UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

UNCERTAINTY ANALYSIS OF PRODUCTION FORECAST IN SHALE SYSTEMS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN NATURAL GAS ENGINEERING AND MANAGEMENT

By

ADEMIDE O. MABADEJE
Norman, Oklahoma
2019

UNCERTAINTY ANALYSIS OF PRODUCTION FORECAST

IN SHALE SYSTEMS

A THESIS APPROVED FOR THE
MEWBOURNE SCHOOL OF PETROLEUM AND GEOLOGICAL ENGINEERING

BY

Dr. Rouzbeh Moghanloo, Chair

Dr. Ashwin Venkatraman

Dr. Deepak Devegowda

Dedication

To my family and friends for their unconditional love and support

Acknowledgements

My journey on attaining this degree would not have been possible without the guidance, encouragement, and continuous support of my advisor, Dr. Rouzbeh Ghanbarnezhad Moghanloo. Dr. Rouzbeh exposed me to several research projects, mentored and taught me the value of perseverance. Thank you very much for everything.

I would like to thank my thesis committee members: Dr. Deepak Devegowda and Dr. Ashwin Venkatraman. I appreciate their invaluable feedback and contributions to my research. Through probing questions and suggestions, they ensured I addressed challenging research issues in a rigorous manner. Without them, my research would not be what it is.

I am grateful to the Mewbourne School of Petroleum and Geological Engineering for granting me scholarships and travel awards to support my studies and research presentations. I appreciate the efforts made to make this a home away from home for me.

To my family: my dad and mom; Gbenga & Feyisara Mabadeje, my siblings; Yinka, Peju, Bomi, and Remi thank you for encouraging me in all my pursuits and supporting me to follow my dreams, even when it took me thousands of miles away. This acknowledgement will be incomplete if I do not recognize Dr. Davud Davudov whose insight, analytical contributions and friendship was invaluable. Special mentions to my loved ones who played different roles during this period in different capacities: Tobi Ore, Dolly Na-Yemeh, Oluwatobi Olorunsola, Precious Jatau, Pascoela da Silva, Femi Adekomaya, Jing Fu, Folasade Ayoola, Shams El Fouly, Omolola Eniodunmo. To those whom I have not mentioned but contributed in one way or the other. Thank you.

Above all, I owe all my gratitude to God; for all that I am and hope to be.

Table of Contents

List of Tables

List of Figures

List of Symbols

$q(t)$	Production flowrate at time *"t"*
q_i	Initial flowrate
D_i	Initial nominal decline rate
$\frac{1}{D(t)}$	Loss ratio
b	Arps' hyperbolic decline exponent ranging between 0 and 1
$\widehat{D}_i$	Decline coefficient
$\hat{n}$	Time exponent
D_∞	Terminal decline coefficient
τ	Characteristic time constant
q_1	Production flowrate at day 1
G_p	Cumulative gas production
m_{Dng}	Slope of log-log plot q/G_p versus time
a_{Dng}	Intercept coefficient of log-log plot q/G_p versus time
β_e	Constant attributed to early life period of the well
β_l	Constant attributed to late-life period of the well
$\hat{b}$	Bias i.e. difference between actual and predicted observations
θ	True parameter i.e. actual parameter
$E[\hat{\theta}]$	Estimator of θ

List of Abbreviations

DCA Decline curve analysis

EUR Estimated ultimate recovery

PLE Power law exponential decline curve model

SE Stretched exponential decline curve model

DNG Duong decline curve model

EE Extended exponential decline curve model

ED Expected disappointment

Abstract

This research evaluates the impact of decision making and uncertainty associated with production forecast in shale oil and gas wells; over 11000 wells completed in the Barnett & Haynesville plays and more than 2000 wells from the Permian Basin. Existing studies show that unconventional reservoirs have complex reservoir characteristics making traditional methods for ultimate recovery estimation insufficient. Based on these limitations, uncertainty is increased during the estimation of reservoir properties, reserve quantification and, evaluation of economic viability. Thus, it is necessary to determine and recommend favorable conditions in which these reservoirs are developed.

In this study, cumulative production is predicted using four different decline curve analysis (DCA) – power law exponential, stretched exponential, extended exponential and Duong models. A comparison between the predicted cumulative production from the models using a subset of historical data (0-3months) and actual production data observed over the same time period determines the accuracy of DCA's; repeating the evaluation for subsequent time intervals (0-6 months, 0-9 months,..) provides a basis to monitor the performance of each DCA with time. Moreover, the best predictive models as a combination of DCA's predictions is determined via multivariate regression. Afterwards, uncertainty due to prediction errors excluding any bias is estimated and expected disappointment (ED) is calculated using probability density function on the results obtained.

Using these results, uncertainty is estimated from the plot of ED versus time for all wells considered. ED drops for wells having a longer production history as more data are used for estimation. Also, the surprise/disappointment an operator experiences when using various DCA methods is estimated for each scenario. However, it appears that power law exponential serves as

the lower boundary of the forecast in the formations considered, whilst the upper boundary switches between stretched exponential (SE) and Duong (DNG) method. The extend exponential DCA model was found to demonstrate an erratic behavior crossing over actual trends multiple times with time.

In conclusion, profitability zones for producing oil in the Permian basin are defined implicitly based on drilling and completion practices which paves the path to determine the "sweet spot" via optimization of fracture spacing and horizontal length in the wells. Also, it can be inferred that the decline rate during production is somewhat related to pore connectivity and it could be a good qualitative indicator of wells in which EOR might be successful although it needs to be investigated further.

The outcome of this research work helps improve the industry's take on uncertainty analysis in production forecast, especially the concept of expected disappointment/pleasant surprise. This study suggests that effects of bias and ED due to decision making can be much greater than what has often regarded; ranging from 0.41 to 0.86, which can change the performance evaluation of shales in terms of economic feasibility.

Chapter 1: Introduction

Importance of Research

Existing studies show that unconventional oil and gas reservoirs have complex reservoir characteristics making traditional methods for ultimate recovery estimation insufficient. Due to these reservoir complexities, uncertainty is increased during the estimation of reservoir properties, reserve quantification and, evaluation of economic viability. It is therefore necessary to determine and recommend favorable conditions in which these reservoirs are developed. The motivation of which owes primarily to the challenge posed by existent reservoir production forecast uncertainties and volume prediction inaccuracies, thus resulting in sub-par field development and planning which are the key drivers of economics in the overall oil and gas production process.

Irrespective of the perception of profitable income in the petroleum industry, various researchers have noticed that the oil and gas industry is performing routinely less than its expectation. This poor performance is generally ascribed to inferior project evaluation and selection due to lingering bias (McVay & Dossary, 2014). Knowing the normal rule for choosing between alternatives in a decision-making situation is to select the option with the maximum estimated value. Due to uncertainty, choosing the maximum induces a form of bias – a systematic bias which when repeated during decision making reduces the value of the estimated expected values obtained. Although some instances of this behavior: post-decision surprise, the optimizer's curse, inevitable disappointment have been reported, its relevance in oil and gas decision making situations is not well established (Begg & Bratvold, 2008).

Disappointment is a psychological reaction to an outcome that does not match up to expectation. The greater the disparity, the greater the disappointment as seen with optimistic

models and/or scenarios. If the alternative occurs, then there is an elation leading to pleasant surprise which occurs when the outcome exceeds expectations.

Bell (1985) considered the effect of disappointment and elation on decision making under uncertainty and found that if an expected incentive is not attained, a decision maker is disappointed. Likewise, if an unexpected incentive is gained there would be an elation "pleasant surprise". Hence, the satisfaction achieved is dependent on prior expectation which is usually made by the decision maker. The higher the expectations, the higher the likelihood of disappointment. This begs the question on how to quantify risk aversion and affinity i.e. how conservative/pessimistic or optimistic/over-confident a model is.

Current methods for risk assessment try to allow for variability and uncertainty by using fixed safety factors as seen in different models formulated over the years. However, this fails to give a complete description of the full range of the possible risks or quantify the uncertainty associated with it, as seen in the oil and gas business. Also, it is difficult to decide how big the safety factors should be. This thesis focuses on eliminating disappointment in order to account for these uncertainties, and the best way to do this is through a process of tracking probabilistic predictions and comparing how actual performance turns out relative to the predictions made, and then adjust subsequent forecasts by using this calibration information (Capen, 1976). This would provide a better basis for making decisions about risks in producing or not producing a well because the full range of possible outcomes can be considered.

Hypothesis

This work hypothesizes that inclusion of uncertainty analysis into production forecast makes the outcome of decline curve analysis more objective; in other words, utilizing the disappointment index can help understand how the dynamics of the prediction error evolves.

Research Objectives

This work aims to integrate concepts from economics, psychology, and petroleum engineering to create a resultant concept - expected disappointment, that adds objectivity to decline curve analysis (DCA) which is naturally a subjective technique used in oil and gas reserve estimation.

The objectives for this study are to:

- Predict the production performance using various DCA methods;
- Identify the most optimistic and the most pessimistic method;
- Introduce objectivity to production forecasting models- decline curve analysis which is a naturally subjective technique;
- Evaluate the effect of bias due to decision making on EUR estimation;
- Incorporate the concept of expected disappointment/pleasant surprise into production forecasting models.

Chapter 2: Literature Review

Decline curve analysis (DCA) models in simple terms are regressions for historical production data. This is one of the most commonly used techniques in the petroleum industry to estimate ultimate recovery (EUR) for producing wells. This section provides an overview of classical decline curve analysis as a means of forecasting production data in conventional plays and modern techniques developed specifically to address production forecasting challenges in unconventional reservoir systems.

Over the years, there exist many production estimation techniques including several variations of decline curve analysis (DCA), analytical and numerical simulation methods. Each one of these methods has its own advantages and disadvantages, but only the DCA methods can use available production data to forecast quickly and, to some extent, accurately.

In conventional reservoirs, the Arps (1945) model synthesizes previous work by defining exponential and hyperbolic declines in mathematical terms for both rate versus time and rate versus cumulative production expressions using the loss ratio concept. The rate versus time exponential decline is given as:

$$q(t) = q_i exp[-D_i t] \quad \text{.....(1)}$$

where q_i is the initial flowrate and D_i is the initial nominal decline rate. The hyperbolic decline is given as:

$$q(t) = \frac{q_i}{(1+bD_it)^{1/b}} \quad \text{.....(2)}$$

where b is the hyperbolic decline exponent ranging between 0 and 1. If $b = 1$ in **Eq. 2** then, the expression for the harmonic decline is expressed as:

$$q(t) = \frac{q_i}{(1+D_it)} \quad \text{.....(3)}$$

The Arps model assumes that for a flowing well the bottomhole pressure and the skin factor is constant, and the flow regime is boundary dominated flow. Although the Arps model is simple and fast, it fails to accurately fit the decline curve of unconventional reservoirs and predict the estimated ultimate recovery (EUR) as it overestimates the EUR for shales because it assumes that a boundary dominated flow regime exists. Knowing that most shale wells take a long time to reach the boundary dominated flow regime, the Arps model cannot be applied as is without significant modifications.

Based on these limitations, various models such as; power law exponential, stretched exponential, extended exponential etc. have been developed for unconventional reservoirs. However, failure to choose the best fit DCA method in each shale play for a particular period may lead to wrong EUR estimation which increases technical uncertainty and in turn, greatly affects profitability.

Lewis & Beal (1918) recognized the importance of developing a dependable and easily applied forecasting tool for estimating reserves for a field. The authors observed that the

percentage decline rate was constant when plotted versus time and if plotted on Cartesian and logarithmic coordinates, power-law and straight-line trends were exhibited respectively.

Cutler (1924) later provided an extensive overview of production forecasting techniques which was similar to Lewis & Beal, he observed that the percentage decline rate was often variable which contrasted with their findings. Thus, concluding that the decline rate could be modelled by a hyperbolic mathematical equation using a trial and error method on production data until a straight-line trend is established.

Johnson & Bollens (1927) introduced the concept of the loss-ratio and its derivative. They were the first authors to give the observations found by prior researchers a mathematical context as shown below:

$$\frac{1}{D(t)} = -\frac{q(t)}{dq(t)/dt} \qquad \text{.....(4)}$$

where $\frac{1}{D(t)}$ is the loss ratio, $q(t)$ is the flowrate, and t is the production time. By extension, the derivative of the loss ratio is defined as:

$$b(t) = \frac{d}{dt}\left[\frac{1}{D(t)}\right] = -\frac{d}{dt}\left[\frac{q(t)}{dq(t)/dt}\right] \qquad \text{.....(5)}$$

where b is the derivative of the loss ratio.

The revelations from Johnson & Bollens (1927) provided the foundation for the classical DCA model proposed by Arps (1945) as seen in **Eq. 1, 2, 3**. Based on these findings, various new models were proposed to account for the Arps' limitation as seen in Maley (1985) which

demonstrated that a b value greater than unity could be obtained as it fits production data from tight gas well. This observation while empirical, was the first to address the applicability of classical decline curve analysis for unconventional reservoirs. The author noted that due to the unbounded nature of the mathematical model, unreasonable estimates of ultimate recovery (EUR) could be obtained.

Rushing et al. (2007) carried out further studies on the forecasting of production data in tight sandstone reservoirs where it was observed that extrapolating rates during the transient flow using an unbounded hyperbolic equation (b>1 from the Arps equation) significantly overestimated the EUR. These findings were further explained in Lee & Sidle (2010) which discussed the significance of the observations and how it relates to reserve estimation.

Recently, Ilk et al (2008, 2009) derived the power law exponential (PLE) decline model which assumes that the b-factor trend declines as a function of time to properly model fracture dominated flow in low permeability reservoirs. This model is developed specifically for shale gas wells and it claims to be capable of efficiently modeling the transition period and boundary dominated flow.

Its expression is given as:

$$q(t) = q_i exp\left[-\widehat{D}_i t^{\hat{n}} - D_\infty t\right] \quad \text{.....(6)}$$

where $\widehat{D}_i$ is the decline coefficient, $\hat{n}$ is the time exponent, and D_∞ the terminal decline coefficient.

Alternatively, Valko (2009) introduced the stretched exponential (SE) decline model that determines the actual production decline by a great number of contributing volumes which have exponential decay rates, but with a specific distribution of characteristic time constants. The

difference between this model and that proposed by Ilk et al. (2008) is the removal of the D_∞ from the stretched exponential decline. The rate-time formulation for the SE model is given in **Eq. 7**:

$$q(t) = q_i exp\left[-(t/\tau)^{\hat{n}}\right] \quad \text{.....(7)}$$

where $\hat{n}$ is the exponent, τ is the characteristic time constant.

Duong (2011) developed the Duong decline curve model (DNG) to describe the long-term linear flow performance associated with unconventional reservoirs. This model is based on the log-log plot of q/G_p (G_p is the cumulative gas production) versus time forms a straight line, which is an empirically derived rule. The model is written as follows:

$$q(t) = q_i t^{-m_{Dng}} exp\left[\frac{a_{Dng}}{(1-m_{Dng})}\left[t^{(1-m_{Dng})} - 1\right]\right] \quad \text{.....(8)}$$

where m_{Dng} is the slope of the log-log plot, and a_{Dng} is the intercept coefficient.

Zhang et al. (2016) proposed a new method called the extended exponential (EE) decline curve analysis which captures both the early and late production profile. The rate-time expression is written as:

$$q = q_i exp\left[-\beta_l t - \beta_e t e^{-t^n}\right] \quad \text{.....(9)}$$

where β_l and β_e are constants accounting for the late-life period and the early period respectively. It should be noted that in early time, the term $\beta_e t e^{-t^n}$ has a governing impact, but over time it reduces and the impact of $\beta_l t$ term increases.

Chapter 3: Methodology

In this section, shale oil production from the Permian basin and shale gas produced from the Barnett & Haynesville plays were analyzed using decline curve analysis. Individual wells from 2010 up to 2017 was studied to determine the first- and second-years production, bias involved, estimated ultimate recovery (EUR) and expected disappointment/pleasant surprise associated. The following assumptions were made for all analysis carried out in this research work:

a) Wells chosen for analysis were void of rapid fluctuations in production rates resulting from completion practices, well shut-in periods leading to noisy data and varying decline rates or worse inaccurate forecasts.
b) An average of 3 data points i.e. 3 months' worth of production data was removed due to flowback in each well.
c) No more than 10% of the wells were multi-lateral wells, and the rest produced from a single pad for the shale oil system. Hence, a mean lateral was chosen as an equivalent lateral length for multilateral wells.

Based on the various assumptions governing the DCA models considered; power law exponential, stretched exponential, Duong model, and extended exponential as highlighted in the literature review were used for analysis. Following Mabadeje & Ghanbarnezhad Moghanloo (2019a) and Mabadeje & Ghanbarnezhad Moghanloo (2019b) work on uncertainty quantification of production forecast in oil and gas systems using a three-step procedure – data preparation, workflow and data analysis of wells in the Permian basin, the ensuing procedures were followed.

Step 1: Data Preparation

For the shale gas system, production data was obtained from www.info.drillinginfo.com for a total of 12000 active wells from both the Barnett and Haynesville shale plays. Also, for the shale oil system, 3000 active wells were selected from the Permian. The data obtained was filtered based on the following criteria:

a) Production history with enough time length,

b) Fluctuations in production rates resulting from flowback in the first 2 to 4 months of production was corrected for using signal processing techniques shown in **Figure 1**

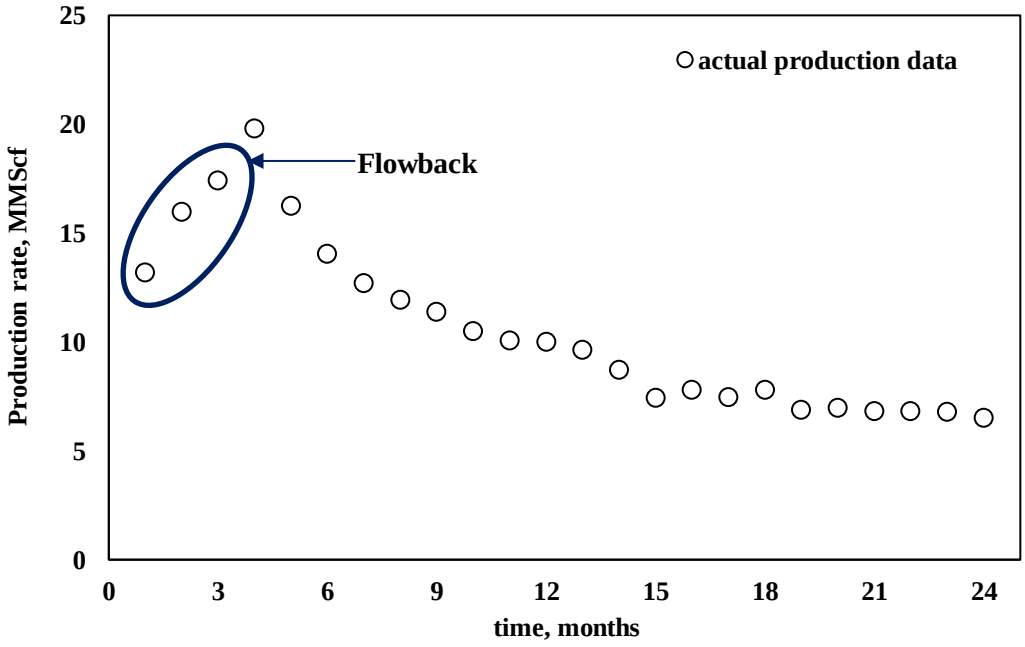

Figure 1: Sample of flowback portion removal in production forecast for well ID-103004447 in the Barnett

c) Cluster regions in the areas of interest were chosen as a sample representation of the entire play as shown in **Figures 2**, **3**, and **4**. Afterwards, over 11000 gas wells and more than 2000 oil wells with good production data are selected for further analysis.

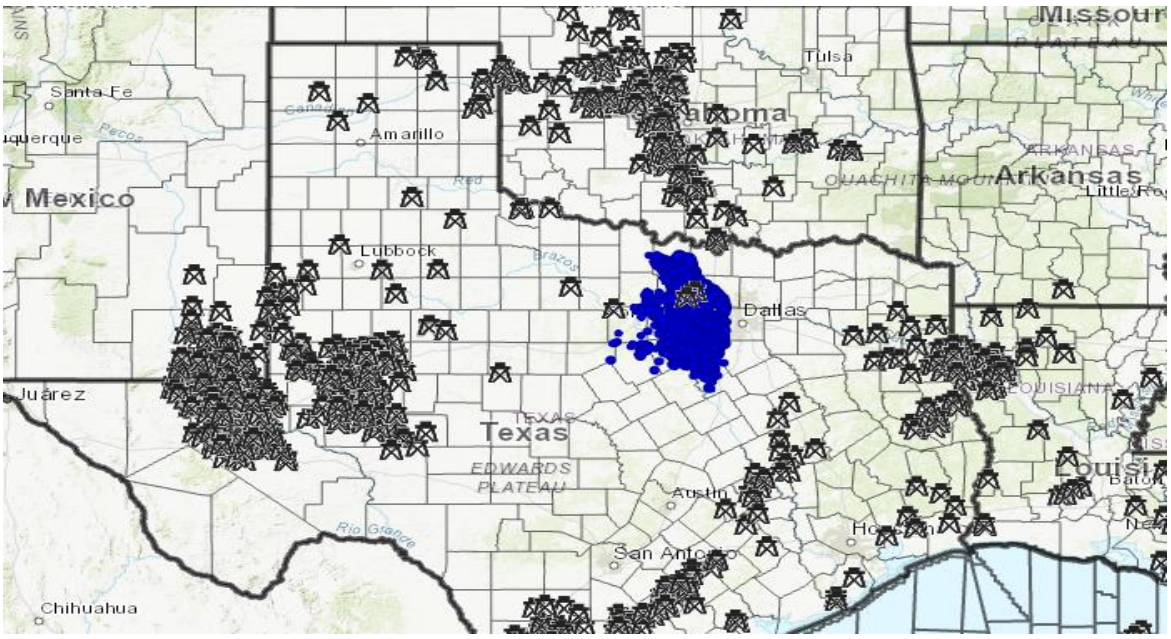

Figure 2: Wells in Barnett shale play and its location
Source: https://info.drillinginfo.com

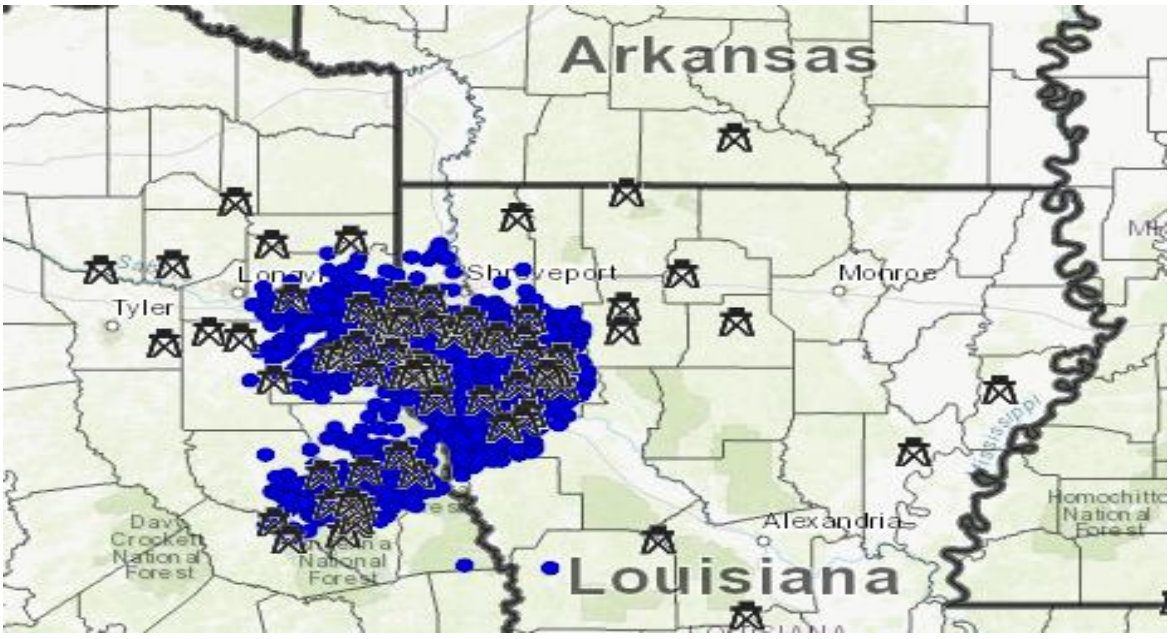

Figure 3: Wells in Haynesville shale play and its location
Source: https://info.drillinginfo.com

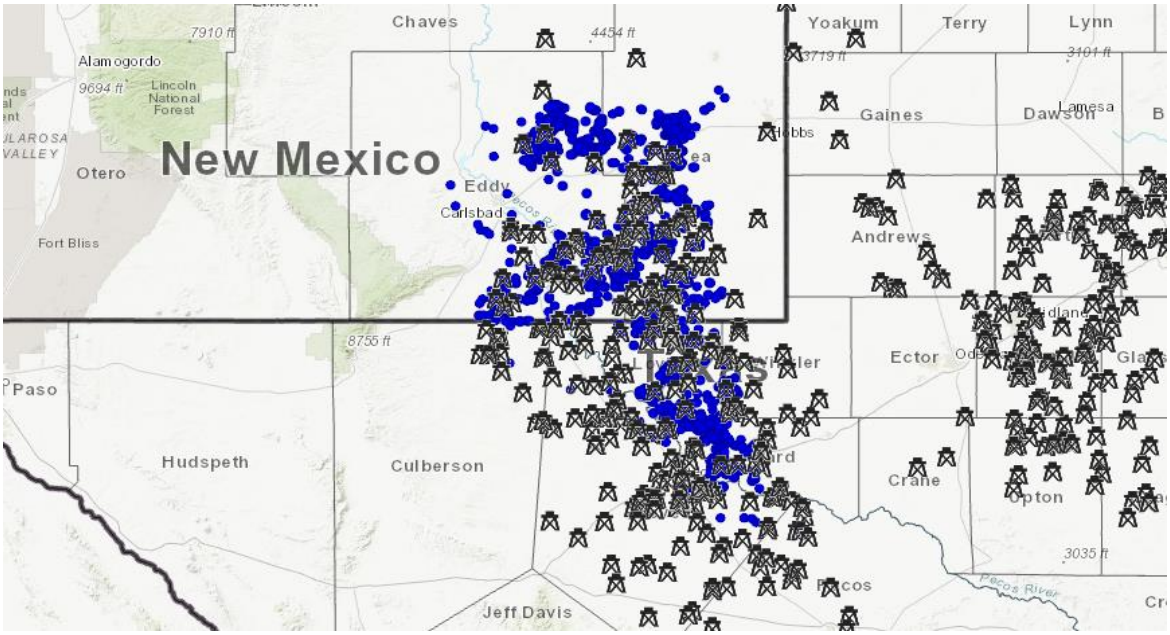

Figure 4: Wells in the Permian basin and its location
Source: https://info.drillinginfo.com

Step 2: Workflow for PLE, SE, DNG, and EE models

a) Production data from the results in the first step (a-c) is inputted per well name, date and production rate into the R code generated (*See Appendix A*) starting at three months' interval.

b) Using the non-linear least squares method on real production data, the variables; $\widehat{D}_i$, $\hat{n}$, and D_∞ from **Eq. 6**, $\hat{n}$, τ, from **Eq. 7**, a_{Dng} , m_{Dng} from **Eq. 9**, n, β_l, and β_e from **Eq. 8** are determined as outputs.

c) Predicted production flowrate values for all models were generated by inputting the output variables obtained in step "b" into **Eq. 6, 7, 8, & 9** for the entire data set whilst forecast and analysis were carried out for a certain time interval as seen in **Figures 5** & **6**.

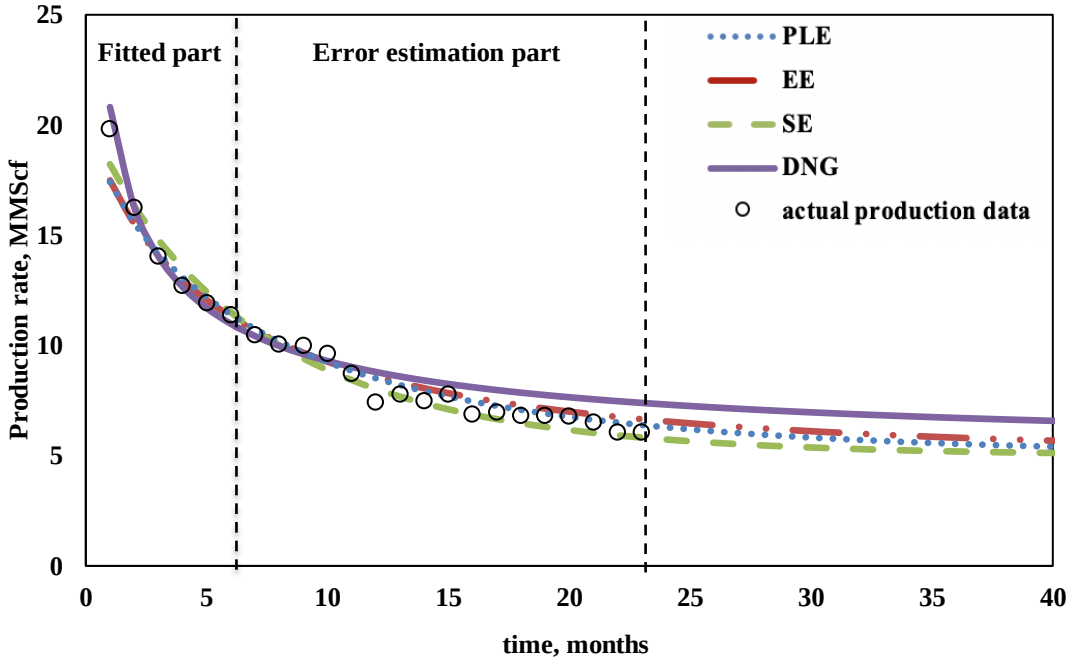

Figure 5: Sample of production forecast obtained from well ID-103004447 in the Barnett for 6 months interval

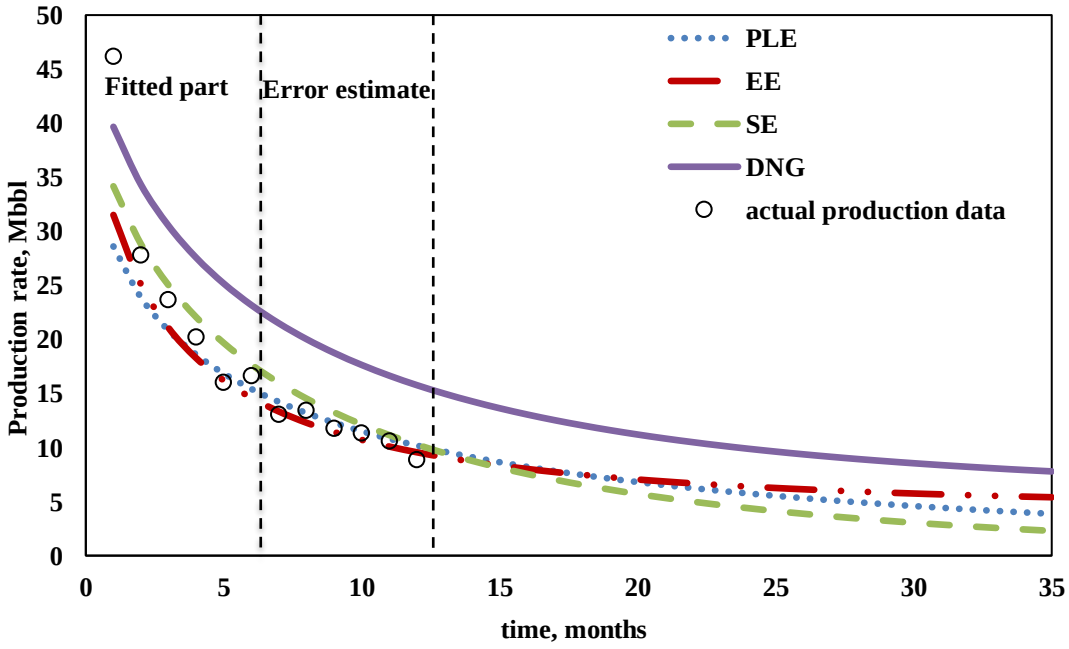

Figure 6: Sample of production forecast obtained from a well in the Permian for 12 months interval

d) The forecasted data was compared with actual production data obtained from Drillinginfo then, observed over the same time period and the percentage error was estimated.

e) Steps (a to d) was repeated for the different time intervals; 6, 9, and 12 months to determine the best-fit model.

Step 3: Data Analysis

a) The bias, $\hat{b}$ was determined using Eq. 10 and uncertainty was quantified using a probability density function consisting of the predicted production flowrates from the DCA models and actual production flowrates coupled with the prediction errors quantified in step 2 – workflow to calculate the expected disappointment for PLE, EE, SE, and DNG DCA's during time intervals 3, 6, 9, and 12 months.

$$\hat{b} = E\left[\hat{\theta}\right] - \theta \tag{10}$$

where $\hat{b}$ is the bias, $E\left[\hat{\theta}\right]$ is the estimator of θ and θ is the true parameter

Due to unavailability of matching completion data such as: injected fluids, amount of proppant used, and lateral lengths for selected wells in Barnett and Haynesville plays from FracFocus. The following procedure was carried out specifically on the shale oil system; wells from the Permian:

b) The estimated ultimate recovery (EUR) was determined for each DCA in each time interval using its corresponding production history. (*See Appendix B*) Oil production was used as a proxy for income generated by each well. So, the EUR obtained was categorized using the

following classification: very poor (0 – 50Mbbl), poor (50 – 100Mbbl), good (100 – 200Mbbl), and excellent (> 200Mbbl) to determine the best measure of well profitability given since there was no access to economic data.

c) The EUR of the classified wells was obtained for all DCA models then normalized using its respective lateral length and plotted versus amount of proppant used per foot.

d) These statistics were then transformed using a suitable scale using logarithms in order to create an approximate normal distribution of the variable i.e. EUR/ft and proppant/ft as shown in **Figures 7** & **8**.

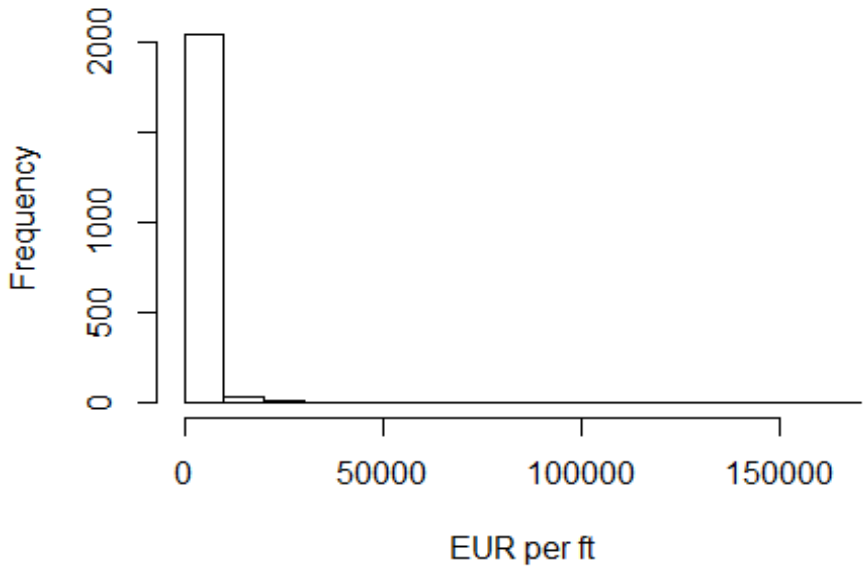

Figure 7: Sample of skewed statistic - EUR/ft using PLE DCA

$$a = \log(b) \quad \text{.....(11)}$$

$$b_{transformed} = \frac{a-\mu_a}{\sigma_a} \quad \text{.....(12)}$$

where: b is the original variable in the data set that needs scaling i.e. EUR/ft from any DCA method and proppant/ft, a is the transformed variable used for clustering, μ_a and σ_a is the mean and standard deviation of a respectively.

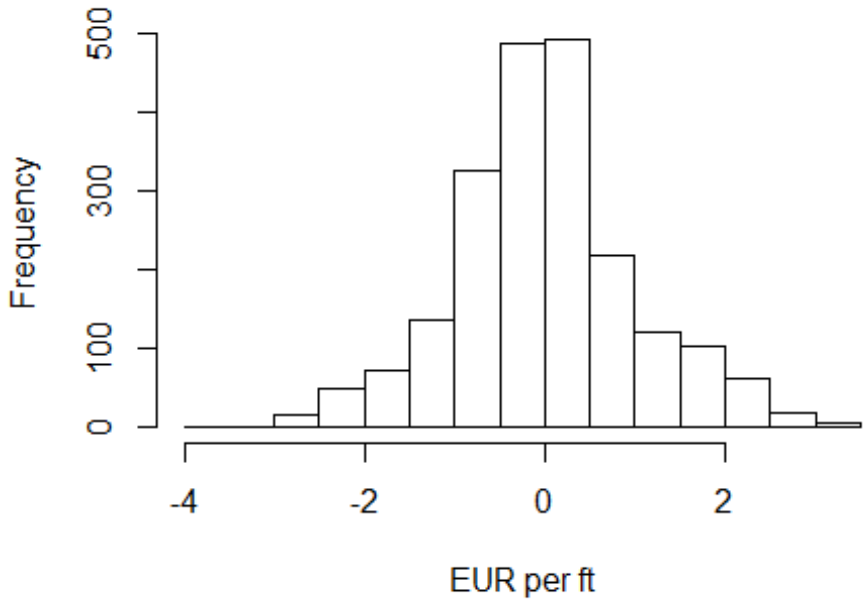

Figure 8: Sample of approximate normal distribution of EUR/ft statistic using PLE DCA

e) Afterwards, density-based clustering was applied to smooth over the data sets for 12 months' time interval and hierarchical clustering was applied to find the number of clusters and inadvertently determine the sweet spot.

Chapter 4: Results & Discussion

By applying the procedure explained in chapter 3, further analysis was carried out on the data sets and the following results were obtained for the shale oil and shale gas systems.

Shale oil system

The results from the shale oil system consisting of over 2000 wells drilled and completed in the Permian basin are explained in threefold as shown in each subsection.

Section 1

From **Figure 9**, it is seen that as the time interval increases, the standard deviation (SD) of the bias involved in each prediction decreases.

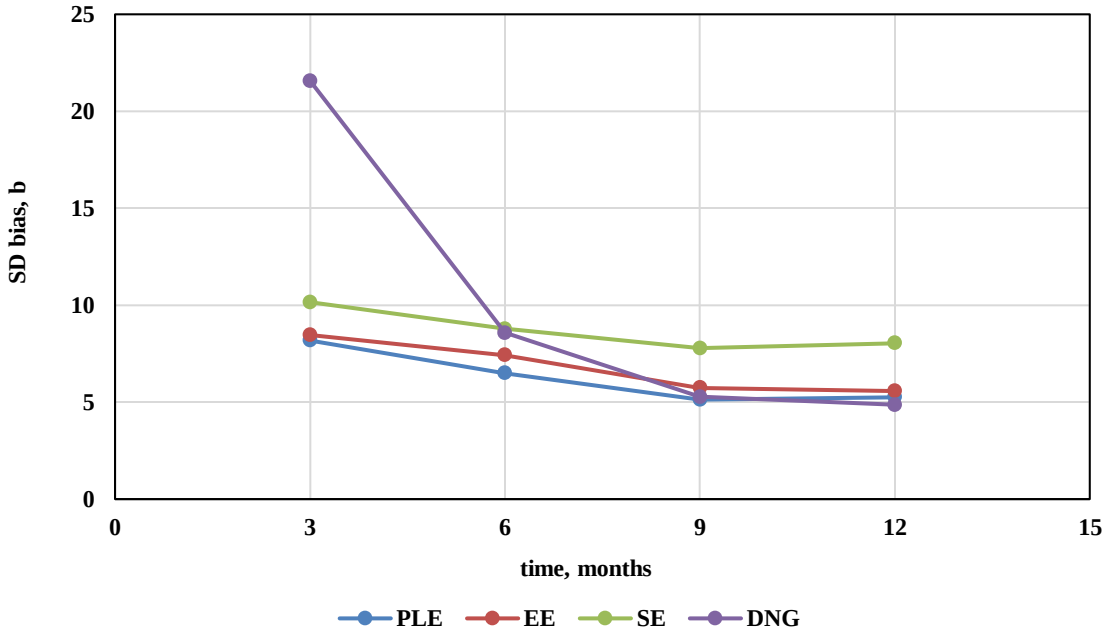

Figure 9: Standard deviation of bias involved in all DCA's in the Permian

To fully understand the effect of the bias in the production forecast using different decline curve analysis models **Figures 10 & 11** shows the standard deviation and average of the percentage error obtained during prediction respectively.

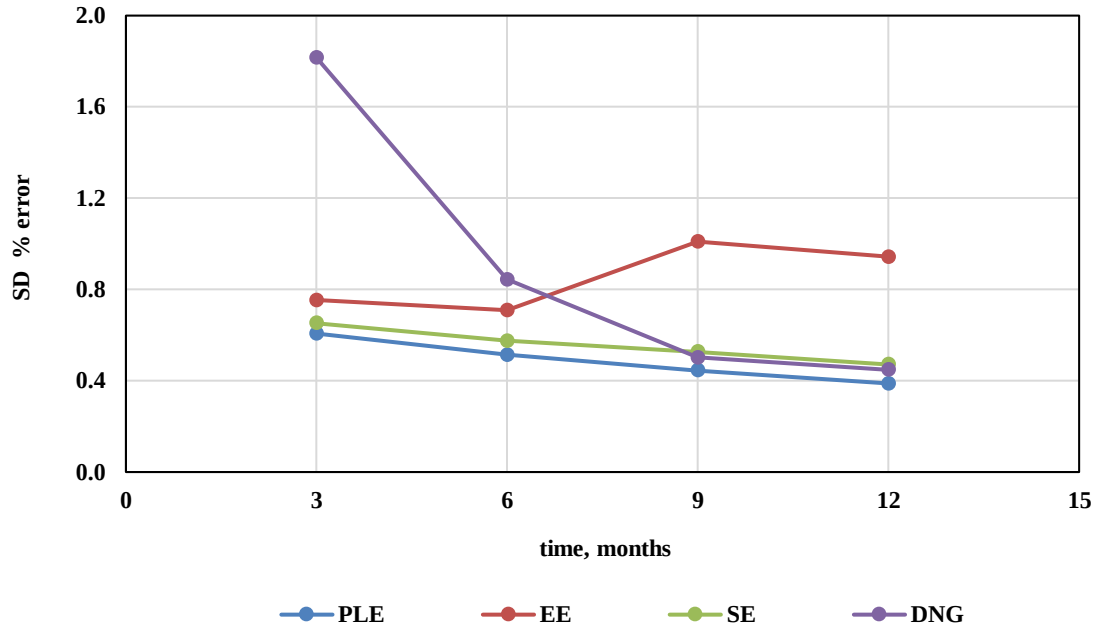

Figure 10: Standard deviation of average errors for all DCA models in the Permian

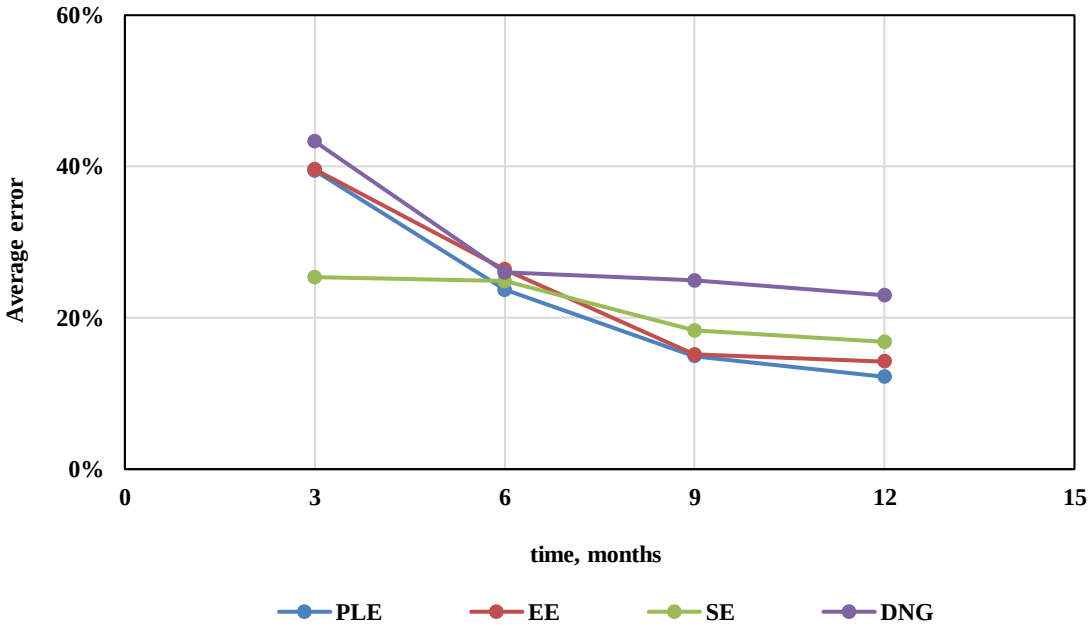

Figure 11: Average errors for all DCA models in the Permian

Looking at both schematics simultaneously shows that SE is the best-fit model for production of oil during the first three months of forecast which coincides with the findings of Akbarnejaj-Nesheli et al. (2012) and Zuo et al. (2016) whose work suggests that SE predicts the transient flow regime rather than boundary dominated flow. However, in the 6, 9, and 12 months' time interval there is a switch in the DCA model from PLE to EE then back to PLE. The crossing over of these models gives the evidence that one DCA cannot be used to forecast the production rate continuously for the Permian and this is expected to hold true for other formations.

Section 2

The results from section 1 shows that as the standard deviation of the errors decrease, the expected disappointment (ED) associated with each DCA model in **Figure 12** decreases. This follows the conclusion from Begg & Bratvold (2008) that as the standard deviation of error of any value considered in their case; Net Present Value decreases, its ED decreases simultaneously for a scenario under evaluation.

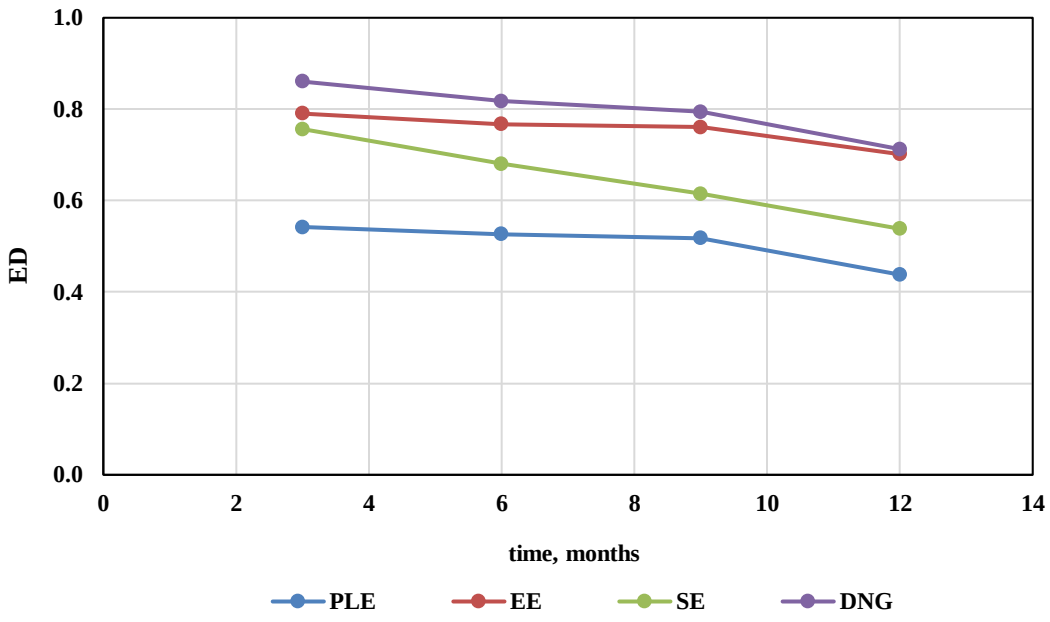

Figure 12: Expected disappointment for all DCA's in the Permian

Also, in **Figure 12** PLE has the lowest ED values (0.54 – 0.44) over the time intervals indicating that it is the closest to reality although it also overpredicts the production flowrate while the ED for Duong's DCA (0.86 – 0.71) is significantly higher and it erroneously overpredicts the production rates due to its inherent bias shown in section 1. Thus, making the power law exponential and Duong's decline curve methods serve as the lower and upper extremities of the predictions whilst the EE and SE methods lie between spectrum with ED values ranging between (0.79 – 0.70) and (0.76 – 0.54) respectively.

From **Figure 13** PLE has the highest PS values (0.39 – 0.28) indicating that it is a very pessimistic model while that of Duong method (0.06 – 0.13) suggests that it is an optimistic model which is buttressed by its severe overprediction.

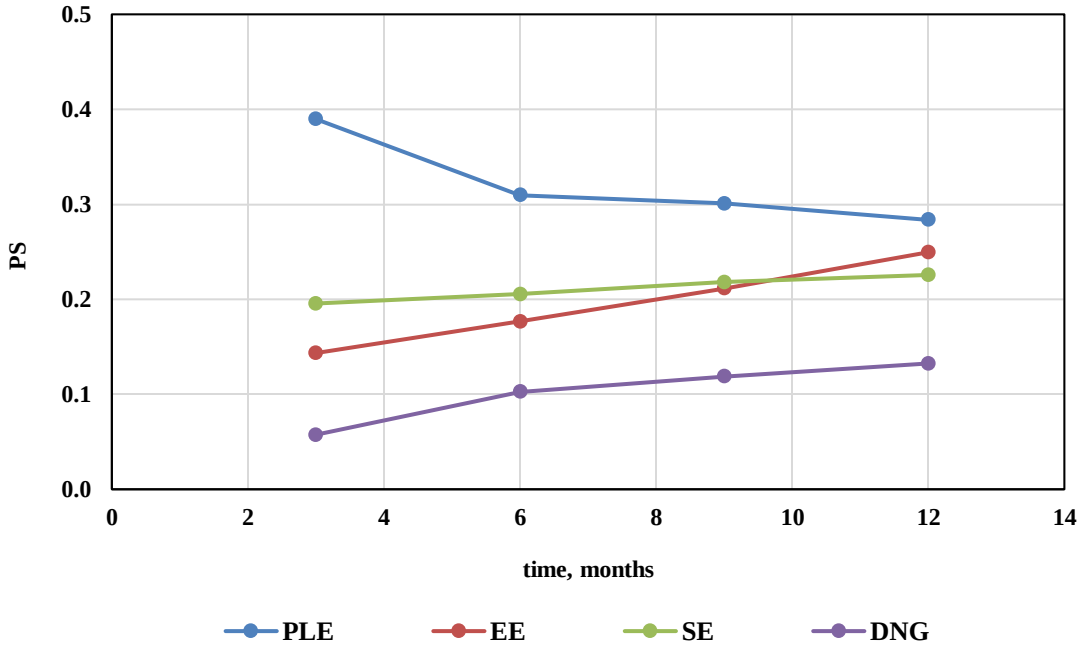

Figure 13: Pleasant surprise for all DCA's in the Permian

Meanwhile, SE tends to have an inherent PS value ballparked at 0.2 irrespective of the time interval and the PS increases linearly for EE with values ranging between (0.14 – 0.25). These concepts – ED and PS are crucial in the decision-making techniques involved in the evaluation of reservoirs because it is inherent for a decision maker to choose the project alternative with the maximum outcome whilst disregarding the alternative with the minimum outcome thus creating a false sense of confidence which leads to an eventual disappointment and loss in revenue if the maximum outcome is chosen.

In totality, from the 12 month interval which is proposed to be the minimum amount of data required for a good forecast (Gupta et al., 2018) it is seen that the slope of the DCA plot becomes somewhat similar and if projected into the future there will exist a point where the ED and PS values for all DCA's taper off and uncertainty due to decision errors will be fully eliminated

Section 3

Considering that different DCA techniques somewhat fit the production history satisfactorily as shown in the figures from section 1 and 2. The obvious differences in these techniques arises from how the EUR was predicted and its governing assumption, this discrepancy begs the question which DCA is truly better and suitable for use at a given time since there is a noticeable inconsistency in the prediction of EUR.

From the previous sections, it is found that the best-fit DCA method for production forecasting changes over time from one model to the other alongside a decrease in the prediction error. The EUR was predicted using historic data via power law exponential, extended exponential, stretched exponential, and Duong decline curve models for all wells.

Comparing **Figures 14, 15, 16,** and **17** it is seen that, the quality of wells defined by EUR prediction using PLE, SE, and DNG decline curve models changes over time. So, very poor wells can become good/excellent wells as more production data is available for forecast. It should be noted that the opposite of the above phenomena (i.e. the number of good wells decreases over time) occurs using EE method which is characterized by a hump during the 6- and 9-months duration in **Figure 10**.

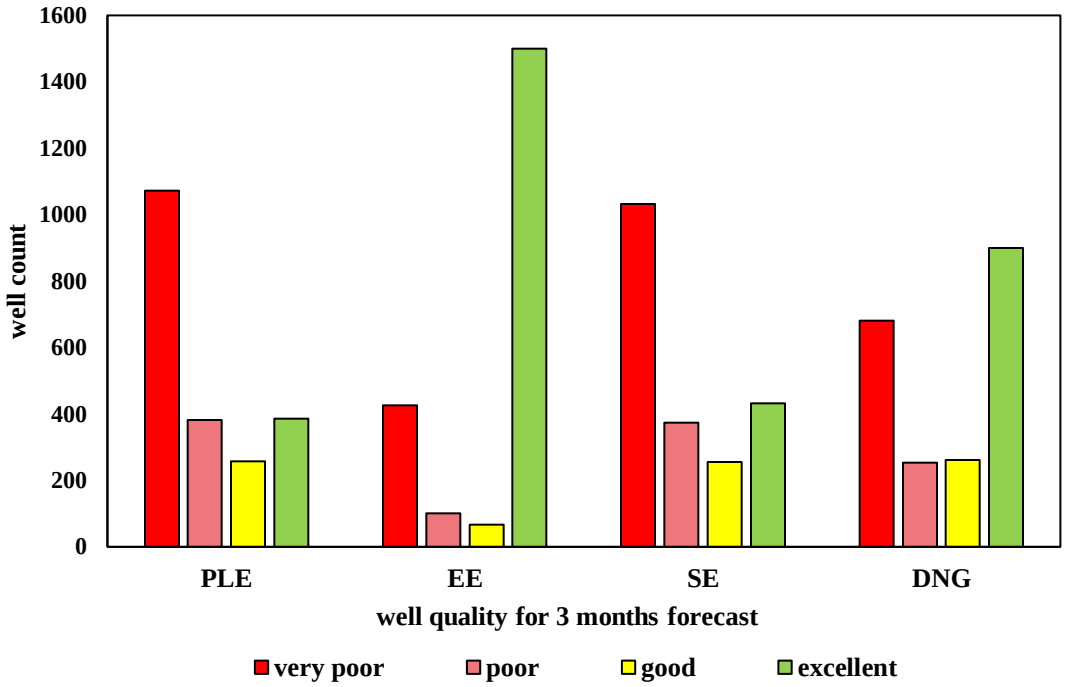

Figure 14:Well quality and performance for 3 months' time interval in the Permian

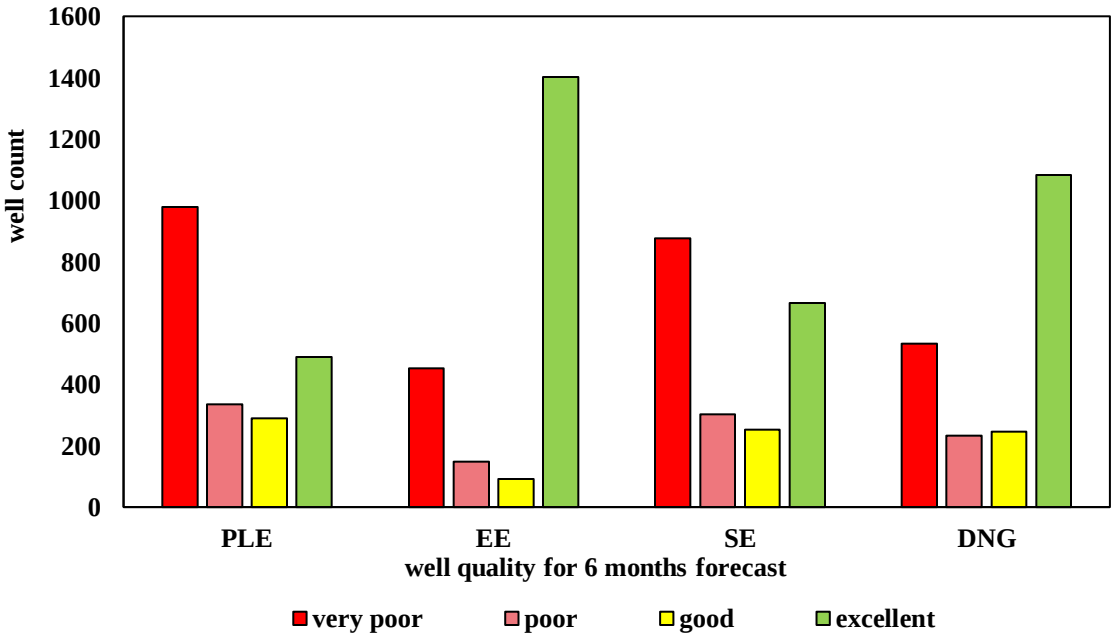

Figure 15: Well quality and performance for 6 months' time interval in the Permian

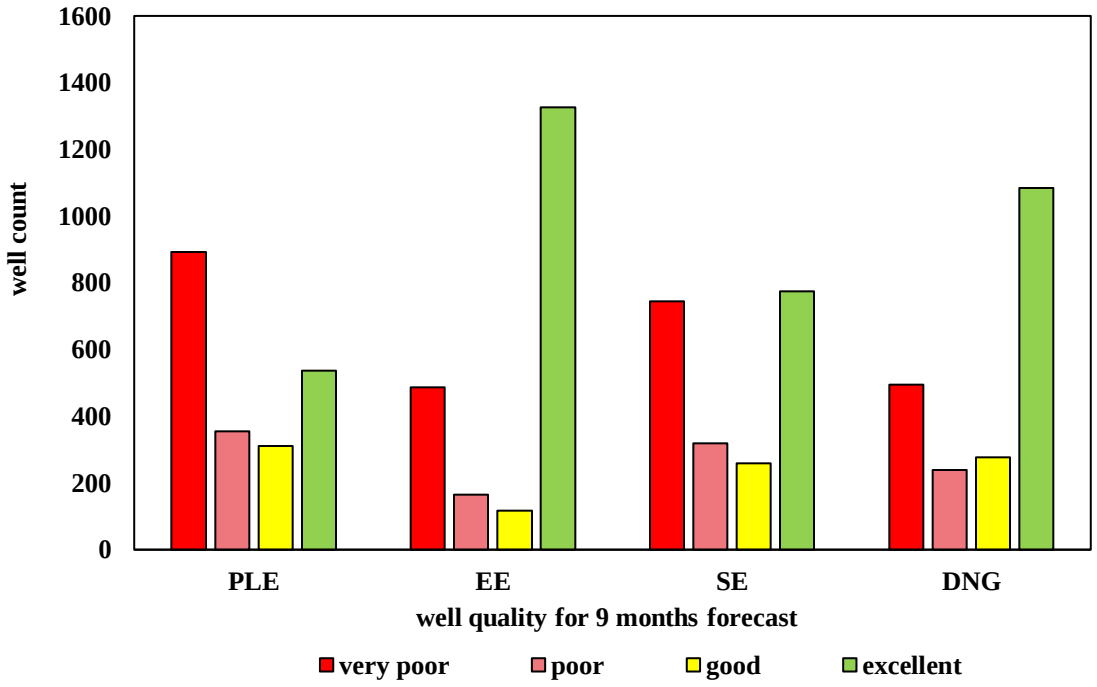

Figure 16: Well quality and performance for 9 months' time interval in the Permian

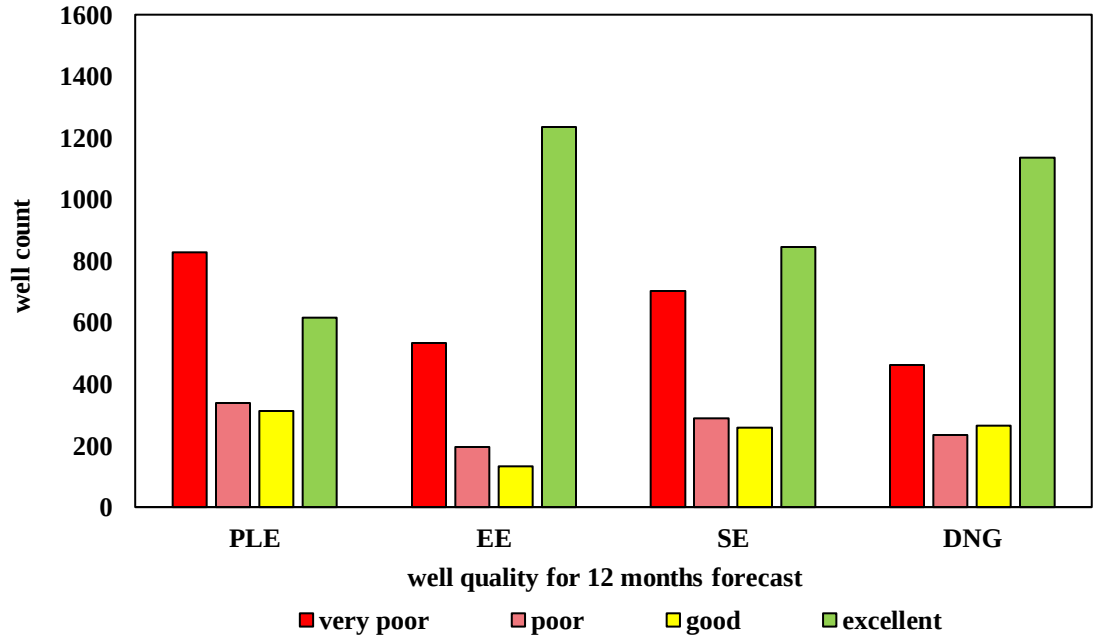

Figure 17: Well quality and performance for 12 months' time interval in the Permian

Using a methodology adaptation from Lolon (2016) to further corroborate these findings, **Figure 18** shows the visualization of four clusters gotten from the 12 month plot of EUR/lateral length versus proppant per lateral length for PLE which demonstrates the profitability zones in the Permian are implicitly based on drilling and completion practices whilst paving the path to determine the "sweet spot" via optimization of fracture spacing and horizontal length in the wells.

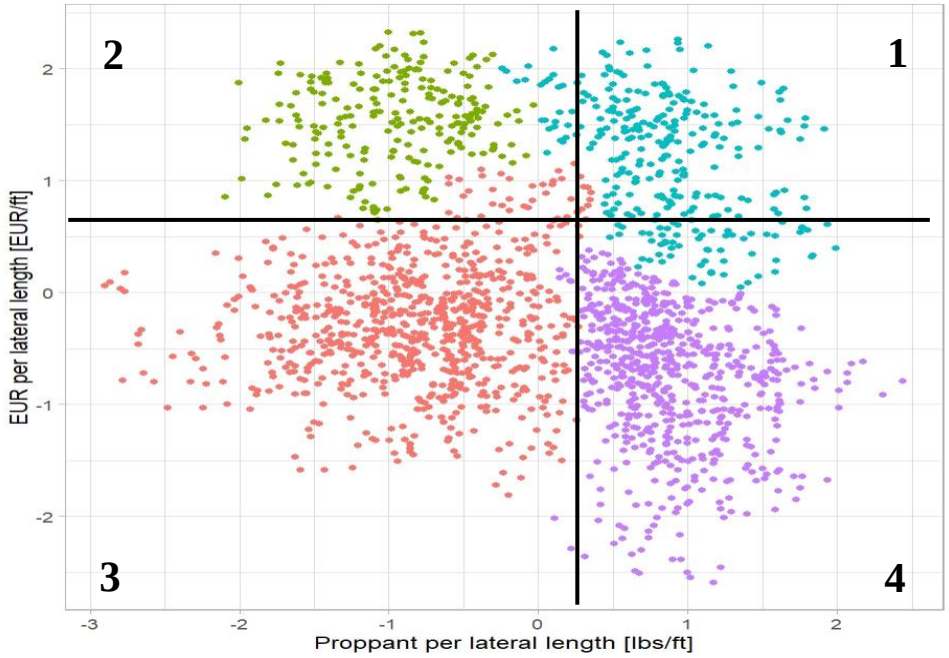

Figure 18: Representation of wells using clusters to show the performance of each well using PLE in the Permian basin

This can be seen in the two zones; zone 2 and zone 4, zone 2 characterizes good wells that have high EUR/ft values using a small amount of proppant per ft during the completion practices of these wells i.e. the sweet spot whilst zone 4 characterizes wells that behave poorly having a low EUR/ft value and very high amount of proppant per lateral length and the remaining wells fall between zones 2 and 4 i.e. zones 1 and 3.

Although the correlation between the variables in these clusters have a weak correlation as shown in **Figure 18**, this might be an indication of the coupling effect that exists between lateral length, proppant amount used, amount of fluid injected, formation type, location and the petrophysical properties of the formation as discussed in King (2010) & Yuan et al. (2017). The same analysis was carried out for all DCA methods considered and similar results were obtained although, the number of wells in these clusters i.e. zone 2 and 4 changes significantly with each method. (*See Appendix D for clustering results using other DCA models*)

Shale gas system

The results from the shale gas system consisting of over 11000 wells in both Barnett and Haynesville plays are elucidated in threefold as shown in each subsection.

Section 1

In **Figures 19** & **20**, it is seen that as the time interval increases, the standard deviation (SD) of the bias involved in each prediction decreases in both gas plays. When extrapolated, it plateaus after the 30 and 16-month mark in Barnett and Haynesville respectively for all DCA models. Also, the magnitude of the SD of bias in Haynesville is higher than Barnett indicating that there is more variability in its prediction.

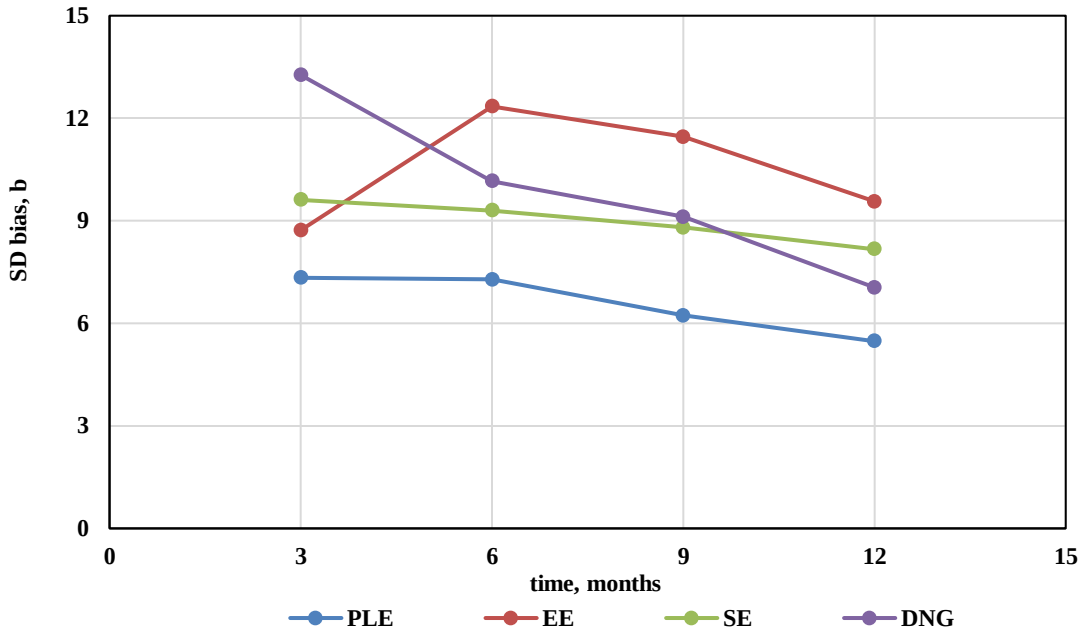

Figure 19: Standard deviation of bias involved in all DCA's in the Barnett play

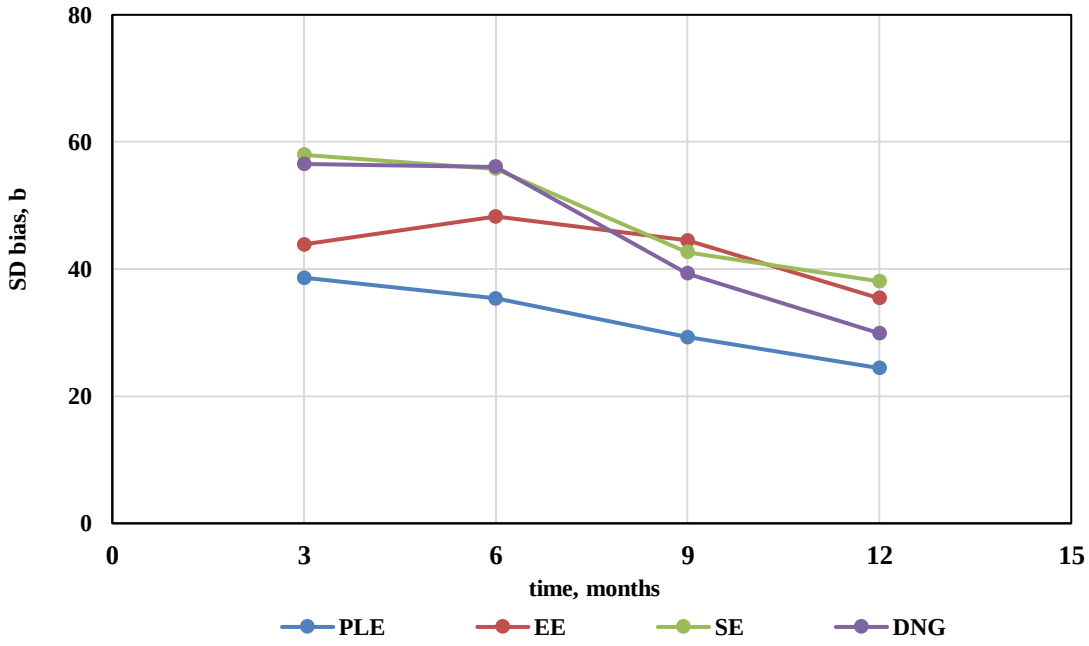

Figure 20: Standard deviation of bias involved in all DCA's in the Haynesville play

To fully understand the effect of the bias in the production forecast using different decline curve analysis models in gas systems, **Figures 21, 22** & **23, 24** shows the standard deviation and mean error obtained during prediction in Barnett and Haynesville. Coupling **Figures 21, 22** & **23, 24** with **Figures 19** and **20** respectively for these plays, shows that during all time intervals considered PLE is the best-fit although it overestimates the cumulative production while SE, DNG significantly overpredicts in Barnett and Haynesville.

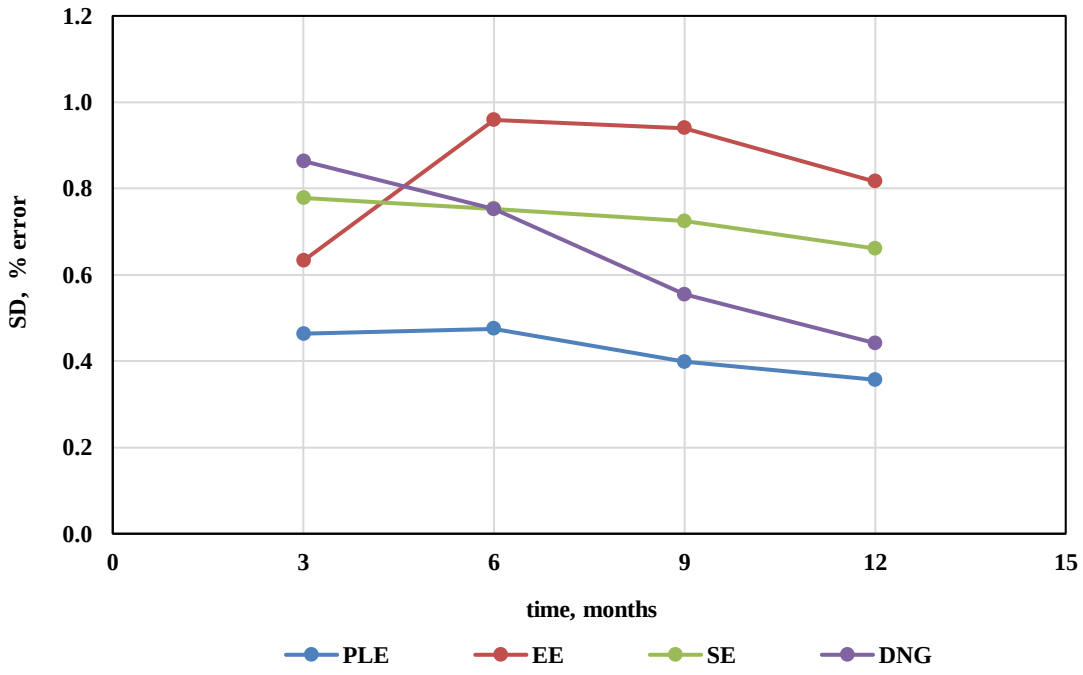

Figure 21: Standard deviation of average errors for all DCA models in the Barnett play

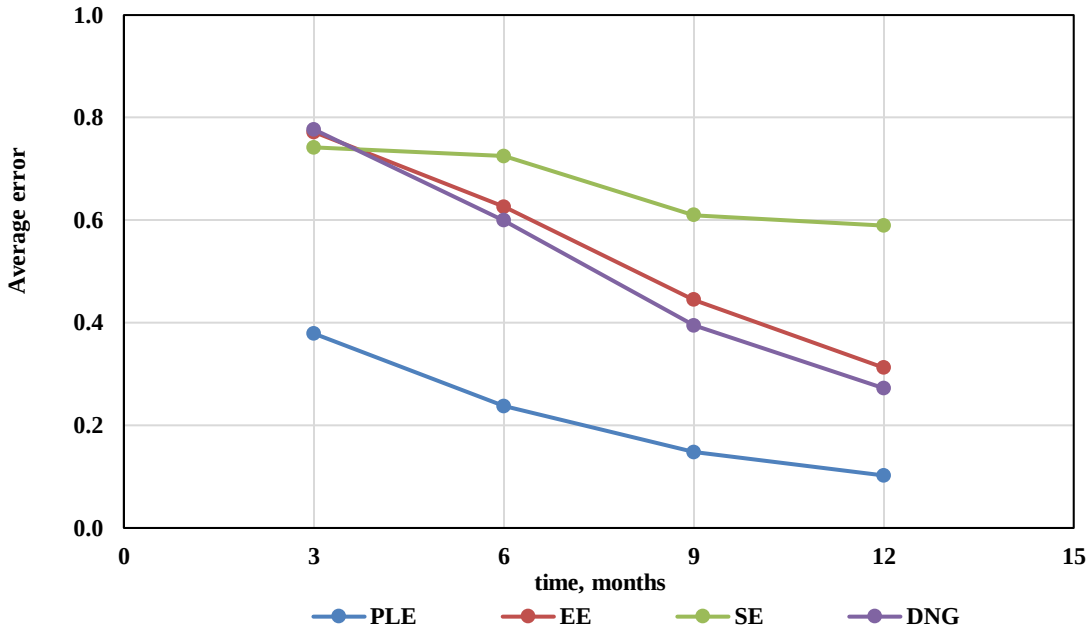

Figure 22: Average errors for all DCA models in the Barnett play

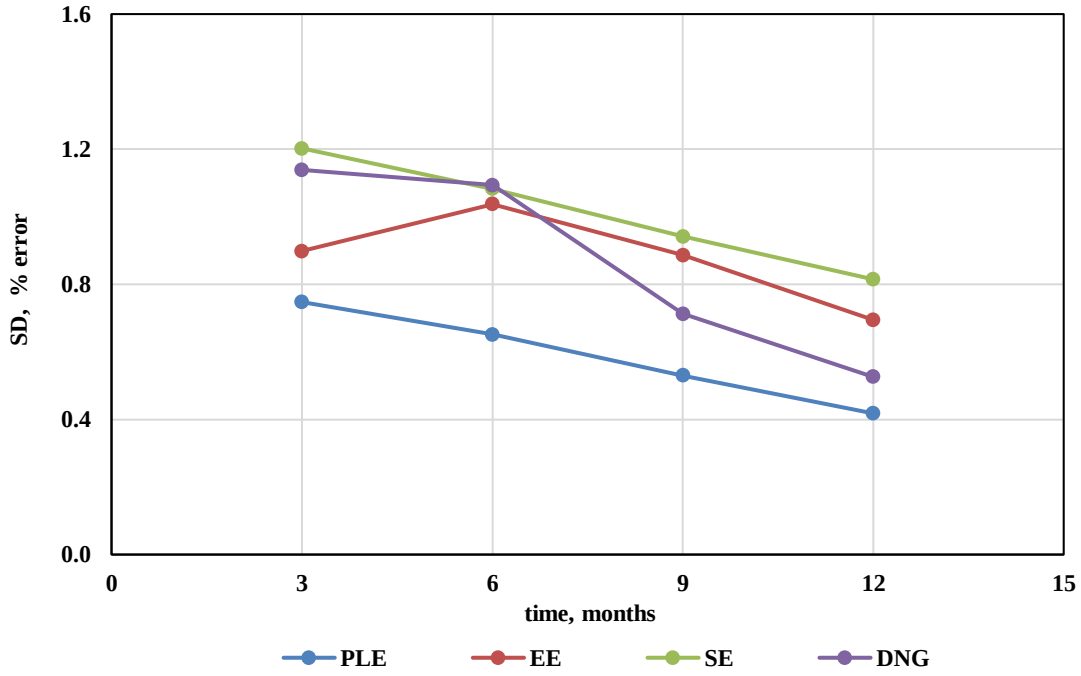

Figure 23: Standard deviation of average errors for all DCA models in Haynesville play

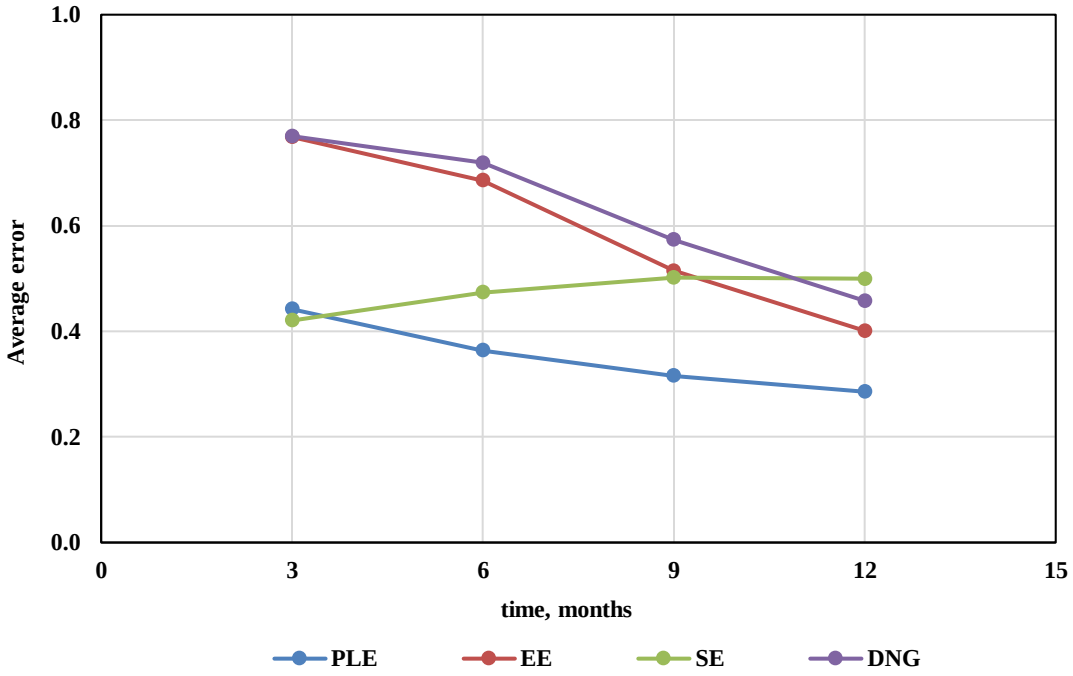

Figure 24: Average errors for all DCA models in the Haynesville play

Section 2

The results from the previous section show that as the standard deviation of the errors decreases, the expected disappointment associated with each DCA model in **Figures 25** and **26** decreases, this follows the conclusion from Begg & Bratvold (2008). Also, in **Figures 25** & **26** PLE has the lowest ED values (0.52 – 0.59) and (0.41 – 0.60) then, the highest ED values SE (0.59 – 0.84) and DNG (0.68 – 0.77) in the Barnett & Haynesville plays respectively. Thus, making the power law exponential and stretched exponential decline curve methods the lower and upper bands of the predictions in the Barnett whilst the PLE and Duong's methods are the lower and upper boundaries of the predictions in Haynesville. This trend was also depicted clearly in the average errors reported in the last section.

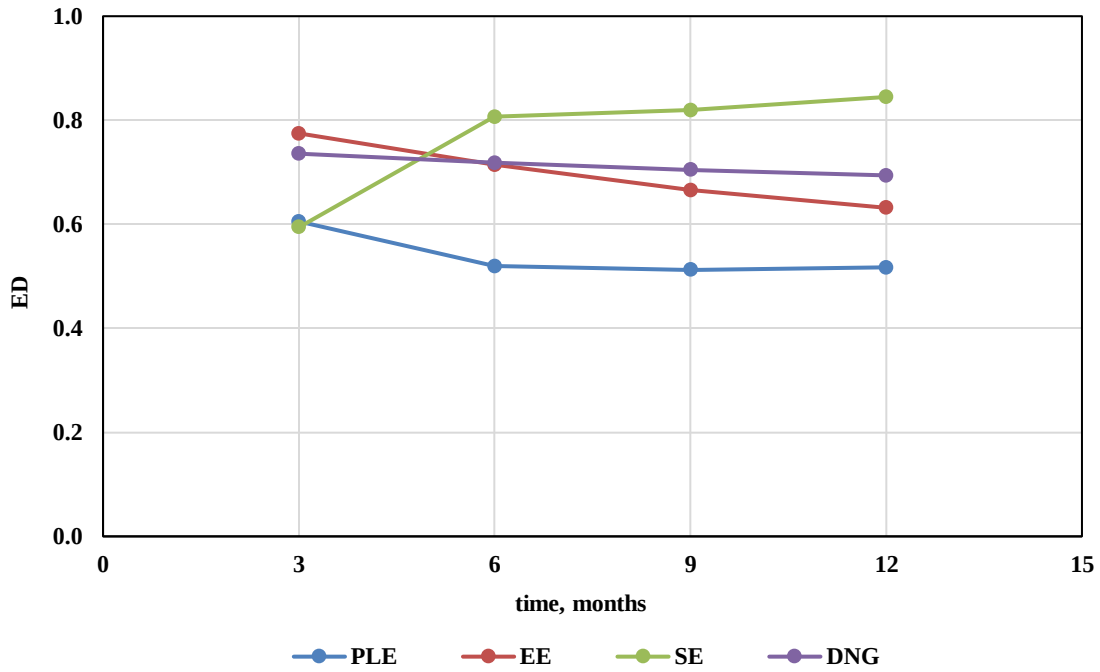

Figure 25: Expected disappointment for all DCA models in the Barnett play

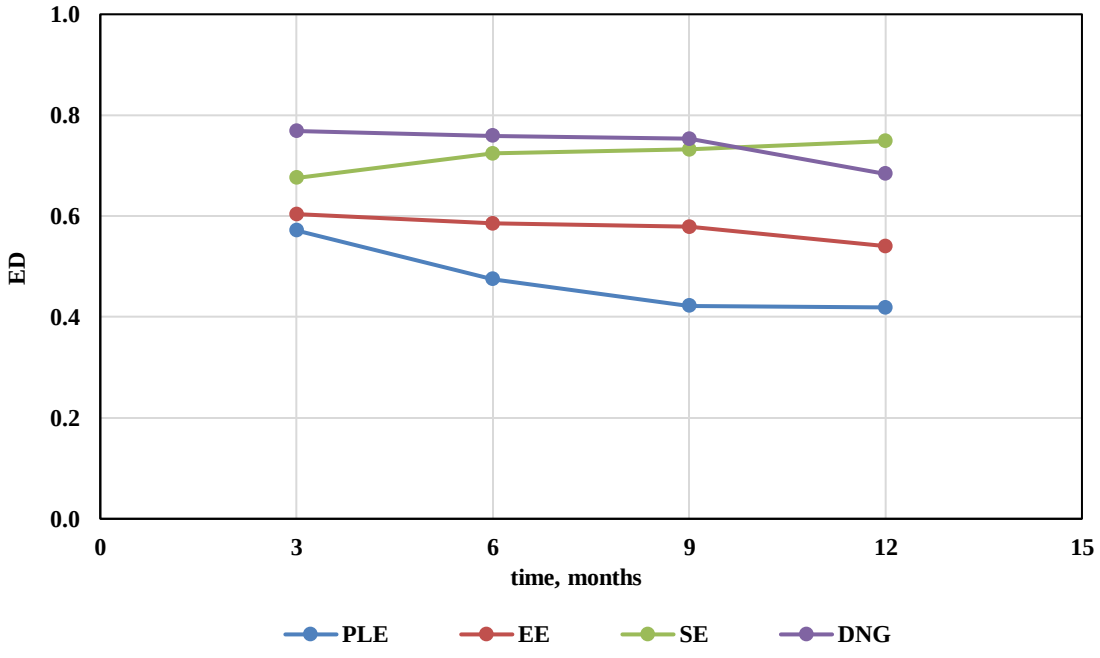

Figure 26: Expected disappointment for all DCA models in the Haynesville play

The concept of expected disappointment is crucial in decision-making techniques involved in the evaluation of reservoirs because it is inherent for a decision maker to choose the project alternative with the maximum outcome whilst disregarding the alternative with the minimum outcome thus creating a false sense of confidence which leads to an eventual disappointment and loss in revenue if the maximum outcome is chosen.

Section 3

From the analysis in section 1 and 2, it is seen that the DCA techniques fit the gas production history somewhat satisfactorily. The obvious differences and inaccuracies in these techniques arise mainly from its governing assumption and reservoir attributes, making it imperative to discuss its petrophysical properties. To have further insight, the decline rates (D_i) of the 7000 Barnett wells and 4361 Haynesville wells was compared.

From **Figure 27**, it is observed that wells in Haynesville have an average decline rate of $0.22\ month^{-1}$ which is higher relative to Barnett's average of $0.16\ month^{-1}$.

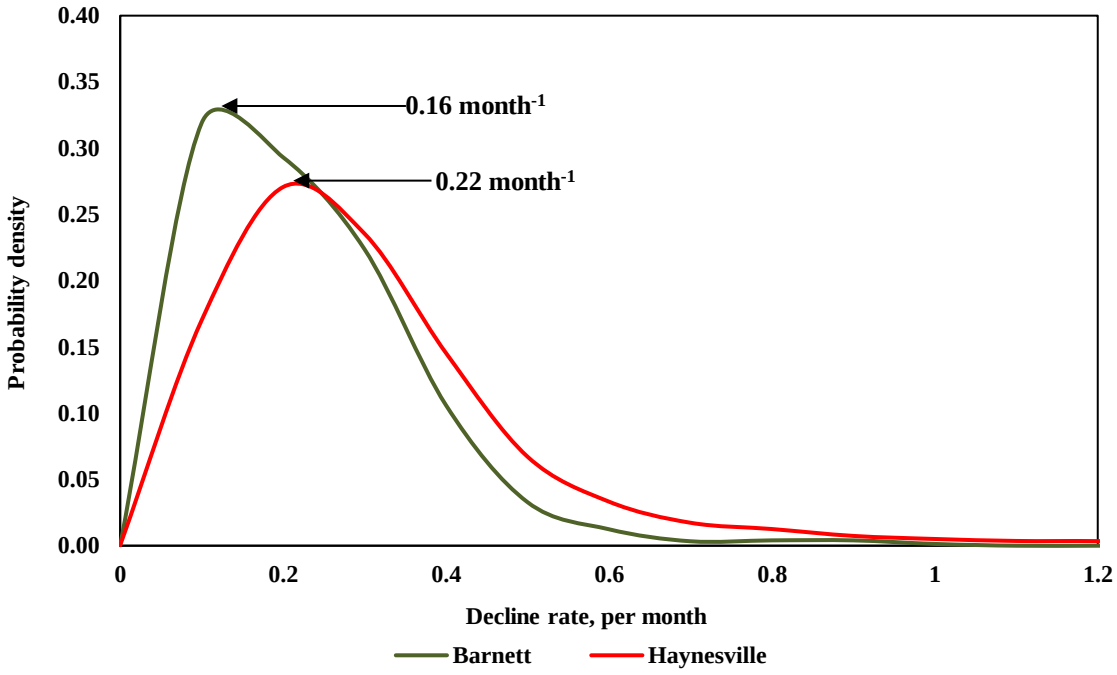

Figure 27: The decline rates in Barnett and Haynesville plays

This can be ascribed to the decrease in reservoir pressure as a result of depletion/production leading to an increase in effective stress. This would result in significant permeability reduction which is a function of its petrophysical properties such as pore type & shape. Usually observed orders of magnitude reduction in matrix permeability can be related to pore connectivity loss and microfractures inside the matrix (Davudov & Moghanloo 2018). It should be noted that depending on the formation under consideration, one of these effects might be dominant or weak. Additionally, hydraulic fracture closure would also have a significant impact on production decline.

Knowing that the pores type in Haynesville are slit-shaped, dispersed and inorganic whilst that of Barnett is cylindrical, close to each other and organic as shown in **Figures 28** & **29** respectively.

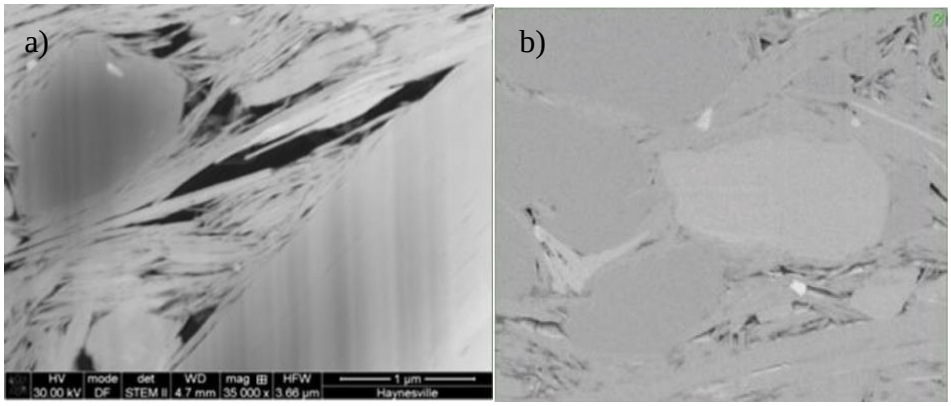

Figure 28: a) Energy dispersive spectroscopy map b) Backscattered electrons of the Haynesville shale
(Curtis et al., 2010)

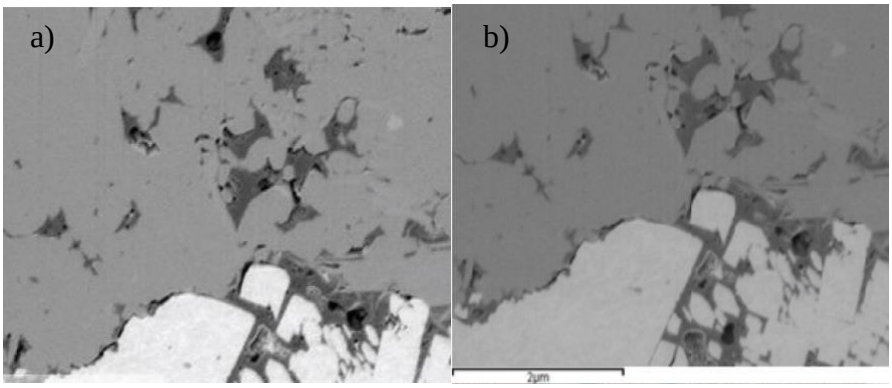

Figure 29: a) Energy dispersive spectroscopy map b) Backscattered electrons of the Barnett shale
(Curtis et al., 2010)

Davudov & Moghanloo (2017) have found that pores in Haynesville have a higher aspect ratio in comparison to pores in Barnett. Therefore, the slit pores observed in Haynesville is susceptible to rapid pore collapse under effective stress that occurs during depletion/production. Consequently, reservoir deliverability is impaired as seen in the production forecasts made.

This discussion also explains why the bias & rate of decline is much higher and the connectivity of the matrix is declining much faster in Haynesville when compared to Barnett. This

might have an impact on future EOR applications since there are two major factors affecting the EOR in shales: good fracture network which will increase the surface area, and connectivity so CO_2 can pass through. Therefore, it can be inferred that since the decline rate during production is somewhat related to pore shape and connectivity. It could be a good qualitative indicator of wells in which EOR might be successful although this needs to be further investigated.

Chapter 5: Conclusions

Although there are various types of DCA models for unconventional reservoirs, this work focuses on only four; power law exponential, stretched exponential, Duong, and extended exponential methods. Existing uncertainty in the production forecast and prediction of EUR in unconventional reservoirs – shales can result in sub-par field development and appraisal plan which is a key driver in the economics involved during production. In this study, 11000+ gas wells from Barnett & Haynesville plays and 2000+ oil wells from the Permian are classified based on its performance over 3, 6, 9 and 12 months. The results obtained shows that as the time interval increases the estimated error decreases for all DCA methods.

From **Table 1**, it is shown that the best-fit DCA model for time intervals – 3, 6, 9, and 12 months is relative and choosing one DCA model for forecast during all time intervals favors a set of parameters over the other which unknowingly introduces a bias in the decision-making process. However, it appears that in the gas plays considered, power law exponential serves as the lower boundary of the forecast whilst the upper boundary – stretched exponential (SE) and Duong (DNG) method always significantly overpredicts the cumulative production as seen in Barnett and Haynesville respectively.

Table 1: Summary of results showing DCA boundaries an operator should use in the Permian, Barnett, and Haynesville plays.

Permian		Barnett		Haynesville	
Lower bound DCA	Upper bound DCA	Lower bound DCA	Upper bound DCA	Lower bound DCA	Upper bound DCA
PLE	DNG	PLE	SE	PLE	DNG

Meanwhile the oil system; Permian follows the same boundary trend as the Haynesville play. Also, the quality of wells defined by cumulative oil production and the amount of proppant used per lateral length changes over time so, very poor wells may become good or excellent wells in the later stage of production when more data become available as input for the production forecast and vice versa.

In conclusion, my research attempts to add some objectivity to DCA which is naturally a subjective technique. It is hoped that the outcome of this work will prompt reservoir evaluators to reliably quantify uncertainty by reducing or eliminating expected disappointment and bias associated with decline curve analysis. Also, it will enable financial institutions to factor uncertainty in loan proposals from oil and gas companies for well development by affixing a number to uncertainty in shale formations. This is necessary in order to avoid overpredicting and underpredicting the production forecast since it implicitly affects profitability.

Chapter 6: Recommendations for Future Work

Knowing that DCA models are not entirely enough to forecast production in unconventional reservoirs due to the variation in characteristics, operational conditions, and time rate equation of each shale play. It is crucial to understand the behavior of each model and apply it properly since one DCA method does not fit all. Hence the following recommendations can be implemented:

1. Creating forecasting models that are not only empirical but can be significantly improved by applying relevant data mining and data science techniques incorporating bias and uncertainty in its workings to increase its capability. These techniques would help find and decouple existing patterns and trends that are implicitly involved with the production data: such as reservoir parameters etc and help mitigate expected disappointment.
2. Investigating the relationship between decline rates during oil and gas production and connectivity of pores as a possible qualitative indicator of potential EOR success in wells.

References

Akbarnejad-Nesheli, B., Valko, P., & Lee, J. W. (2012, January). Relating fracture network characteristics to shale gas reserve estimation. *In SPE Americas Unconventional Resources Conference*. Society of Petroleum Engineers.

Arps, J. J. (1945). Analysis of decline curves. *Transactions of the AIME, 160*(01), 228-247.

Begg, S. H., & Bratvold, R. B. (2008, January). Systematic Prediction Errors in Oil and Gas Project and Portfolio Selection. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Bell, D. E. (1985). Disappointment in decision making under uncertainty. *Operations research*, *33*(1), 1-27.

Capen, E. C. (1976). The Difficulty of Assessing Uncertainty (includes associated papers 6422 and 6423 and 6424 and 6425). *Journal of Petroleum Technology*, *28*(08), 843-850.

Curtis, M. E., Ambrose, R. J., & Sondergeld, C. H. (2010, January). Structural characterization of gas shales on the micro-and nano-scales. In *Canadian unconventional resources and international petroleum conference*. Society of Petroleum Engineers.

Davudov, D., & Moghanloo, R. G. (2017). Scale-Dependent Pore and Hydraulic Connectivity of Shale Matrix. *Energy & Fuels*, *32*(1), 99-106.

Davudov, D., & Moghanloo, R. G. (2018). Impact of pore compressibility and connectivity loss on shale permeability. *International Journal of Coal Geology*, *187*, 98-113.

Duong, A. N. (2011). Rate-decline analysis for fracture-dominated shale reservoirs. *SPE Reservoir Evaluation & Engineering*, *14*(03), 377-387.

Gupta, I., Rai, C., Sondergeld, C., & Devegowda, D. (2018, September). Variable Exponential Decline-Modified Arps to Characterize Unconventional Shale Production Performance. *In Unconventional Resources Technology Conference, Houston, Texas, 23-25 July 2018* (pp. 3552-3568). Society of Exploration Geophysicists, American Association of Petroleum Geologists, Society of Petroleum Engineers.

Ilk, D., Anderson, D. M., Stotts, G. W., Mattar, L., & Blasingame, T. (2010). Production data analysis--Challenges, pitfalls, diagnostics. *SPE Reservoir Evaluation & Engineering*, *13*(03), 538-552.

Ilk, D., Rushing, J. A., & Blasingame, T. A. (2009, January). Decline-Curve Analysis for HP/HT Gas Wells: Theory and Applications. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

King, G. E. (2010, January). Thirty years of gas shale fracturing: What have we learned?. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Lee, W. J., & Sidle, R. (2010). Gas-reserves estimation in resource plays. *SPE Economics & Management, 2*(02), 86-91.

Lewis, J. O., & Beal, C. H. (1918). Some New Methods for Estimating the Future Production of Oil Wells. *Transactions of the AIME, 59*(01), 492-525.

Lolon, E., Hamidieh, K., Weijers, L., Mayerhofer, M., Melcher, H., & Oduba, O. (2016, February). Evaluating the Relationship Between Well Parameters and Production Using Multivariate Statistical Models: A Middle Bakken and Three Forks Case History. In *SPE Hydraulic Fracturing Technology Conference*. Society of Petroleum Engineers.

Mabadeje, A.O., & Ghanbarnezhad Moghanloo, R. (2019a, April). Uncertainty Analysis of Production Forecast in Permian Basin. In *SPE Oklahoma City Oil & Gas Symposium*. Society of Petroleum Engineers.

Mabadeje, A.O., & Ghanbarnezhad Moghanloo, R. (2019b, April). Uncertainty Analysis in Production Forecasting-Part II: A case study on the Barnett and Haynesville Shale Plays. In *SPE Western Regional Meeting*. Society of Petroleum Engineers.

McVay, D. A., & Dossary, M. N. (2014). The value of assessing uncertainty. *SPE Economics & Management, 6*(02), 100-110.

Mohaghegh, S. 2017. *Shale Analytics Data-Driven Analytics in Unconventional Resources*. Springer International Publishing. ISBN 978-3-319-48753-3.

Rushing, J. A., Perego, A. D., Sullivan, R., & Blasingame, T. A. (2007, January). Estimating reserves in tight gas sands at HP/HT reservoir conditions: use and misuse of an Arps decline curve methodology. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Valkó, P. P. (2009, January). Assigning value to stimulation in the Barnett Shale: a simultaneous analysis of 7000 plus production hystories and well completion records. In *SPE hydraulic fracturing technology conference*. Society of Petroleum Engineers.

Yuan, G., Dwivedi, P., Kwok, C. K., & Malpani, R. (2017, June). The Impact of Increase in Lateral Length on Production Performance of Horizontal Shale Wells. In *SPE Europec featured at 79th EAGE Conference and Exhibition*. Society of Petroleum Engineers.

Zhang, H., Rietz, D., Cagle, A., Cocco, M., & Lee, J. (2016). Extended exponential decline curve analysis. *Journal of Natural Gas Science and Engineering*, *36*, 402-413.

Zuo, L., Yu, W., & Wu, K. (2016). A fractional decline curve analysis model for shale gas reservoirs. *International Journal of Coal Geology*, 163, 140-148.

Appendix A: R Code Generated for Production Forecast

The R code written using the procedure explained to carry out decline curve analysis using the PLE, EE, SE, and DNG methods as explained in chapter 3 are as follows:

```
#' title: "Production Forecasting Using Decline Curve Analysis in Shales"
#' author: "Ademide Mabadeje"
#' date created: September 7th, 2018
#' last date edited: January 14th, 2019

library("plyr")
library("ggplot2")
library("dplyr")
library("qpcR")
# library("caret")

# set working directory to desktop
setwd("~/Desktop/R script")

# call functions
source("functions.R")

# choose data file and specify snapshot for modelling
fileName = "Reservoirname.CSV"
```

```
threshold = 5   #Economic limit for oil/gas production during forecast
timeStart = 1   #Start time for data fitting part
timeStop = 12   #Last data point used for data fitting part

# read in data file
play=read.csv(fileName)

# check for missingness
totalMissing = function(x) sum(is.na(x))
apply(play,2,totalMissing)

missingInd = which(is.na(play$Monthly.Gas) == TRUE)

if (length(missingInd) > 0)
{
  play = play[-missingInd,]
}

# create output filename
pleSol = paste("PLE Full solution for times(t)",timeStart,"to", timeStop, fileName, sep = " ")
agPleSol = paste("PLE aggregated for times(t)",timeStart,"to", timeStop, fileName, sep = " ")
eeSol = paste("EE Full solution for times(t)",timeStart,"to", timeStop, fileName, sep = " ")
```

```
agEeSol = paste("EE aggregated for times(t)",timeStart,"to", timeStop, fileName, sep = " ")
seSol = paste("SE Full solution for times(t)",timeStart,"to", timeStop, fileName, sep = " ")
agSeSol = paste("SE aggregated for times(t)",timeStart,"to", timeStop, fileName, sep = " ")
dngSol = paste("Doung Full solution for times(t)",timeStart,"to", timeStop, fileName, sep = " ")
agDngSol = paste("Doung aggregated for times(t)",timeStart,"to", timeStop, fileName, sep = " ")



# Prepare data
# Note: Input data should be sorted by Entity.ID
# idsorted = sort(unique(play$Entity.ID))
ids = as.factor(unique(play$Entity.ID)) # 5362 wells
play=play[,c("Entity.ID","Monthly.Production.Date","Monthly.Gas")]
# convert entity id to factor
play$Entity.ID = as.factor(play$Entity.ID)

# check for missingness
totalMissing = function (x) sum(x == 0)
apply(play,2, totalMissing)

# remove monthly gas values that are <= 5
zeroMonthlyGas = which(play$Monthly.Gas <= threshold)
play = play[-zeroMonthlyGas,]
```

```
# remove initial increasing points
playClean = mainClean(play, ids, minDays = 3, maxDays = 1000)

# update ids
ids = as.factor(unique(playClean$Entity.ID)) # 5362 wells


## MODELLING
# nlsPLE
playNls = nlsPowerLaw(playClean, ids, timeStart, timeStop)

# nlsEE
playEE = nlsEE(playClean, ids, timeStart, timeStop)

# nlsSE
playSE = nlsSE(playClean, ids, timeStart, timeStop)

# nlsDoung
playDoung = nlsDoung(playClean, ids, timeStart, timeStop)

idNls = unique(playNls$Entity.ID)
playNls = rmsePredict(playNls,idNls, timeStop, "PLE")
```

```
idNlsEE = unique(playEE$Entity.ID)
playEE = rmsePredict(playEE,idNlsEE, timeStop, "EE")

idNlsSE = unique(playSE$Entity.ID)
playSE = rmsePredict(playSE,idNlsSE, timeStop, "SE")

idNlsDoung =  unique(playDoung$Entity.ID)
playDoung = rmsePredict(playDoung,idNlsDoung, timeStop, "Doung")
## OUTPUT TO EXCEL

# nlsPLE
# write full solution to excel
write.csv(playNls, pleSol)

# write aggregate df
agplayNls =  aggregate(. ~ Entity.ID, playNls, mean)
agplayNls = agplayNls[, -which(colnames(agplayNls) == "Monthly.Production.Date")]
agplayNls = agplayNls[, -which(colnames(agplayNls) == "newDate")]

# write aggregate df to excel
write.csv(agplayNls, agPleSol)
```

```
## nlsEE
# write full data set to excel
write.csv(playEE, eeSol)

# write aggregate df to excel
agplayEE =  aggregate(. ~ Entity.ID, playEE, mean)
agplayEE = agplayEE[, -which(colnames(agplayEE) == "Monthly.Production.Date")]
agplayEE = agplayEE[, -which(colnames(agplayEE) == "newDate")]

# write aggregate df to excel
write.csv(agplayEE, agEeSol)

## nlsSE
# write full data set to excel
write.csv(playSE, seSol)

# write aggregate df to excel
agplaySE =  aggregate(. ~ Entity.ID, playSE, mean)
agplaySE = agplaySE[, -which(colnames(agplaySE) == "Monthly.Production.Date")]
agplaySE = agplaySE[, -which(colnames(agplaySE) == "newDate")]

write.csv(agplaySE, agSeSol)
```

```
## nlsDuong

# write full data set to excel

write.csv(playDoung, dngSol)


# write aggregate df to excel

agplayDoung =  aggregate(. ~ Entity.ID, playDoung, mean)

agplayDoung = agplayDoung[, -which(colnames(agplayDoung) == "Monthly.Production.Date")]

agplayDoung = agplayDoung[, -which(colnames(agplayDoung) == "newDate")]


write.csv(agplayDoung, agDngSol)
```

Appendix B: R Code Generated for Cumulative Production & EUR

To estimate the cumulative production up to a certain time duration and estimate the EUR for each well in the play of interest, the full dataset output for each DCA model from appendix A was inputted using following codes:

```
#' title: "Cummulative Production and EUR determination"
#' author: "Ademide Mabadeje"
#' date created: December 3rd, 2018
#' last date edited: January 14th, 2019

# inputs:
# output: return results
source("functions.R")

inputData = read.csv("EE Full solution for times(t) 1 to 12 use Producing Entity Monthly Production copy.CSV", header = T)
tInf = 5000 #the time frame in months required to estimate EUR

npStart = 1 #the start date in months for cumulative production
npStop = 3 #the end date in months for cumulative production

#' The results and write section should be changed simultaneously for the DCA under use
#resultsPle = npEurPle(inputData,tInf, npStart, npStop)
```

```
#resultsSe = npEurSe(inputData,tInf, npStart, npStop)

resultsEe = npEurEe(inputData,tInf, npStart, npStop)

#resultsDoung = npEurDoung(inputData,tInf, npStart, npStop)


# plot

#with(resultsDoung,plot(eurDoung,NpDoung))# xlim = c(0,2e7))) # change 2e7 to sweet you


write.csv(resultsEe, "Ee 1 to 3.CSV")

#write.csv(resultsPle, "Ple 1 to 3.CSV")

#write.csv(resultsSe, "Se 1 to 3.CSV")

#write.csv(resultsDoung,"Duong 1 to 3.CSV")


#windows()

#plot(cleanData$Monthly.Gas, cleanData$qDng)
```

Appendix C: R Code Functions Created for Call-backs

```
## cleaner algorithm

# input: original data, unique ids, minDay, maxDay

# output: cleaned data

# initialize deleteRow


#  declare variable deleteRows to store index of all rows to be deleted

## function cleanData (completed)   # retuns index of points to be deleted

# a. For each id

# 1. Find corresponding row indices

# 2. load month gas values for this row index


# 3. ## function findInitialIncrease (completed)   # returns index of last increase point

# if monthly gas in step 2 is greater than

# Run dca (find initial increasing points and return index)

# else return index of all monthly data


# 4. Delete all rows corresponding to index in 3

# 5. Repeat step 1 to 4 for each id


# return cleanData


# function prepareNewDate (completed)
```

```
# 6. update unique ids for cleaned data

# 7. create new variable => newDate

# b. for each id

# 1. find corresponding row indices

# 2. if number is indices in b.1 is < minDay or >maxDay

# save index

# else set newDate => row index normalized to start at 1

# 7. delete all saved indices in b.2

# return data


# mainClean

# 1. cleanData :: findInitialIncrease :: updateRowsToDelete

#     return index

# 2. delete rows returned by cleanData

# 3. prepare new Date()

#     returns  modified data


# function findInitialIncrease

findInitialIncrease = function (monthlyGas) # returns last index  of increase

{

  # function finds initial increase trend and returns index of these points

  # Note:   function returns 0 for non increasing data. remove this index from

  #         output before use to adjust any matrix
```

```
  # status: tested with 3 inputs. Performs as expected


  if (length(monthlyGas) == 0)
  {
    print("Wrong Input to findInitialIncrease. length (input) is zero. Function will return null ")
    return(NULL)
  }
  else if (length(monthlyGas) <= 2)
  {
    print("Wrong Input to findInitialIncrease. length (input) is less than 2. Function will return
null ")
    return(NULL)
  }
  else
  {
    i = 1
    check1stRun = 0


    print(length(monthlyGas))


    while(monthlyGas[i] < monthlyGas[i + 1] && length(monthlyGas) >= (i + 1))
    {
      i = i + 1
```

```
  check1stRun = check1stRun + 1
  print("here")
}
# at this point i is the beginning of well decrease


# run second time starting after first decrease
j = i + 1
check2ndRun = 0;
# print(paste("here. j is ", j))

while(monthlyGas[j] < monthlyGas[j + 1] && length(monthlyGas) >= (j + 1))
{
  # print(monthlyGas[j])
  # print(monthlyGas[j + 1])
  j = j + 1
  check2ndRun = check2ndRun + 1;
}

if (check2ndRun > 0) # means another increase was found after first decrease
{
  return(j - 1)
}
```

```
    else
    {
      if (check1stRun > 0)
        return(i - 1)
      else
        return(i)
    }

  }
}



# function CleanData
cleanData = function(data, idlist, end) # returns index of points to be deleted
{
  deleteRow = numeric()

  for (i in 1:length(idlist))
  {

    # a. For each id
    # 1. Find corresponding row indices
    rowInd = which(data$Entity.ID == idlist[i])
```

```
# 2. load month gas values for this row index
mg = data$Monthly.Gas[rowInd];
# 3. if length of monthly gas in step 2 is greater than 2
if (length(mg) > 2)
{
  # Run dca (find initial increasing points and return index)
  print(paste("ID:" ,idlist[i], "length of well is ", length(rowInd), sep = " "))
  endIndRowInd = findInitialIncrease(mg)
  # print(endIndRowInd)
  # print(">2")

  if (endIndRowInd >= 1 )
  {
    deleteMgInd = rowInd[1:endIndRowInd]
    # print(deleteMgInd)
    deleteRow = updateRowsToDelete(deleteRow, deleteMgInd)
  }

}
# else return index of all monthly data
else
{
  # print("<=2")
```

```
      deleteRow = updateRowsToDelete(deleteRow, rowInd)
    }



  }

  # print(deleteRow)
  return(deleteRow)
}



# function updateRowsToDelete
updateRowsToDelete = function(deleteRow, addRow)
{
  deleteRow = c(deleteRow, addRow)

  return(deleteRow)
}



# function prepareNewDate ()
prepareNewDate = function (data, idlist, minDays, maxDays) # adds new Date and number of
days field to data
```

```
{
 # create new variable newDate and numDays
 data$newDate = 0
 data$numDays = 0
 deleteRow = numeric()

 # for each id
 for (i in 1:length(idlist))
 {
  # find row number
  rowInd = which(data$Entity.ID == idlist[i])

  if (length(rowInd) == 0)
  {
   print(paste(idlist[i], "Well ID has no data point. Returning Null...", sep = " "))
   return (NULL)
  }
  # if current well production days is between minDays and maxDays
  else if (length(rowInd) >= minDays && length(rowInd) <= maxDays)
  {
   # set newDate to row number normalized to start at 1

   print(paste("Current well length is ",length(rowInd), sep = " "))
```

```
    data$newDate[rowInd] = rowInd - min (rowInd) + 1
    data$numDays[rowInd] = length(rowInd)
   }
   else
   {
    # wells don't meet user defined constraints. store index for deletion
    deleteRow = updateRowsToDelete(deleteRow,rowInd)
   }

 }

 if (length(deleteRow) > 0)
 {
  data = data[-deleteRow,]
 }

 return (data)
}

## function mainClean
mainClean = function(data, idlist, minDays, maxDays)
{
```

```
  # 1. cleanData :: findInitialIncrease :: updateRowsToDelete
  #     return index
  rowDelete =  cleanData(data, idlist)

  # 2. delete rows returned by cleanData
  data = data[-rowDelete,]

  # 3. update idlist
  idlist = unique(data$Entity.ID)

  # 3. prepare new Date()
  #     returns  modified data
  data = prepareNewDate(data, idlist, minDays, maxDays)

  return(data)
}


## modelling algorithm
# initialize parameters to be estimated by nls
# for each id
  # find row index
  # select t1 to t2 of row index
```

```
  # extract data from monthly gas for above index
  # set qi, build train model on monthy gas extracted above
  # assign nls parameters to variables in data frame

# use given law to predict production rate

# function nlsPowerLaw
nlsPowerLaw = function ( cleanData, idlist, firstDayIndex,lastDayIndex )
{

  cleanData$Di = 0
  cleanData$Dinf = 0
  cleanData$n = 0
  cleanData$qInitial = 0
  cleanData$AIC = 0
  cleanData$RMSEmodel = 0
  cleanData$qPLE = 0
  deleteRows = numeric()

  varNames = colnames(cleanData)

  for (i in 1:length(idlist))
  {
```

```
id1Ind = which(cleanData$Entity.ID == idlist[i]) # index of current ID
# print(id1Ind)

if (firstDayIndex >= 1 && firstDayIndex <= length(id1Ind) &&
    lastDayIndex >= 1 && lastDayIndex <= length(id1Ind) )
{
  qi = cleanData$Monthly.Gas[id1Ind[1]]

  # print(length(cleanData$Entity.ID))

  modelData = cleanData[id1Ind[firstDayIndex:lastDayIndex],]
  # print(modelData)
  print(paste("model is being built from time index ", firstDayIndex, "to", lastDayIndex,
              "ID: ",idlist[i],sep = " "))
  colnames(modelData) = varNames

  # print(i)
  nls.model = nls(data = modelData , log(Monthly.Gas/qi) ~ -Di*newDate^n -Dinf*newDate,
                  start = list(Di = 0.5 , Dinf = 0.5, n = 0.5), control = list(warnOnly = T),
                  algorithm = "port",lower = c(0,0,0), upper = c(1,Inf,1))

  x = coef(nls.model)
  cleanData$Di[id1Ind] = x[1] # Di
```

```
    cleanData$Dinf[id1Ind] = x[2] # Dinf
    cleanData$n[id1Ind] =  x[3] # n
    cleanData$qInitial[id1Ind] = qi
    cleanData$AIC[id1Ind] = AIC(nls.model)
    cleanData$RMSEmodel = RMSE(nls.model)



    print("here")

  }
  else
  {
    deleteRows = updateRowsToDelete(deleteRows,id1Ind)

  }

}

print(deleteRows)

if (length(deleteRows) > 0)
{
  print("deleteRows")
```

```
    cleanData = cleanData[-deleteRows,]
  }

  cleanData$qPLE = with(cleanData,qInitial*exp(-Di*newDate^n - Dinf*newDate))

  print("Power Law Modelling complete...")
  return(cleanData)

}


## function nlsEE
nlsEE = function(cleanData, idlist, firstDayIndex,lastDayIndex)
{
  # initialize parameters to be estimated by nls
  cleanData$Bi = 0
  cleanData$Be = 0
  cleanData$nEE = 0
  cleanData$qEE = 0
  cleanData$qInitial = 0
  cleanData$AIC = 0
  cleanData$RMSEmodel = 0
```

```
varNames = colnames(cleanData)


# for each id
for (i in 1:length(idlist))
{
  # find row index
  id1Ind = which(cleanData$Entity.ID == idlist[i]) # index of current ID


  if (firstDayIndex >= 1 && firstDayIndex <= length(id1Ind) &&
      lastDayIndex >= 1 && lastDayIndex <= length(id1Ind) )
  {
    # set qi
    qi = cleanData$Monthly.Gas[id1Ind[1]]


    # extract data from monthly gas for above index
    modelData = cleanData[id1Ind[firstDayIndex:lastDayIndex],]
    # print(length(modelData))


    print(paste("model is being built from time index ", firstDayIndex, "to", lastDayIndex,
                "ID: ",idlist[i],sep = " "))
    colnames(modelData) = varNames


    # build train model on monthy gas extracted above
```

```
    nls.model = nls(data = modelData , log(Monthly.Gas/qi) ~ -Bi*newDate -
Be*newDate*exp(-newDate^nEE),
                start = list(Bi = 0.5 , Be = 0.5, nEE = 0.5), control = list(warnOnly = T),
                algorithm = "port",lower = c(0,0,0), upper = c(1,Inf,1))

    # assign nls parameters to variables in data frame
    x = coef(nls.model)
    cleanData$Bi[id1Ind] =  x[1] # Bi
    cleanData$Be[id1Ind] =  x[2] # Be
    cleanData$nEE[id1Ind] =  x[3] # nEE
    cleanData$qInitial[id1Ind] = qi
    cleanData$AIC[id1Ind] = AIC(nls.model)
    cleanData$RMSEmodel[id1Ind] = RMSE(nls.model)

   }

  }

  # use Extended Exponential law to predict production rate
  cleanData$qEE = with(cleanData,qInitial*exp(-Bi*newDate - Be*newDate*exp(-
newDate^nEE)))
  print("Extended Exponential Modelling complete...")
  return(cleanData)
```

```
}
## nlsSE
nlsSE = function(cleanData, idlist, firstDayIndex,lastDayIndex)
{
  # initialize parameters to be estimated by nls
  cleanData$tau = 0
  cleanData$nSE = 0
  cleanData$qSE = 0
  cleanData$qInitial = 0
  cleanData$AIC = 0
  cleanData$RMSEmodel = 0

  varNames = colnames(cleanData)

  # for each id
  for (i in 1:length(idlist))
  {
    # find row index
    id1Ind = which(cleanData$Entity.ID == idlist[i]) # index of current ID
    #print(length(id1Ind))

    if (firstDayIndex >= 1 && firstDayIndex <= length(id1Ind) &&
        lastDayIndex >= 1 && lastDayIndex <= length(id1Ind) )
```

```
{
 # set qi
 qi = cleanData$Monthly.Gas[id1Ind[1]]

 # extract data from monthly gas for above index
 modelData = cleanData[id1Ind[firstDayIndex:lastDayIndex],]
 print(length(modelData))

 print(paste("model is being built from time index ", firstDayIndex, "to", lastDayIndex,
          "ID: ",idlist[i],sep = " "))
 colnames(modelData) = varNames

 # build train model on monthy gas extracted above
 nls.model = nls(data = modelData , Monthly.Gas ~ qi*exp(-(newDate/tau)^nSE),
               start = list(tau = 0.5, nSE = 0.5), control = list(warnOnly = T),
               algorithm = "port",lower = c(0,0), upper = c(Inf,1))

 # assign nls parameters to variables in data frame
 x = coef(nls.model)
 cleanData$tau[id1Ind] =  x[1] # tau
 cleanData$nSE[id1Ind] =  x[2] # nSE
 cleanData$qInitial[id1Ind] = qi
 cleanData$AIC[id1Ind] = AIC(nls.model)
```

```
      cleanData$RMSEmodel[id1Ind] = RMSE(nls.model)

    }

  }

  # use Extended Exponential law to predict production rate
  cleanData$qSE = with(cleanData,qInitial*exp(-(newDate/tau)^nSE))
  print("SE Modelling complete...")
  return(cleanData)
}

# nlsDoung
nlsDoung = function(cleanData, idlist, firstDayIndex,lastDayIndex)
{
  # initialize parameters to be estimated by nls
  cleanData$aDng = 0
  cleanData$mDng = 0
  cleanData$qDng = 0
  cleanData$qInitial = 0
  cleanData$AIC = 0
  cleanData$RMSEmodel = 0
```

```
varNames = colnames(cleanData)


# for each id
for (i in 1:length(idlist)) #FIXME_adjust
{
  # find row index
  id1Ind = which(cleanData$Entity.ID == idlist[i]) # index of current ID
  # print(length(id1Ind))


  if (firstDayIndex >= 1 && firstDayIndex <= length(id1Ind) &&
      lastDayIndex >= 1 && lastDayIndex <= length(id1Ind) )
  {
    # set qi
    qi = cleanData$Monthly.Gas[id1Ind[1]]


    # extract data from monthly gas for above index
    modelData = cleanData[id1Ind[firstDayIndex:lastDayIndex],]
    print(nrow(modelData))


    print(paste("model is being built from time index ", firstDayIndex, "to", lastDayIndex,
                "ID: ",idlist[i],sep = " "))
    colnames(modelData) = varNames
```

```
    # build train model on monthy gas extracted above
    nls.model = nls(data = modelData , log(Monthly.Gas/qi) ~ log(newDate^(-mDng)) +
(aDng/(1-mDng))*(newDate^(1-mDng) - 1),
                start = list(aDng = 0.1, mDng = 1.1), control = list(warnOnly = T),
                algorithm = "port",lower = c(0.01,1.01), upper = c(1.99,1.99))

    # assign nls parameters to variables in data frame
    x = coef(nls.model)
    cleanData$aDng[id1Ind] =  x[1] # tau
    cleanData$mDng[id1Ind] =  x[2] # nSE
    cleanData$qInitial[id1Ind] = qi
    cleanData$AIC[id1Ind] = AIC(nls.model)
    cleanData$RMSEmodel[id1Ind] = RMSE(nls.model)

   }

 }

 # use Extended Exponential law to predict production rate
 cleanData$qDng = with(cleanData,qInitial * newDate^(-mDng) * exp((aDng/(1-
mDng))*(newDate^(1-mDng)-1)))
 print("Duong Modelling complete...")
 return(cleanData)
```

```
}

# function RMSEpredict
rmsePredict = function (dataNls, idlist, timeStop, method)
{
  dataNls$RMSEpred = 0
  numPoints = length(dataNls$RMSEpred)
  print(numPoints)
  # for each id
  for (i in 1:length(idlist))
  {
    # extract rowInd for timeStop to end
    rowInd = which(dataNls$Entity.ID == idlist[i]) # index of current ID
    predInd = rowInd[timeStop:length(rowInd)]

    # use monthlyGas and qPLE from above to create a new data frame
    # add conditional for method
    if (method == "PLE")
    {
      temp = data.frame(obs = dataNls$Monthly.Gas[predInd], pred = dataNls$qPLE[predInd])
      print(method)
    }
    else if (method == "SE")
```

```
  {
    temp = data.frame(obs = dataNls$Monthly.Gas[predInd], pred = dataNls$qSE[predInd])
    print(method)
  }
  else if (method == "EE")
  {
    temp = data.frame(obs = dataNls$Monthly.Gas[predInd], pred = dataNls$qEE[predInd])
    print(method)
  }
  else if (method == "Doung")
  {
    temp = data.frame(obs = dataNls$Monthly.Gas[predInd], pred = dataNls$qDng[predInd])
    print(method)
  }



  colnames(temp) = c("obs","pred")
  # find RMSEpred
  dataNls$RMSEpred[rowInd] = myRMSE(temp, numPoints)

}
```

```
  return(dataNls)
}



myRMSE = function(pred.df, numPoints)
{
  result = sqrt((sum((log(pred.df$pred) - log(pred.df$obs))^2))/numPoints)
  return(result)
}



npEurPle = function(inputData,tInf, npStart, npStop)
{
  ids = unique(inputData$Entity.ID)

  results = as.data.frame(ids) # create output data frame
  results$NpPle = 0
  results$eurPle = 0
  results$NpActual = 0

  for (indId in 1:length(ids))
  {
    # select each well
```

```
currId = ids[indId]
rowInd = which(inputData$Entity.ID == currId)

# set variables
n = inputData$n[rowInd[1]]
Di = inputData$Di[rowInd[1]]
Dinf = inputData$Dinf[rowInd[1]]
qInitial = inputData$qInitial[rowInd[1]]



# results
# Np
npInd = rowInd[max(npStart,1):min(npStop, length(rowInd))]
results$NpPle[indId] = sum(inputData$qPLE[npInd]) # should be for each well
results$NpActual[indId] = sum(inputData$Monthly.Gas[npInd]) # should be for each well

# Eur
# calculate qPle at each well
currEur = numeric(tInf)

for (tTime in 1:tInf)
{
  currEur[tTime] = qInitial*exp(-Di*tTime^n - Dinf*tTime)
```

```
    }

    # sum up the values
    results$eurPle[indId] = sum(currEur)

  }

  return(results)
}



npEurSe = function(inputData,tInf, npStart, npStop)
{
  ids = unique(inputData$Entity.ID)

  results = as.data.frame(ids) # create output data frame
  results$NpSe = 0
  results$eurSe = 0

  for (indId in 1:length(ids))
  {
    # select each well
    currId = ids[indId]
```

```
rowInd = which(inputData$Entity.ID == currId)


# set variables
nSE = inputData$nSE[rowInd[1]]
tau = inputData$tau[rowInd[1]]
Dinf = inputData$Dinf[rowInd[1]]
qInitial = inputData$qInitial[rowInd[1]]



# results
# Np
npInd = rowInd[max(npStart,1):min(npStop, length(rowInd))]
results$NpSe[indId] = sum(inputData$qSE[npInd]) # should be for each well

# Eur
# calculate qPle at each well
currEur = numeric(tInf)

for (tTime in 1:tInf)
{
  currEur[tTime] = qInitial*exp(-(tTime/tau)^nSE)
}
```

```
    # sum up the values
    results$eurSe[indId] = sum(currEur)


  }
  return(results)
}

npEurEe = function(inputData,tInf, npStart, npStop)
{
  ids = unique(inputData$Entity.ID)

  results = as.data.frame(ids) # create output data frame
  results$NpEe = 0
  results$eurEe = 0

  for (indId in 1:length(ids))
  {
    # select each well
    currId = ids[indId]
    rowInd = which(inputData$Entity.ID == currId)

    # set variables
    nEE = inputData$nEE[rowInd[1]]
```

```
  Bi = inputData$Bi[rowInd[1]]
  Be = inputData$Be[rowInd[1]]
  qInitial = inputData$qInitial[rowInd[1]]



  # results
  # Np
  npInd = rowInd[max(npStart,1):min(npStop, length(rowInd))]
  results$NpEe[indId] = sum(inputData$qEE[npInd]) # should be for each well

  # Eur
  # calculate qPle at each well
  currEur = numeric(tInf)

  for (tTime in 1:tInf)
  {
    currEur[tTime] = qInitial*exp(-Bi*tTime - Be*tTime*exp(-tTime^nEE))
  }

  # sum up the values
  results$eurEe[indId] = sum(currEur)

}
```

```
  return(results)
}

npEurDoung = function(inputData,tInf, npStart, npStop)
{
  ids = unique(inputData$Entity.ID)

  results = as.data.frame(ids) # create output data frame
  results$NpDoung = 0
  results$eurDoung = 0

  for (indId in 1:length(ids))
  {
    # select each well
    currId = ids[indId]
    rowInd = which(inputData$Entity.ID == currId)

    # set variables
    aDng = inputData$aDng[rowInd[1]]
    mDng = inputData$mDng[rowInd[1]]
    qInitial = inputData$qInitial[rowInd[1]]
```

```
  # results
  # Np
  npInd = rowInd[max(npStart,1):min(npStop, length(rowInd))]
  results$NpDoung[indId] = sum(inputData$qDng[npInd]) # should be for each well

  # Eur
  # calculate qDoung at each well
  currEur = numeric(tInf)

  for (tTime in 1:tInf)
  {
    currEur[tTime] = qInitial * tTime^(-mDng) * exp((aDng/(1-mDng))*(tTime^(1-mDng)-1))
  }

  # sum up the values
  results$eurDoung[indId] = sum(currEur)

 }
 return(results)
}
```

Appendix D: Clustering results using EE, SE, DNG DCA's in Permian

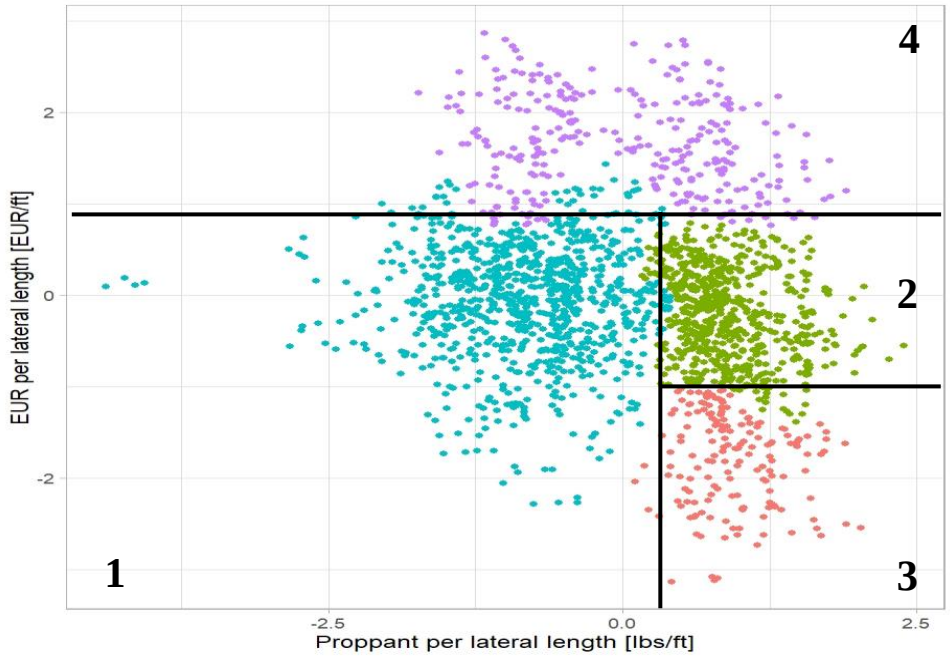

Figure 30: Representation of wells using clusters to show the performance of each well using EE in the Permian basin

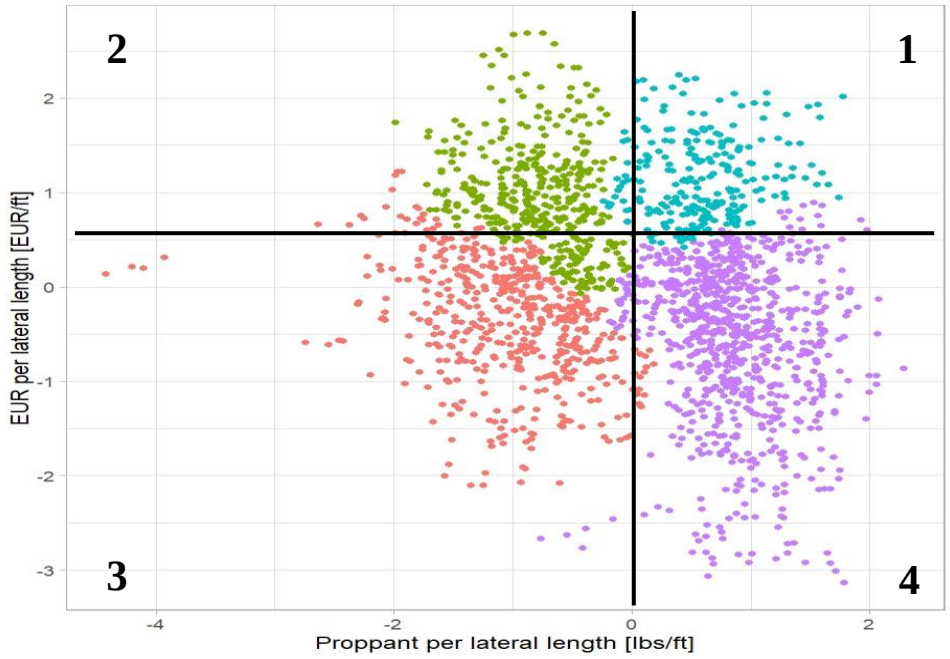

Figure 31: Representation of wells using clusters to show the performance of each well using SE in the Permian basin

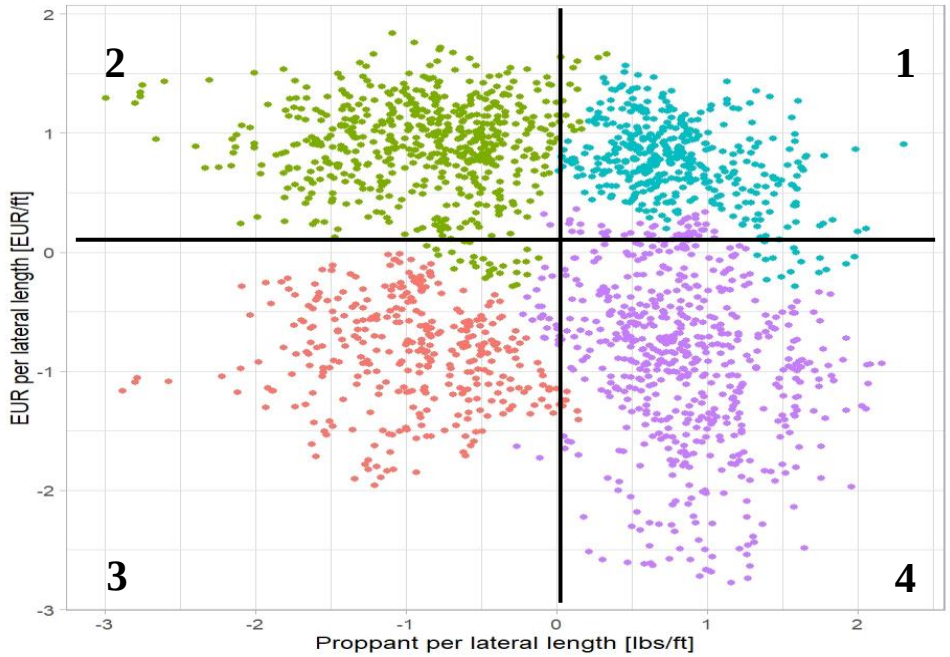

Figure 32: Representation of wells using clusters to show the performance of each well using DNG in the Permian basin