an online dataset of observations of the distribution and production of oil and gas wells in the US, which was collected by a company called “Drillinginfo”. The dataset includes oil and gas production data for 90 years across various US states. The numerical and categorical features were both employed in order to forecast USA oil production.

The model is a deep GRU recurrent neural network, an extension of a standard recurrent neural network. The potential for storing information over the long term without a memory unit is one key element of our proposed model. Empirical testing shows that our DGRU can achieve feasible results for a time series forecasting task. The findings also show that the DGRU achieves high computational performance in terms of training time, since each individual recurrent layer of the DGRU model consists only of two gates, namely the reset gate and update gate, making the configuration of our model is much easier than LSTM and RNN. The proposed model shows higher performance than other standard models, and as a consequence, our DGRU model can be applied to the long-term dependencies of a complex time series dataset.

Future directions could involve the use of intelligent AI system in the cloud. Cloud solution has a significant impact on reducing the price of oil and gas operations to its minimum cost. More structure and unstructured dataset will be used to aid petroleum engineer gain deep insight into reservoir characterising, and drilling operations. The Deep Generative Models could be implemented which is capable of extracting high-level representations form high dimension unstructured data.

CRediT authorship contribution statement

Raghad Al-Shabandar: Contributed in exploring online time series dataset and also undertaken the data analysis, Carry out the experiments, Wrote the manuscript, Proposed the predictive models, Revised the manuscript. **Ali Jaddoa:** Discussed the results and commented on the manuscript, Contributed to paper outline format, Supervised the project. **Panos Liatsis:** Contributed to the final version of the manuscript, Evaluate the results, Provided feedback and assist in shaping the research, analysis and manuscript. **Abir Jaafar Hussain:** Supervised the finding of work, Wrote the algorithms, Contributed to the interpretation of the results, Revised the manuscript.

Declaration of competing interest

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

Acknowledgement

Part of this work was support by Liverpool john Moores University, United Kingdom as Ph.D. funded project.

References

Aizenberg, I., Sheremetov, L., Villa-vargas, L., & Martinez-mu noz, J. (2016). Multilayer Neural Network with Multi-Valued Neurons in time series forecasting of oil production. *Neurocomputing*, *175*, 980–989. http://dx.doi.org/10.1016/j.neucom.2015.06.092.

An, J., Mikhaylov, A., & Moiseev, N. (2019). Oil price predictors : Machine learning approach. *International Journal of Energy Economics and Policy*, *9*(5), 1–6.

Anderson, R. N. (2017). 'Petroleum Analytics Learning Machine' for optimizing the Internet of Things of today's Digital Oil Field-to-refinery Petroleum System. In *IEEE international conference on big data* (pp. 4542–4545). IEEE.

Azzedin, F., & Ghaleb, M. (2019). Towards an architecture for handling big data in oil and gas industries : Service-oriented approach. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *10*(2).

Berneti, S. M., & Shahbazian, M. (2011). An imperialist competitive algorithm artificial neural network method to predict oil flow rate of the wells. *International Journal of Computer Applications*, *26*(10), 47–50.

Bikmukhametov, T., & Jäschke, J. (2019). Oil production monitoring using gradient boosting machine learning algorthim. 1(52). (pp. 514–519). http://dx.doi.org/10.1016/j.ifacol.2019.06.114.

Braswell, G. (2013). Artificial intelligence comes of age in oil and gas. *Journal of Petroleum Technology*, *65*(1), 50–57.

Chen, W., Gang, M. A., Junfei, L. I., Zheng, D. A. I., & Jinyuan, L. I. U. (2019). Prediction of corrosion rate of submarine oil and gas pipelines based on IA-SVM model. *IOP Conference Series: Earth and Environmental Science*, *242*(2), Article 022023. http://dx.doi.org/10.1088/1755-1315/242/2/022023.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. ArXiv Preprint arXiv:1406.1078.

Connor, J. T., Martin, R. D., Atlas, L. E., & Ieee, M. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, *5*(2), 240–254.

Crockett, B., & Kurrey, K. (2014). Smart Decision Making Needs Automated Analysis Making sense out of big data in real-time. In *SPE intelligent energy conference & exhibition*. Society of Petroleum Engineers.

Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., & Tang, N. (2017). DeepER–Deep Entity Resolution. 11(11). http://dx.doi.org/10.14778/3236187.3236198.

Eia (2018). U.S. Energy information administration websit. Retrieved from https://www.eia.gov.

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2147–2154).

Feblowitz, J. (2013). Analytics in oil and gas: The big deal about big data. In *SPE digital energy conference*. Society of Petroleum Engineers.

Fetkovich, M. J., Vienot, M. E., Bradley, M. D., & Kiesow, U. G. (1987). Decline curve analysis using type curves: Case histories. *SPE Formation Evaluation*, *2*(4), 637–656.

Frausto-Solís, J., Chi-Chim, M., & Sheremetov, L. (2015). Forecasting oil production time series with a population-based simulated annealing method. *Arabian Journal for Science and Engineering*, *40*(4), 1081–1096.

George, E. P. B., Gwilym, M. J., & Gregory, C. R. (2015). *Time series analysis: Forecasting and control*.

Giles, C. L. E. E., & Lawrence, S. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. 44(1–2). (pp. 161–183).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press, Retrieved from http://www.deeplearningbook.org.

Hammerton, J., Osborne, M., Armstrong, S., & Daelemans, W. (2002). Introduction to special issue on machine learning approaches to shallow parsing. *Journal of Machine Learning Research*, *2*, 551–558.

Hochreiter, S. (1998). Recurrent neural net learning and vanishing gradient. *International Journal Of Uncertainity, Fuzziness and Knowledge-Based Systems*, *6*(2), 107–116.

Hochreiter, S., Bengio, Y., & Frasconi, P. (2011). Gradient flow in recurrent nets : the difficulty of learning long-term dependencies.

Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction : Deep belief networks with multitask learning. 15(5). (pp. 2191–2201).

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2018). LSTM fully convolutional networks for time series classification. *IEEE Access*, *6*, 1662–1669. http://dx.doi.org/10.1109/ACCESS.2017.2779939.

Khan, M., Alnuaim, S., Tariq, Z., & Abdulraheem, A. (2019). Machine Learning Application for Oil Rate Prediction in Artificial Gas Lift Wells. In SPE Middle East Oil and Gas Show and Conference. Society of Petroleum Engineers.

Korjani, M., Popa, A., Grijalva, E., Cassidy, S., & Ershaghi, I. (2016). *A new approach to reservoir characterization using deep learning neural networks*. Society of Petroleum Engineers.

Korovin, I. S. (2017). Use of recurrent and convolutional neural networks for the problem of long term condition prediction for equipment of an oil-and-gas production enterprise. *Transactions on Engineering and Technology Research*, http://dx.doi.org/10.12783/dtetr/eeta2017/7713, (eeta).

Lee, C., & Chang, C. (2008). New evidence on the convergence of per capita carbon dioxide emissions from panel seemingly unrelated regressions augmented Dickey – Fuller tests. *Energy*, *33*, 1468–1475. http://dx.doi.org/10.1016/j.energy.2008.05.002.

Ma, X., & Liu, Z. (2018). Predicting the oil production using the novel multivariate nonlinear model based on Arps decline model and kernel method. *Neural Computing and Applications*, *29*(2), 579–591.

Masoudi, P., Arbab, B., & Rezaei, H. (2014). Net pay determination by artificial neural network : Case study on Iranian offshore oil fields. *Journal of Petroleum Science and Engineering*, *123*, 72–77.

Mohaghegh, S. D. (2005). Recent developments in application of artificial intelligence in petroleum engineering. *Journal of Petroleum Technology*, *57*(4), 86–91.

Mohammadpoor, M., & Torabi, F. (2019). Big Data analytics in oil and gas industry : An emerging trend. *Petroleum*, http://dx.doi.org/10.1016/j.petlm.2018.11.001.

Moreno, J. J. M. (2011). Artificial neural networks applied to forecasting time series. 23(2). (pp. 322–329).

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1).

Ockree, M., Brown, K. G., Frantz, J., Deasy, M., & John, R. (2018). Integrating big data analytics into development planning optimization. In *SPE/AAPG Eastern regional meeting*. Society of Petroleum Engineers.

Sagheer, A., & Kotb, M. (2019). Time series forecasting of petroleum production using deep LSTM. *Neurocomputing*, (323), 203–213. http://dx.doi.org/10.1016/j.neucom.2018.09.082.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. (pp. 1–21).

Ferreira da Silva, L. C., Portella, R. C., Emerick, A. A., & Favilla Ebecken, N. F. (2007). An imperialist competitive algorithm artificial neural network method to predict oil flow rate of the wel. In *Latin American & caribbean petroleum engineering conference*. USA: Society of Petroleum Engineers.

Singh, A., & Pateriya, R. (2019). Artificial Oil Lift production forecasting and analysis using Autoregressive and Deep learning Models.

Sivarajah, U., Kamal, M. M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*, *70*, 263–286.

Song, K., Saraf, D. N., & Gupta, M. M. (2013). Production forecasting of petroleum reservoir applying higher-order neural networks (HONN) with limited reservoir data. 72(2). (pp. 23–35).

Tian, Y., & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. In *IEEE international conference on smart city*. IEEE, http://dx.doi.org/10.1109/SmartCity.2015.63.

U. S. Energy Information Administration (2019). U.S. distribution and production of oil and gas wells. Retrieved from https://openei.org/datasets/dataset/u-s-distribution-and-production-of-oil-and-gas-wells/resource/91e504bb-f128-40c5-b106-53f851db2ef3.

Weninger, F., Bergmann, J., & Schuller, B. (2015). Introducing CURRENNT: The Munich open-source CUDA RecurREnt neural network toolkit. *Journal of Machine Learning Research*, *16*, 547–551.

Yu, L., Dai, W., & Tang, L. (2016). A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting. *Engineering Applications of Artificial Intelligence*, *47*, 110–121. http://dx.doi.org/10.1016/j.engappai.2015.04.016.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks : The state of the art. *International Journal of Forecasting*, *14*(1), 35–62.

Zhao, L. T., Zeng, G. R., Wang, W. J., & Zhang, Z. G. (2019). Forecasting oil price using web-based sentiment analysis. 12(22). (p. 4291).