The Dissertation Committee for Chiranth Manjunath Hegde Certifies that this is the approved version of the following dissertation:

End-to-end Drilling Optimization using Machine Learning

Committee:

Kenneth Gray, Supervisor

Hugh Daigle

Harry Millwater

Michael Pyrcz

Afif Halal

End-to-end Drilling Optimization using Machine Learning

by

Chiranth Manjunath Hegde

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

The University of Texas at Austin

August 2018

Dedication

To Ajji, Amma, and Appa.

Acknowledgments

As a freshman in college at the National Institute of Technology Karnataka in Surathkal, I never thought I would pursue a doctorate in engineering, especially in a field that I am so passionate about. Over the course of my undergraduate degree, I became more curious about science and engineering, so much so that at the end of my bachelors I applied to pursue a Ph.D. in Petroleum Engineering.

The opportunity for me to come to the Austin would not have been possible without the help of my graduate research advisor, Dr. Kenneth E. Gray, who has been a source of advice, mentorship, and guidance since our first email correspondence. Dr. Gray's knowledge about the industry stretches over the duration of the field's development, and his overview has been instrumental in shaping my research. His inputs have always been valuable and his enthusiasm to adopt new technologies never cease to amaze me.

I would like to thank Dr. Karra Ram Chandar who helped me with my initial research endeavors in Surathkal – coaching me and providing me with opportunities which inspired me to pursue a graduate degree. I thank Srisharan Shreedharan for being my peer, research partner, roommate, and friend over the course of my undergraduate degree. My desire to compete with you made me work harder.

I would also like to thank Dr. Hugh C. Daigle and Dr. Harry R. Millwater for being on my committee and providing a valuable critique of my work over for the past four years. I thank Dr. Michael Pyrcz and Dr. Afif Halal for being on my Ph.D. committee.

I would like to thank all the people I met during my summer internships in industry. My internships over the summer have always been a fruitful experience in which I learned more than I could have imagined. Each internship has helped me push my boundaries even further.

I would like to thank my family for supporting my decision to pursue my graduate studies thousands of miles away from home. I thank Suchitra Hegde, Manjunath Hegde, Girija Rajivalochan, Savita Rajivalochan, and Ishwar Hegde.

Several people have been a constant source of support over the past four years throughout my Ph.D. A big thank you to Shalini Sinha for listening to my rants, problems, accomplishments, and aspirations over the past four years. I'm grateful for the support of my friends Taussef Siraj, Sai Uppati, Vivek Ramachandrann, Arvind Krishnan, Vivek Prakash, Scott Wallace, Matt Ramos, Andrew Johnson, Colin Schroeder, Martha Gross, Taylor Isbell, Ananya Ravi, Shruti Mandhani, Vikas Kamat, and Nikitha Shiv.

I thank Ana Dison for giving me the opportunity to mentor undergraduate students for four semesters as a part of the GLUE program. Part of my love for teaching and mentoring can be attributed to this program. Ana has been a constant source of support helping me apply for scholarships and travel grants over the course of my Ph.D.

I would also like to thank sponsors of the Wider Windows Industrial Affiliate Program: British Petroleum; BHP Billiton; Chevron; ConocoPhillips; Halliburton; Marathon Oil; National Oilwell Varco; Occidental Oil and Gas; and Shell Oil Company. Research guidance and constructive critiques at program review meetings were very

helpful. Additionally, field data sets provided by Chevron, Marathon, and National Oilwell Varco for use in model validations are gratefully acknowledged.

Finally, I would like to thank the other students in my research group. Our meetings have been very useful, and I wish you all the best in your academic and professional careers.

Abstract

End-to-end Drilling Optimization using Machine Learning

Chiranth Manjunath Hegde, Ph.D.

The University of Texas at Austin, 2018

Supervisor: Kenneth Gray

Drilling costs occupy a significant portion of oil and gas project’s budget. Optimization of drilling - increasing speed, reducing vibrations, and minimizing borehole instability - can lead to significant savings and hence have been extensively studied. Currently, most drilling optimization tools (or models) only tackle a single drilling metric: they seek to optimize either the rate of penetration (ROP), torque on bit (TOB), mechanical specific energy (MSE) or drilling vibrations. Models are often built independent of other metrics (without coupling) and do not accurately represent downhole conditions since drilling metrics are interrelated. This may lead to over or underestimation of the metric optimized which can severely reduce the effect of optimization.

The objective of this dissertation is to introduce techniques, strategies, and algorithms that can be used to build a fully coupled drilling optimization model. Drilling optimization is studied by first optimizing ROP– where models for ROP prediction and inference are constructed using machine learning. Strategies and algorithms for determining optimal drilling parameters using ROP models are discussed. The unique problem posed by data-driven models are solved using meta-heuristic algorithms.

A coupled model is constructed by building ROP, TOB, and MSE models conjointly using the random forests algorithm. Drilling vibrations – axial, lateral, and torsional – are modeled using a machine learning classification algorithm. This classification algorithm used to restrict the optimization space, ensuring that optimal parameters do not induce vibrations ahead of the bit. This model is used to investigate the effect of optimizing ROP and MSE on field data.

A workflow is introduced linking all the aforementioned models into an end-to-end drilling optimization tool. The tool can be used as a recommendation system where field-measured data are used to determine and implement optimal drilling parameters ahead of the bit. The dissertation illustrates the use of statistical (or machine) learning techniques to address the problems encountered in drilling optimization.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Oil and gas will account for more than half of the world's energy by 2040 (Petroleum, 2016). Oil and gas are extracted from rocks (or reservoirs) found deep below the earth's surface. Drilling is a process used to gain access to these reservoirs by excavating rock. This process creates a circular wellbore with the use of a drill bit, a rotating mechanism, a drill string (used to transfer weight to the bit), and drilling fluid (used to counteract downhole pressure, bit cooling and transfer of excavated rock to the surface). The first documented spring hole well was drilled in 1806 to a depth of 58 feet by David and Joseph Ruffner (Pees, 2004). Today, technological advances have led to wells reaching measured depth and horizontal reaches of over 10000 meters. Since drilling is unobservable, measurement tools are used to infer downhole conditions. Data from these measurement tools collected on the surface of the rig as well as downhole (near the bit) can be analyzed to improve the process. This thesis will discuss strategies, workflows, models, and algorithms which can be used to analyze, improve, and optimize drilling using data.

1.1 Motivation and Problem Description

Drilling occupies a substantial portion of an oil and gas project budget; hence any cost-saving measure directly relates to reduced costs. The cost of drilling wells over the past couple of years have increased inspite of technological advances (Aerts, Guus; Brun, 2015). The cost of drilling a well varies depending on the (geographic) location, geology of the reservoir, type of play (onshore vs offshore), technologies used, operator, service company, and expertise of staff. According to a recent report published by the EIA (EIA, 2016), the average cost for drilling an onshore well in North America is around $6-7 million (Figure 1.1) and an offshore well can range from $50 million to $200 million. Moreover, drilling operational costs occupy a sizable portion of well's costs: 30-40% (onshore) and around 60% (offshore). Given the expense of drilling to reach oil and gas deposits, improving drilling efficiency can be imperative to the success of an oil and gas project.

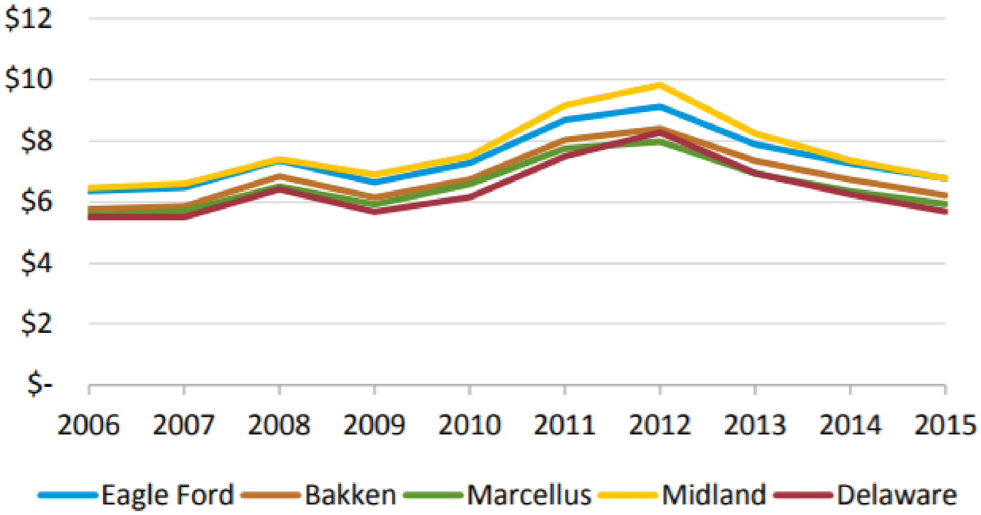

Figure 1.1: Average cost of drilling wells in North American shale plays from 2006-2015 (EIA, 2016)

Over the past couple of decades, engineers and researchers have made efforts to reduce the cost of drilling by increasing the speed of drilling, avoiding drilling dysfunction (by reducing drilling vibrations), and minimizing the amount of energy used to excavate rock. These cost-saving efforts include the optimization of different drilling metrics or key performance indicators (KPIs) such as rate of penetration (ROP), the intensity of drilling vibrations, mechanical specific energy (MSE), and cost per foot of drilling. Other factors which affect the overall drilling optimization process such as hole quality, borehole assembly (BHA), geomechanics, bit hydraulics, and mud selection have been studied. The objective of these optimization efforts is to reduce operating costs due to drilling while ensuring a high-quality wellbore is drilled (which can be used for oil and gas production with minimal treatment). This dissertation will address the optimization of drilling operational related metrics such as ROP, MSE, and cost-per-foot.

The rate of penetration (ROP) during drilling is a direct measure of the time taken to drill a well, apart from other times involved such as trips, bit change, and downtime. Controlling the ROP can be extremely important in drilling since it directly relates to time saved. Hence, maximizing ROP using data measured on the rig is a common (and most widely adopted) goal for drilling optimization. Mechanical specific energy (MSE) measures of the amount of energy required to break a unit volume of rock. If the MSE is minimized while drilling a well, least amount of energy will be expended in excavating or breaking the rock – maximizing the productivity of drilling. Vibrations are an unwanted effect of the drilling process; excessive vibrations damage the drilling bit (which may result in premature bit change), lead to waste of energy, and cause drilling dysfunction. According to literature (Wang et al., 2010), around 40% of drilling dysfunctions can be attributed to drilling vibrations. Reducing vibrations should automatically result in the better use of energy imparted to the bit (reducing MSE) and improve the productivity of the well.

The modeling and optimization of these key drilling metrics have not been consolidated – description of the entire process is scarce, even with respect to a single drilling metric. While many prediction techniques and algorithms have been published, only a few papers describe strategies to use these algorithms to improve ROP or MSE. The most widely used drilling models are empirical models which were developed decades ago based on laboratory experiments. In this new age, where large amounts of data are collected during drilling operations, several statistical and machine learning techniques can be applied to optimize drilling, perhaps more efficiently. Additionally, the popular models used today have many limiting assumptions – many of which do not hold. For example, some ROP optimization models assume the absence of drilling vibrations (Caicedo, Calhoun, & Ewy, 2005). Drilling models are also built independent of other metrics in drilling – there is no coupling between interrelated drilling parameters such as ROP, MSE, torque and drilling vibrations. As a result, these models do no accurately describe downhole conditions.

This dissertation attempts to demystify drilling optimization. It will explore prediction, inference, and optimization of several metrics which can be used for drilling optimization. Algorithms and strategies are discussed to optimize and couple interrelated metrics in a complex drilling environment.

1.2 Objectives

The objective of this dissertation is to address several research questions which are essential in the design of an end-to-end drilling optimization tool (or workflow).

ROP has always been the most popular metric which is used to measure drilling productivity. However, these models used for ROP predictions were discovered half a century ago. New algorithms leveraging statistical and machine learning techniques (data-driven techniques) need to be evaluated as potential tools to predict ROP. These developed ROP models can be used for determining optimal parameters ahead of the bit which when implemented increase ROP. Optimal parameters are determined using a suitable optimization algorithm – which depends on the ROP model employed. Meta-heuristic algorithms are evaluated to optimize data-driven ROP models and gradient-based algorithms for empirical ROP models. Their applicability is studied as a real-time and post-drilling processing tool.

Algorithms to combine empirical models and machine learning models – hybrid models – have been developed. These models provide tradeoffs between model prediction accuracy and interpretability. Additionally, they can be used for ROP model interpretation: a useful tool to make on-the-fly drilling decisions.

Most ROP models and optimization workflows completely ignore the effect of drilling vibrations: the biggest ROP inhibitor. Existing vibration models exist predominantly for planning purposes. Real-time vibration prediction tools are scarce. The use of machine learning for real-time vibration intensity classification has yet to be studied.

Only one model considers the effects of ROP, vibrations, and MSE (together) – ExxonMobil's drilling advisory system (DAS). Even this model does not couple bottom hole drilling parameters which are known to be interrelated. The development of a fully coupled drilling optimization model has been outlined.

The quantification of ROP optimization on other drilling metrics such as MSE, TOB, and vibrations requires the use of a coupled model. A coupled model can be used to evaluate the effects of optimizing one drilling metric over another.

1.3 Outline

The outline of this thesis roughly follows the research objectives that are defined in section 1.2. *Chapter 2* describes in detail the ROP modeling process. ROP is modeled using traditional (or empirical model) models – which are the workhorse of the industry – and data-driven models. Several models are evaluated based on their predictive accuracy by comparing them to data measured in the field. Torque-on-bit (TOB) modeling using data-driven methods are also discussed. *Chapter 3* develops a new class of models – the hybrid models – which combine traditional and data-driven models. These models provide a tradeoff between the two models – providing higher accuracy than traditional models while not losing much model interpretability, unlike the data-driven models. *Chapter 4* discusses ROP optimization. The models are "inverted" to determine optimal rig-based parameters which when implemented ahead of the bit will improve ROP. Several optimization algorithms are evaluated to optimize traditional and data-driven models. The intuition behind the success of data-driven models in ROP prediction and optimization is discussed. *Chapter 5* illustrates the effect of different metrics for drilling optimization. ROP, MSE, TOB, and cost-per-foot are evaluated as potential metrics to be optimized while drilling. *Chapter 6* lays out a method to model drilling vibrations in real-time using surface and downhole data. Axial, lateral or torsional vibrations are modeled using machine learning classification algorithms. This vibration model described herein can be used to constrain the optimization algorithm such that the selected optimal parameters do not induce drilling vibrations. *Chapter 7* combines previously described concepts into two drilling optimization models or workflows. The first model – TVOPT – is a ROP optimization model which controls drilling vibrations. It uses traditional ROP models to predict ROP, gradient ascent (constrained by the vibration model) to find optimal drilling parameters. The second model – coupled machine learning optimization model (CMOPT) – is a fully coupled end-to-end drilling optimization model based on machine learning. *Chapter 8* summarizes the previous chapters, summarizes the research conducted and identifies areas for future work. The appendices cover added details which are left out of the chapters.

Chapter 2: Prediction of Rate of Penetration (ROP) and Torque on bit (TOB) during drilling[1],[2]

This chapter compares two strategies for ROP modeling while evaluating their accuracy, reliability, and effectiveness. Physics-based (deterministic, traditional or empirical) models and data-driven (statistical learning or machine learning) models are used for ROP prediction in drilling a vertical well with a PDC bit. ROP is a function of many variables which include, but are not limited to, parameters on the surface, lithology, geology, bit design, mud, human factors, downhole conditions, and mud rheology. There are bit-specific models, lithology-specific models, and operator-preferred models. This chapter will address the preferred traditional models in the industry which are the Bingham(Bingham, 1964), Motahhari (Motahhari, Hareland, & James, 2010), Hareland (Hareland & Rampersad, 1994) and the Bourgoyne and Young Model (Bourgoyne Jr & Young Jr, 1974). The traditional models require weight on bit, flow rate, the strength of rock and rotary speed (RPM) as input parameters for ROP prediction. Data-driven models are built using the same input parameters. Both classes of models are analyzed to show that machine learning models perform better in terms of accuracy and reliability. Models are analyzed on different formations to ensure a robust evaluation. Both types of the models are evaluated by running simulations on data measured during drilling a well in the Williston Basin, North Dakota. Results indicate that for the same inputs, data-driven

[1] Hegde, C., & Gray, K. E. (2017). Use of machine learning and data analytics to increase drilling efficiency for nearby wells. *Journal of Natural Gas Science and Engineering*, *40*, 327–335. https://doi.org/10.1016/j.jngse.2017.02.019

[2] Hegde, C., Daigle, H., Millwater, H., & Gray, K. (2017). Analysis of rate of penetration (ROP) prediction in drilling using physics-based and data-driven models. *Journal of Petroleum Science and Engineering*, *159*, 295–306. https://doi.org/10.1016/j.petrol.2017.09.020

The author of this thesis is the primary author of both papers models perform more accurately (lower error in prediction) and more reliably (higher R^2). This chapter also utilizes data-driven models to model the torque-on-bit (TOB).

2.1 ROP MODELS

The rate of penetration (ROP) measures the speed at which a wellbore is drilled. The ROP can be calculated by measuring the depth drilled in an interval of time with the units ft/hr. Higher ROP implies faster drilling: better rig performance, increasing the productivity of the rig. ROP along with a few other parameters can indicate drilling kicks (flow of formation fluids into the wellbore while drilling), over- or under pressured conditions, and stick-slip in drilling (absorption and release of energy due to torsional vibrations), adding to its importance. Since ROP is a direct measure of the time taken to drill a well, apart from other times involved such as trips, bit change, and downtime, controlling the ROP can be extremely important in drilling. Maximizing ROP is one form of optimizing drilling which reduces drilling time. This section will introduce different strategies to model the rate of penetration in drilling; these models can be used to predict and control drilling rates. The strategies predict ROP using drilling control parameters – parameters which can be controlled on the surface of the rig such as rotary speed (RPM), weight-on-bit (WOB) or flow-rate. ROP modeling techniques can be classified into three broad categories: traditional models, data-driven models, and hybrid models (covered in Chapter 3). Traditional models are based on equations which were developed based on experimental results and field-based intuitions. Data-driven models include the use of machine learning or statistical learning algorithms to develop a model for ROP prediction. Hybrid models combine these two concepts by using machine learning algorithms to combine or fit deterministic models. They can be used for prediction or model inference –

providing trade-offs between accuracy and interpretability – and will be covered in detail in Chapter 3.

2.1.1 Traditional ROP Models

Given the interest in ROP prediction and analysis, traditional models were developed for ROP prediction based on laboratory experiments. These models have been improved over the past few decades to incorporate advances such as bit technology, drilling in unconventional reservoirs, or introduction of more parameters. Most of these models have been based on the physics of drilling with empirical coefficients to incorporate changes in lithology, geology, and other factors not readily measured. Empirical coefficients are determined and adjusted as the well is drilled. Data acquired during drilling (or data from pad/offset wells) can be used to determine these empirical coefficients. The empirical coefficients are constrained (by defining upper and lower bounds) so that the values utilized are based on physics-based and engineering judgment.

One of the earliest ROP models was developed by Maurer (Maurer, 1962) where the author applied a rock cratering approach developing a ROP formula for roller-cone bits. The parameters included in the model are weight-on-bit (WOB), rotary speed (RPM), bit diameter and rock strength. This paper introduced an important concept called rock floundering: beyond a certain WOB, there was no improvement in ROP because of the reduction in hole cleaning ability. The accumulation of cuttings around the bit makes it too difficult clean at the bit which would impair ROP improvements due to increasing WOB. Hence, increasing WOB beyond a certain limit would not result in an increased ROP. A ROP prediction model introduced by Bingham (Bingham, 1964) – hereby referred to as the Bingham model – uses weight on bit (WOB), rotations per minute (RPM), and bit diameter as inputs. An empirical constant ‘k’ was used as a formation dependent parameter. This model stressed the importance of hole cleaning ability and its relation to ROP. A model introduced by Eckel (Eckel, 1967) incorporated the effects of drilling mud on ROP. A Reynolds number function was used to correlate ROP with mud properties. It was shown that an increase in the Reynolds number function correlated well with high ROP measurements. Based on this paper it was concluded that a mud with a low kinematic viscosity would be recommended for easier drilling or higher ROP yield.

Bourgoyne and Young's (BY) model (Bourgoyne Jr & Young Jr, 1974) introduced a more sophisticated model – hereby referred as the BY model – with additional parameters to include more physical and geological aspects involved in drilling. This model is perhaps the most comprehensive traditional model to date which describes ROP. The model contained eight input parameters: formation strength, normal compaction trend, under compaction, differential pressure, bit diameter and bit weight, rotary speed, tooth wear, and bit hydraulics.

Walker (Walker, Black, Klauber, Little, & Khodaverdian, 1986) introduced a model which utilized triaxial rock strength tests and the Mohr-Coulomb failure criterion to develop a roller cone ROP equation dependent on WOB, borehole pressure, rock porosity, average grain size, and in-situ formation compressive strength. Winters (Winters, Warren, & Onyia, 1987) developed a model which separated the effects of drilling into physical breakage of the rock and hole cleaning. This model works well in low differential pressure but fails in cases of higher differential pressures (Soares, 2015).

A modified model for ROP modeling of drag bits was introduced which helped improve ROP accuracy (Hareland & Rampersad, 1994) – hereby referred to as the Hareland model. The original drag bit model contained three empirical parameters to model lithology and other eccentric factors. This ROP model was used to reduce the cost per foot of drilling wells in the North sea successfully (Bratli, Hareland, Stene, Dunsaed, &

Gjelstad, 1997; Nygaard, Hareland, Budiningsih, Terjesen, & Stene, 2002). This model was later modified to include the effect of the confined compressive strength of rock (Motahhari et al., 2010) and the effect of PDMs (positive displacement motors) – hereby called the Motahhari Model. It is still used today given the prolific use of PDC bits in drilling. This chapter will evaluate traditional ROP models using one model from each era of ROP modeling – Bingham model, BY model, Hareland model, and Motahhari model.

2.1.2 Data-driven ROP Models

The intuition behind data-driven techniques is to build ROP models purely based on data collected during drilling utilizing surface measured parameters such as weight on bit, rotations per minute, and flow rate to predict ROP. Machine learning can be used for accurate ROP prediction during drilling within a given facies or even for multiple facies in succession (with adequate training data). Machine Learning (ML) methods are advantageous since they do not contain any empirical constants or drill bit specifications and are not bound to a specific borehole assembly (BHA). The ML model predictions depend only on the input data, and selection of input parameters to the model (commonly known as features in machine learning circles).

Neural networks, a nonlinear statistical model, has been used for predicting ROP (Bilgesu, Tetrick, Altmis, Mohaghegh, & Ameri, 1997). Further exploration of neural networks using different input parameters for ROP prediction has been studied (Jahanbakhshi, 2012). A model using two input parameters – RPM and WOB – was used within a similar framework for ROP optimization through automation (Dunlop et al., 2011). Neural networks are powerful tools which are well suited for high dimensional modeling (Krizhevsky, Sutskever, & Hinton, 2012; LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015), however, underperform (when compared to several simpler statistical learning techniques) for low dimensional problems. This led to the use of simpler statistical learning techniques such as bagging and random forests which predict ROP with a higher accuracy as compared to previously used data-driven techniques (Chiranth Hegde, Wallace, & Gray, 2015).

2.2 Dataset and Methodology

Experiments are conducted on a depth-based drilling data set spanning from 6001 ft to 9128 ft true vertical depth (TVD). Data from twelve of these formations have been used for ROP prediction analysis. This data was acquired from a vertical section of a well drilled in the Bakken shale. The data contains drilling parameters measured on the surface in a depth-based format – per 0.25 ft of drilled depth. Depth based data or filtered times series data are preferred for ROP modeling since they tend to be less noisy (G S Payette et al., 2017; S. P. Wallace, Hegde, & Gray, 2015), and they provide an easy means to overlay these data with the depths of formation tops (created by geologists). A simplified stratigraphic column is shown in Figure 2.1 (left). The entire interval of data used for model validation in this dissertation was drilled with the use of a Smith 616 PDC bit. ROP has been plotted against depth and color-coded by each formation in Figure 2.1 (right). Experiments are conducted to test the accuracy of traditional and data-driven models for the ROP prediction by comparing the predicted values to the data measured in the field.

2.2.1 Experiments

Models are individually built on each formation. Traditional models are conditioned to fit the data; data-driven models are built based on the data. In the case of traditional models, the functional relationship of the input parameters remains constant and empirical values are tuned to fit the data. However, in the case of data-driven models, the data is used to learn (or form) these functional relationships between the input and output features based on data. Figure 2.2 outlines the implementation of ROP prediction simulations in this chapter. A portion of the well is drilled (without any modeling), and the data collected during drilling this interval are called the training data. In Figure 2, 100 ft. of drilled data has been used as training data. All traditional models have empirical constants which depend on the geological properties of the rock, bit design, and drilling conditions. These empirical constants must be determined or calibrated to each formation. One way to calibrate them is by using the training data to determine the empirical coefficients of the traditional models - conditioning the models to the data so that they can be used to predict ROP ahead of the bit. A data-driven model is built on the training data using a fitting or learning algorithm. As opposed to just calibrating the data-driven modeling using training data, the functional relationship between these input parameters is learned using this data. Both the traditional and data-driven models may then be tested for prediction accuracy and goodness of fit (R^2) over an arbitrary test data interval as illustrated in Figure 2.2. The paper uses this simulation experiment to determine the accuracy and reliability of both types of models. Incorporation of data-driven models in field operations will be addressed in later chapters. The data-driven models use the same input parameters – to train models – as those used in traditional models (RPM, weight-on-bit, flow rate, and rock strength). This ensures that the physics between the two models remains comparable, often called feature engineering in the machine learning circles. ROP may be a function of many other input parameters such as pressure at the bit, cutting cleaning at the bit, mud weight, and mud type. However, it is assumed that the effects of these other parameters are represented in the data that is used to build the model. The output parameters of both models are ROP.

The predicted ROP – using both models – are compared to the actual measured data (test data). This comparison provides a way to evaluate each model. It is necessary to avoid overfitting and underfitting the data with the developed model. If a model has low training error and high test error, it is said to have overfit the data. Overfitting occurs due to excessive variance of the model and underfitting due to excessive bias of the model. Bias is accumulated when a simple model is used to explain a complex real-world phenomenon. Variance is associated with the error accumulated in the model when data sets are changed – it measures the ability of the model to generalize. The bias-variance tradeoff is a constant theme in machine learning (James et al., 2013); picking the best model – which results in low bias and low variance – is often best managed by cross-validation.

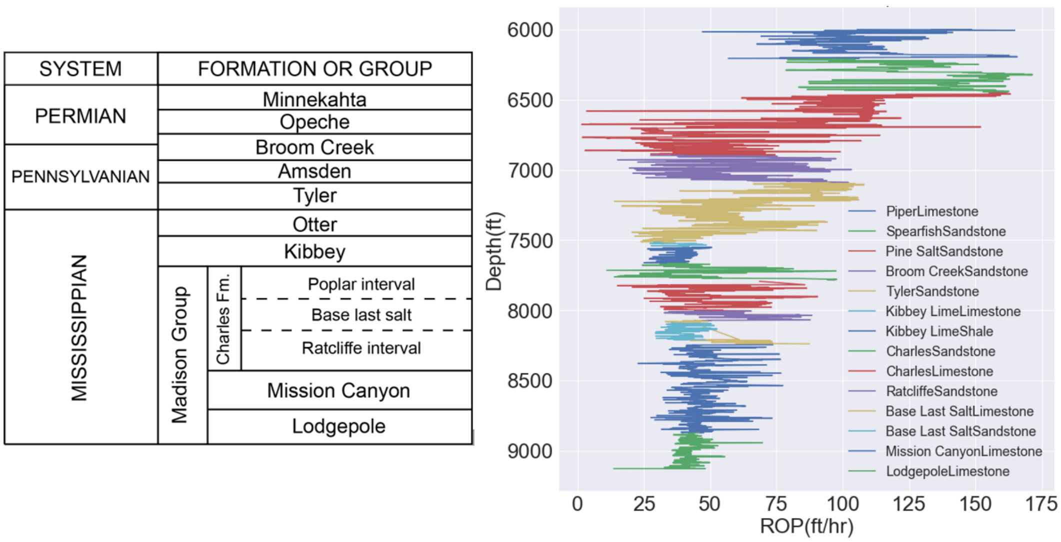

Figure 2.1: (left) Generalized stratigraphic column for the Williston Basin, North Dakota (Theloy, 2014); (right) ROP vs Depth plot over different formations in a vertical section of a well drilled in Williston Basin, North Dakota. ROP, TOB, and Vibrations models are built on field data collected surface and downhole while drilling this well (C. M. Hegde & Gray, 2018)

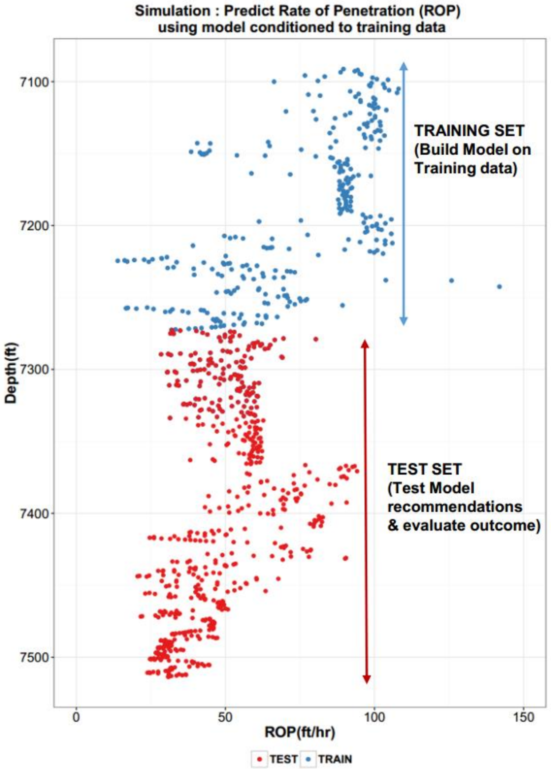

Figure 2.2: ROP simulation experiment schematic showing training and test set. The training set is utilized to build the model or determine empirical constants. The test set is used to evaluate the model. The test set is not used for model training to ensure an unbiased evaluation of model accuracy. Since the main aim of ROP modeling is prediction ahead of the bit – the test set in this case is data ahead of the bit

2.2.2 Data Exploration and Feature Selection

A pairs plot is often used to plot the data and measure correlations between field variables (Figure 2.3). Pairs plots are particularly useful to discern patterns in data. For example, a simple linear relationship between two parameters (if seen in the pairs plot) can be modeled using linear models. Each plot in Figure 2.3 has a "window", which can be numbered for easy evaluation. Numbering is similar to matrix indexing, i.e. window(i,j) would represent a window in the i^{th} row and j^{th} column. The X-axis in each subplot represents the units for an i^{th} input parameter. Hence each input parameter or each window would have an *X* and *Y* axis representing the data plotted in that window. Each window in the plot contains two parameters plotted against each other. For example, window (1,2) plots depth on the *X-axis* and ROP on the *Y-axis*. Window (2,1) displays the correlation of the variables plotted in the window (1,2). Input features for data-driven models can be selected using the correlation between control and response variables. An analysis of the pairs plot can result in discarding some input features based on low correlation to the target or redundancy. The analysis of this pairs plot shows that none of the variables are highly correlated with each other – changes of collinearity are low.

2.2.3 Train, Validation and Test Sets

Data are divided into different sets to avoid overfitting. The training set include the data used to fit the model or train an algorithm; the validation set is used to pick hyper-parameters or fine-tune models; the test set is the held-out (or blind) set used for an unbiased evaluation of model error (Figure 2.2). Data are partitioned into these sets to balance the bias-variance tradeoff which exists in these models (James et al., 2013). A model is said to have excessive bias or underfit when a very simple algorithm is used to model a complicated process – if linear regression is used to model nonlinear data. A model is said to have excessive variance (and overfits the dataset) if the performance of the model deteriorates when applied to new data. In the case of overfitting, the model fits the noise present in the original dataset; it is unable to generalize to new data and performs poorly when deployed into production. Underfitting is common when dealing with traditional models since simple (power-law based or convex-like) models are used to model ROP. Overfitting is a common phenomenon associated with statistical and machine learning models since more complex algorithms tend to fit the noise in measurements. The tradeoff between over and underfitting is a bias-variance trade-off visualized for a hypothesized case in Figure 2.4.

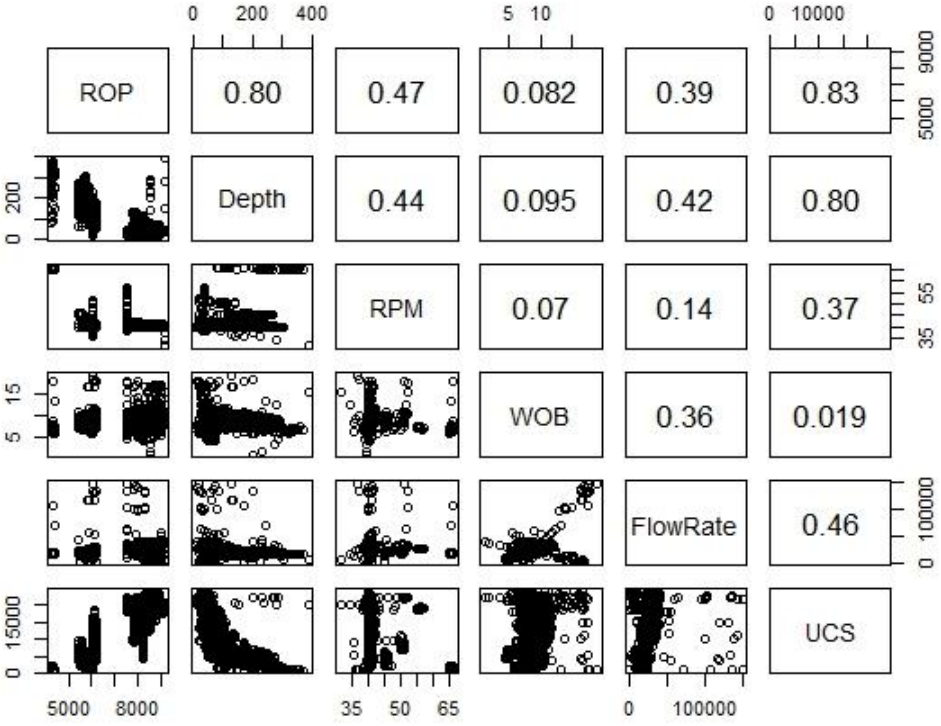

Figure 2.3: Pairs plots of drilling parameters evaluated for field data. Each drilling parameters is plotted as a scatter plot against another measured parameter. The Top right triangle of the plot matrices shows the correlation between the two variables plotted in that window.

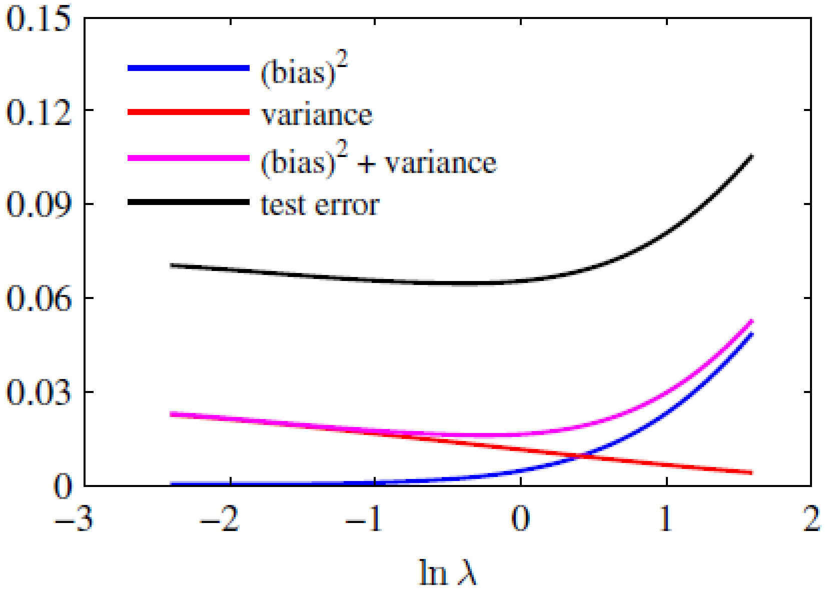

Figure 2.4: Plot of squared bias and variance, together with their sum for a simulated dataset. Also shown is the average test set error for a test data set the size of 1000 points. The minimum value of (bias)^2 + variance occurs around ln λ = −0.31, which is close to the value that gives the minimum error on the test data. In this figure, λ is a hyperparameter used to vary the flexibility of the model which controls the bias and variance (Bishop & Christopher, 2016)

The bias-variance tradeoff stems from the definition of error and the expected value of the test error (Equation 2.1).

$\mathrm{E}(\mathrm{Y} - \widehat{\mathrm{Y}}) = \mathrm{Var}(\widehat{\mathrm{Y}}) + Bias(\widehat{\mathrm{Y}})^2 + Var(\epsilon)$ (Equation 2.1)

where, Y is the response, $\widehat{\mathrm{Y}}$ is the estimated function, ϵ is the irreducible error (most often normal; occurs due to measurement error, sensitivity of instruments etc.). It is common practice to use a validation set along with the training set to evaluate the performance of the trained model on the test set. The validation set is a small portion of the training set – 20-30% – used to estimate test error and fine-tune the model. The main disadvantage of the validation set approach is the loss of data for training.

Cross-validation solves this problem by reusing the training set for validation repeatedly. In *K*-fold cross-validation the dataset is split into *K* parts. If *K*=5, the training set would be split in a 1:5 ratio. The smaller split is used as a validation set and the larger split for training, i.e., 4 parts training and 1-part validation. In each iteration, 4 parts of the data would be used for training and one part for validation. This is randomized and repeated until each of the five parts is used for validation. Another similar method commonly employed is the leave out one cross-validation (LOOCV), where all but 1 data point is used for training and the model is tested on the single left out data point. This process is repeated until all points have been tested (or all points have been left out). Analysis has shown that K-fold cross-validation results in a better estimate of test error as compared to vanilla cross-validation or leave-one-out-cross-validation (James et al., 2013).

2.3 THEORY AND CALCULATIONS

This section explores the theory behind each traditional and data-driven model explored in this chapter. Broadly models can be classified as:

- Traditional (Physics-based) models
    1. Bingham model
    2. Motahhari Model
    3. Hareland Model
    4. Bourgoyne and Young (BY) Model
- Data-driven models
    1. Linear Regression Model
    2. K Nearest Neighbors Model
    3. Tree based Model
    4. Bagging Model

5. Random Forest Model
6. Neural Network Model

- Hybrid Models (covered in chapter 3)
    1. Hybrid - One
    2. Hybrid - N

2.3.1 Traditional Models

It is important to look at the governing equations of the physics-based models. This will lead to an understanding of the input parameters, and their importance in drilling. Bingham's model (Bingham, 1964) was the earliest of the traditional models considered in this chapter. The model was designed to be applied to any bit-type.

$ROP = aRPM\left(\frac{WOB}{D_b}\right)^b$, (Equation 2.2)

where ROP is the rate of penetration (ft/hr), WOB is the weight on bit (klbs or lbs), RPM is the rotary speed of the drill (revolutions/sec), D_b is the bit diameter (in), and 'a' and 'b' are constants determined for a given rock formation. These constants can be determined by using a fitting algorithm (such as least squares) on the training data. The constants represent a quantification of the ease of drilling through a particular formation.

Hareland's model (Hareland and Rampersad, 1994) proposed a bit-specific model – specific to the drag bit – which is the second traditional model used in this paper.

$ROP = 14.14N_cRPM\frac{A_v}{D_b}$, (Equation 2.3) where N_c is the number of cutters, Av is the area of rock compressed ahead of a cutter (in^2), and the other variables repeat themselves from the Bingham model. A_v is set based on the type of drag bit; in the case of a polycrystalline diamond cutter (PDC) bit it can be formalized as:

$$A_v = \cos\alpha \sin\theta\left(\left(\frac{d_c}{2}\right) cos^{-1}\left(1-\frac{4WOB}{cos\theta\pi N_c\sigma_c d_c^2}\right)-\left(\frac{2WOB}{cos\theta\pi N_c\sigma_c}-\frac{4WOB^2}{(cos\theta\pi N_c\sigma_c d_c)^2}\right)^{0.5}\left(\frac{WOB}{cos\theta\pi N_c\sigma_c}\right)\right)$$

(Equation 2.4)

where α is the cutter side rake angle (degrees), θ is the cutter back rake angle (degrees), d_c is the cutter diameter (in), and σ_c is the unconfined compressive strength (psi).

Motahhari's model (Motahhari et al., 2010) was PDC-bit-specific, and incorporated a wear function:

$ROP = W_f\left(\frac{RPM^{\gamma}WOB^{\alpha}}{D_b UCS}\right)$, (Equation 2.5)

where UCS is the unconfined rock strength (psi), W_f is the wear function, G is the model coefficient which represents the drillability, α and γ are ROP related model exponents.

The Bourgoyne and Young (BY) model (Bourgoyne Jr & Young Jr, 1974) estimates ROP as a function of eight parameters as shown in Equation 2.6.

$\frac{dD}{dt} = Exp(a_1 + \sum_{j=2}^{8} a_j x_j)$ (Equation 2.6)

where D is the well depth (ft), t is time (hr), a_1 is the formation strength parameter, a_2 is the normal compaction trend exponent, a_3 is the undercompaction exponent, a_4 is the pressure differential exponent, a_5 is the bit weight exponent, a_6 is the rotary speed exponent, a_7 is the tooth wear exponent, and a_8 is the hydraulic exponent. Coefficients a_1

through a_8 are determined with a multiple regression technique, using several data points to determine the eight unknowns that best fit a specific set of field data.

The models covered in this section have been derived based on laboratory and field based experiments incorporating the different physical phenomena during drilling. They all contain some empirical constants which are dependent on the formation being drilled. These empirical constants are calculated using the training set as outlined in Figure 1, conditioning each model to the data measured in a specific formation. A complete analysis of constant bounds and calculation of ROP has been covered by in detail by Soares (Soares, Daigle, & Gray, 2016). The traditional model introduced in this section seem like data-driven themselves – they attempt to fit an analytical or smooth convex function with some empirical parameters (which are determined using the training set). They seem like the data-driven model created decades ago. However, they work well to integrate the physics such as the bit wear function.

2.3.2 Data-driven Models

The previous section analyzed models which were derived using the physics of the borehole. Their main limitation is the constraint placed on equation form. Looking at Equation 2.2 – 2.6 it is evident that only power-law models were used to model ROP. While this analysis may work for certain formations, it has a possibility of producing less accurate results. The underfitting of the model indicates that some input features which affect the ROP have been left out or incorrectly modeled. A more flexible model would circumvent this issue – provide the ability to model ROP based on the data collected for a specific formation. This leads to the use data-driven models, which use the training data to determine the relationship between the input parameters for ROP prediction. Pairwise correlation of drilling data is plotted in Figure 2.3. Input parameters correlated to ROP such as Depth, RPM and flow-rate can be used for fitting a ROP model. However, the incorporation of domain knowledge or feature engineering is used to ensure that the model is built in a robust fashion. Figure 2.5 illustrates the process of building a data-driven model. A model is built as a function of input feature (or parameters) to determine or predict the response *Y* (ROP). The input features chosen in this paper are easily measured on the surface, which makes them inexpensive to obtain. Some of these input parameters can be changed easily on the drilling-rig to improve or maximize ROP (Chapter 4). Data-driven models are fit using a fitting algorithm – commonly referred to as a machine learning algorithm. Some commonly used fitting algorithms are explained.

2.3.2.1 Linear Regression

Linear regression is the simplest machine learning algorithm which can be used to fit data. It makes an important assumption – data are linearly related. ROP is modeled as a linear function of the input features (Equation 2.7). This method is simple, easy and efficient in cases of linear data:

$\text{ROP} = \sum_{n=1}^{N} a_n x_n$, (Equation 2.7)

where x_n are feature vectors, and a_n are constants determined based on the formation being drilled. The constants a_n can be determined by minimizing the least squares or " l_2" loss (defined as: $\left(ROP_{predicted} - ROP_{actual}\right)^2$).

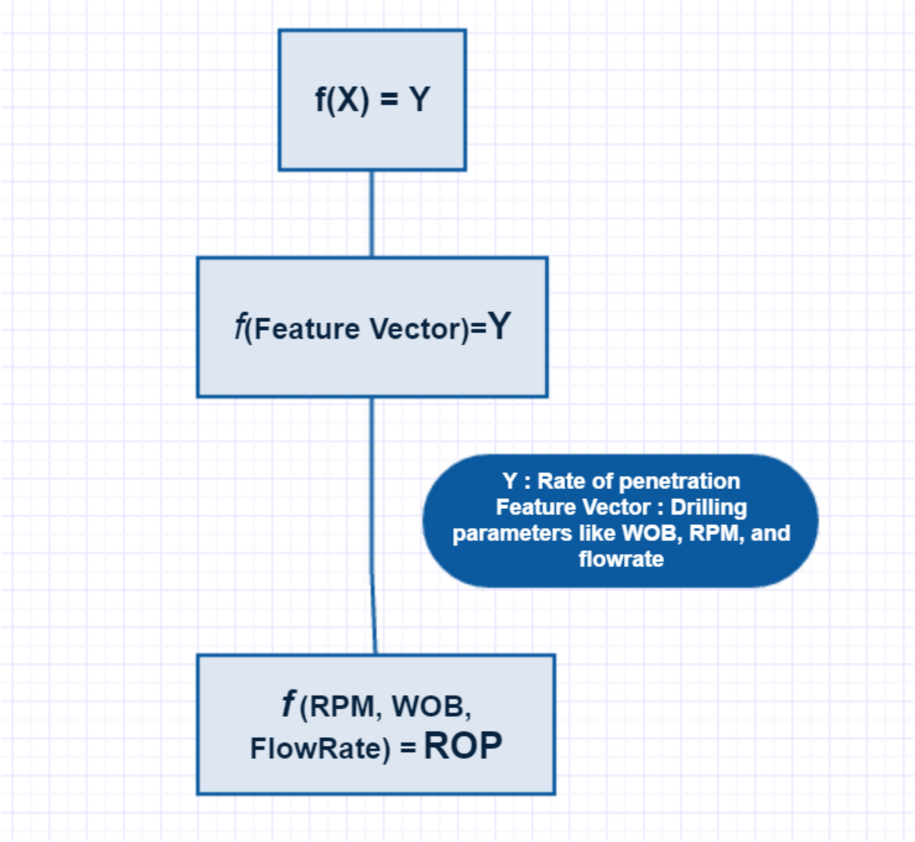
f(X) = Y
f(Feature Vector)=Y
Y : Rate of penetration
Feature Vector : Drilling parameters like WOB, RPM, and flowrate
f (RPM, WOB, FlowRate) = ROP

Figure 2.5: Schematic showing the process of building a data-driven model. The response variable (Y) is fitted as a function of several input parameters – RPM, WOB, Flow-rate, and UCS – to model ROP. Any suitable fitting algorithm can be used to model ROP as a function of these input vectors

2.3.2.2 K Nearest Neighbors (KNN)

The KNN is a non-parametric method which does not assume any functional form or make linearity assumptions. For a specified integral value of K and a prediction point x_0, the algorithm searches for K closest neighbors (N_o). The estimation is the average of all the training data in N_o. This model will perform poorly in high dimensional spaces (with the increase of input parameters or dimensions, the ability of the algorithm to find nearest neighbors decreases). This algorithm can be used to predict ROP independently as a standalone data-driven model or by combining different deterministic model predictions using the hybrid-One algorithm.

$$\mathrm{ROP} = \frac{1}{K}\sum_{n=1}^{K} ROP_{N_0}, \qquad \text{(Equation 2.8)}$$

2.3.2.3 Trees

Trees represent a simple non-linear method to predict data. A simple tree has been built to predict ROP (Figure 2.6). It consists of a series of splitting rules. The first split assigns the observation having flow rate > 374 to the right branch. The next parameter evaluated is UCS: if it's less than 5428 psi then a ROP prediction of 110 ft/hr is returned. On the other hand, if the flow rate was less than 374 and RPM is less than 63 then a prediction of 53 ft/hr would be returned. Overall the tree stratifies data into different segments based on input features.

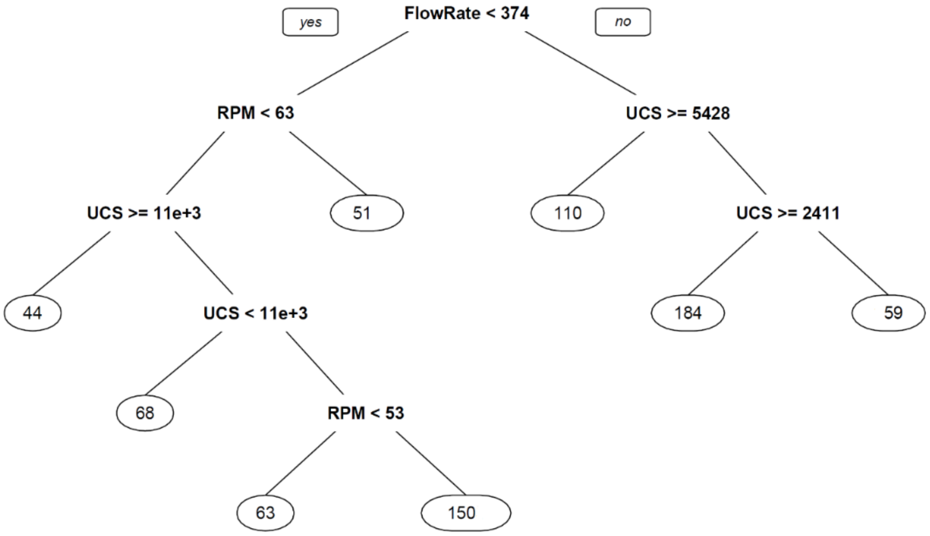

Figure 2.6: Schematic of a tree used to predict ROP. The input variables are branched off (or split) based on a greedy approach to minimize the l_2 error of the result.

The stratification of input features takes place by splitting the input feature space using an algorithm. The tree itself is built using a greedy algorithm (building the best possible tree would result in a combinatorial explosion (Friedman, Hastie, & Tibshirani, 2001)). The sample predictor space of 'N' dimensions is divided into *m* distinct non-overlapping regions (R_m). For each observation that falls into a region R_i the mean value of the training values in R_i is the predicted ROP. The goal is to find regions (R_i) which minimize the sum of squared errors of the entire training set. Trees are typically grown until a stopping criterion is reached: a minimum of three observations in each region. For simplicity, consider a case where ROP is predicted only using RPM and WOB as input features. A simplified two-dimensional feature space is split using a simple (greedy) algorithm is plotted in Figure 2.7. If WOB is less than 2500 lbs: the average value of scatter that resides within the region "R_1" is returned as the ROP prediction. Similarly, if WOB is greater than 2500 lbs and RPM is less than 60 – ROP prediction is the average of the scatter enclosed in "R_3".

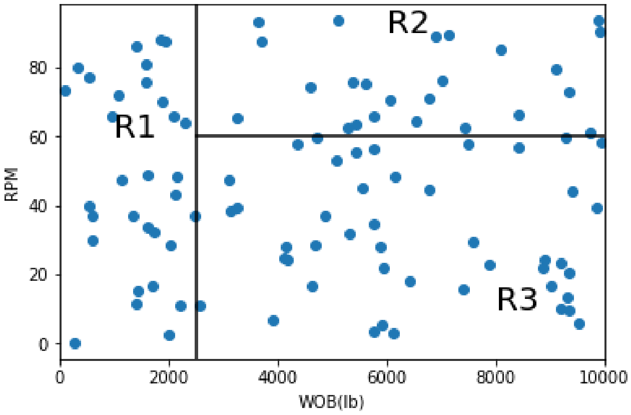

Figure 2.7: Three region partition for ROP modeling using RPM and WOB as input predictors. The regions are calculated using the tree building algorithm are labeled as R_1, R_2 and R_3. If WOB is less than 2500 lbs – ROP prediction of the average of the scatter within R_1 is returned as a result.

Although trees by design are used to predict nonlinear data, they can severely overfit. One way to avoid this is the use of pruning in trees where long trees are grown and then shortened by "pruning" (James et al., 2013). However, trees are the building blocks of ensemble algorithms like bagging, boosting, and random forests which are efficient predictors and prevent overfitting. A three-dimensional view of a sample feature space is shown in Figure 2.8.

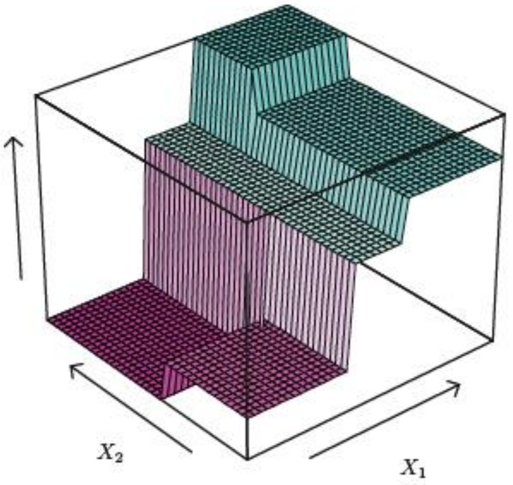

Figure 2.8: Three-dimensional view of sample space of two features X_1 and X_2 predicting a response function (James et al., 2013)

2.3.2.4 Bagging

The accuracy of trees can be improved by using it within an ensemble algorithm. Trees suffer from high variance – when built on two separate halves of the training dataset they are likely to be yield different results. An algorithm with low variance (like linear regression) would not have this problem. The variance of samples (X_1, X_2, ...X_n) can be reduced by averaging them since the variance of the mean ($\bar{X}$) of these samples is $\frac{\sigma 2}{n}$. In this setting, multiple trees can be averaged to reduce their overall variance, preventing overfitting. These trees can be built on pseudo independent data sets carved from the original dataset using bootstrapping (Efron, 1987). In summary, many trees are grown on each bootstrapped training sample (which emulates a new dataset) and averaged to reduce the overall variance. This powerful algorithm described using trees can be generalized and applied to any other machine learning algorithm to help reduce overfitting. Each ROP predictor *f_1,f_2,..* are calculated using B separate training sets and averaged to obtain a single low-variance model as shown in Equation 2.9.

$\hat{f}_{avg}(\mathrm{x}) \ = \frac{1}{B}\sum_{b=1}^{B}\widehat{f^{b}}(x)$, (Equation 2.9)

2.3.2.5 Random Forests

Bagging improves the results of the tree algorithm. However, with the inclusion of a decorrelation of input features, it is possible to improve accuracy even further. The random forests (Breiman, 2001) algorithm is an extension of the bagging algorithm which applies a condition to decorrelates trees. This decorrelation is performed by selecting a subset of input features at each split of building a tree. Rather than considering all available input drilling features (ROP, WOB, UCS, flow-rate, etc.), only a subset of input features – selected using cross-validation – are used to build a tree. The random forest also allows a feature ranking chart which can be used to determine the importance of input features used in a model. This feature importance chart can act as a proxy for model inference – a high importance placed on WOB would indicate that the change in ROP is most sensitive to WOB (example plotted in Figure 2.9). However, it is important to note that this analysis represents the results from a greedy algorithm which may not always be indicative of the true situation; additionally, no information can be inferred about the direction of influence of each input feature.

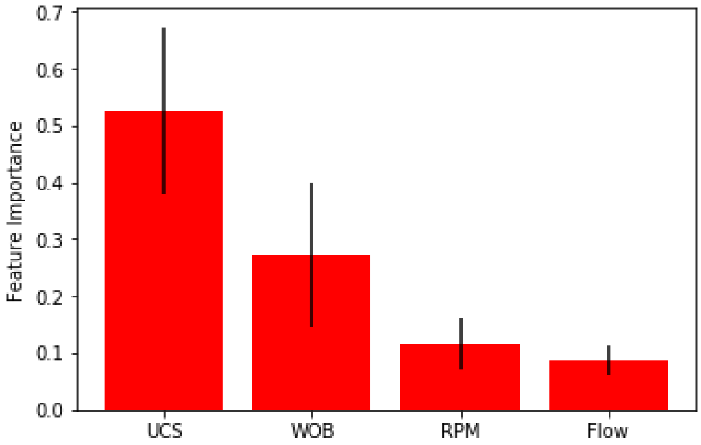

Figure 2.9: Random forest feature importance ranking. The feature importance of the model shown in this figure ranks rock strength followed by WOB as the most important parameters affecting ROP

2.3.2.6 Neural Networks and Deep Learning

Neural networks are universal function approximators – they can be used to approximate any function (Ng, 2000). They were commonly referred to as artificial neural networks (ANN) since they mimicked the way the brain is hypothesized to work. However, in recent times (with sufficient rebranding) they are often referred to as neural nets, deep nets or neural networks. A neural network builds a function approximator with the use of neurons and layers (Figure 2.10). A deep neural net is a neural network which has multiple layers – commonly used for approximation of high dimensional functions. Neural networks commonly require massive amounts of data to train an accurate model; they work really well for applications where the input features are high dimensional such as images and videos (> 10,000 dimensions). For low-dimensional problems (<500 dimensions), traditional statistical learning methods such as linear regression, logistic regression, random forests, and bagging generally outperform neural networks.

Deep neural nets (also called forward feed nets, artificial neural nets, or multi-layer perceptrons) are the quintessential deep learning models. Their main goal is to estimate an unknown function (Y) in terms of some input parameters (X_i). The name feedforward implies that information only flows forward in the model as compared to the recurrent models (Schmidhuber, 2015). Feedforward neural networks are called networks because they are represented by composing together many different functions. The model is associated with a directed acyclic graph describing how the functions are composed together. For example, three functions f^1, f^2, f^3 are connected in a chain to form the function: $f = f^3(f^2\ (f^1(x)))$. In this case, these chained functions are represented as layers for easy visualization; f^1 here represents the first layer of the neural network. Neural networks, in general, have three sets of layers: the input layer, the middle (hidden) layers, and the output layer. The input layer consists of inputs features which are used to learn the unknown function (or response). The output layer consists of the output or response. The hidden layers consist of several non-linear transformation functions multiplied with a weight (determined using a learning algorithm), which "warp" the input high dimensional data into a "simpler" lower dimensional space so that a lower order model can be fitted. It is common practice to use stochastic gradient descent as the learning algorithm (Rumelhart, Hinton, & Williams, 1988) and rectified linear units for the non-linearity function (Krizhevsky et al., 2012). A simplified version of a feed-forward neural network has been shown in Figure 2.10.

The design of a deep learning network is not very different from fitting a function (such as linear regression). The estimation of this function is modeled as an optimization problem: it consists of an objective function, learning algorithm, and a model family. The objective function should represent how well the model estimated the output. Negative log likelihood (Goodfellow, Bengio, & Courville, 2016) is commonly used for classification in machine learning and "*l_2 loss*" for regression. In the case of neural networks, the cost function is nonconvex – unlike linear regression – and cannot be solved using linear solvers. Instead, an iterative gradient based approach is used to drive the cost as low as possible using an algorithm called gradient descent. Feedforward neural networks with multiple hidden layers are often called deep neural networks. The discovery of neural networks dates back many decades, however, their adoption for commercial applications has been recent – mostly attributed to the increase in training data and computational power.

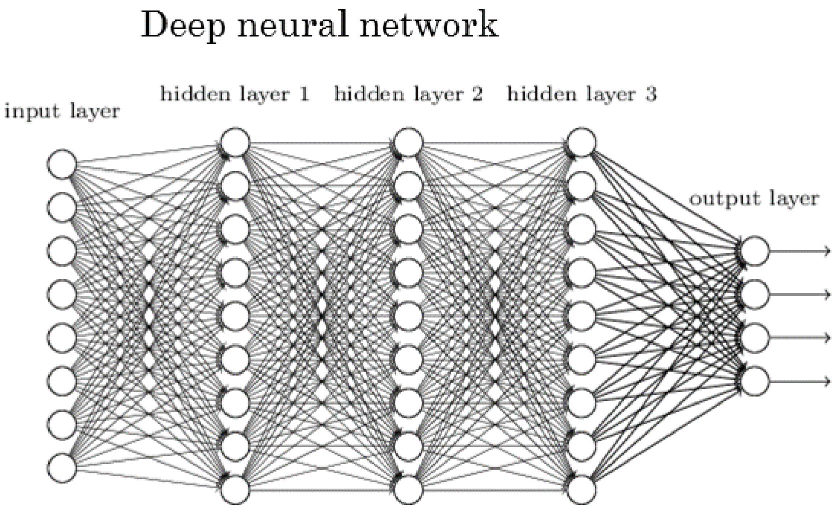

Figure 2.10: Schematic of a neural network with three hidden layers. This qualifies as a deep neural network based on its definition

Deep learning is a part of a broader class of machine learning algorithms which have been applied to various fields such as computer vision, speech recognition, text analysis, machine translation (Goodfellow et al., 2016; C. M. Hegde, Awan, & Wiemers, 2018; LeCun et al., 2015; Schmidhuber, 2015). For drilling optimization applications – data is low-dimensional – and neural networks will often overfit. The dataset used for analysis in this chapter does not have sufficient data to prevent overfitting. As a result, neural networks are not used for ROP modeling since traditional statistical learning methods provide a better solution conforming with the Occam's Razor theorem (Blumer, Ehrenfeucht, Haussler, & Warmuth, 1987).

2.3.2.7 Model Ensembles

An ensemble data-driven model (or ensemble learning ROP model) combines other data-driven algorithms (such as linear regression, bagging, neural nets, random forests, etc. (C. M. Hegde, 2016)) to create an ensemble or combination of data-driven models (Equation 2.10). This method is particularly useful in prediction of highly non-linear data where simple models can be combined for more accurate estimations.

$$\text{ROP} = w_1 DD_1 + w_2 DD_2 + w_3 DD_3, \text{ (Equation 2.10)}$$

where DD_n are data-driven models (random forest, neural network or linear regression), and w_n are constants determined based on the formation being drilled. 'w_n' can be determined by using the stacking algorithms such as the feature-weighted linear stacking (Sill, Takács, Mackey, & Lin, 2009). The constants of the additive models can also be determined using other machine learning algorithms such as linear regression or random forests. In most cases, adding a constraint – the sum of model weights should sum up to 1 – converts the problem into a form (geometric programming) that is easy to solve (Friedman et al., 2001). The premise behind this algorithm is that the deficiencies of one model will be counteracted by other models – a team effort for ROP prediction.

2.4 PARAMETRIC STUDY OF TRAINING SET LENGTHS

The accuracy of data-driven models predominantly relies on the quality, range, and volume of the training data. This section is dedicated to the parametric analysis of training data: its range and volume for efficient ROP prediction. The change in accuracy of ROP prediction based on changing the type and size of the training set relative to the test set is evaluated.

2.4.1 Type of training data

Since the accuracy of data-driven models is largely dependent on the training data, this section evaluates three different types of training data to predict ROP illustrated in Figure 2.11. The first kind of training data (case 1) is the data obtained while drilling the formation in question– formation specific training data. The third kind of training set (Case 3) is the data obtained while drilling preceding formations (or upper levels), which are used to predict ROP in a different formation: for example, using Broom Creek drilling data to predict ROP in Tyler formation (as shown in Figure 2.11 (left) as case 3). Case 3 is a situation which is encountered when the bit enters a new formation, and no prior data for that formation is available. ROP modeling in a new formation should use data of the same formation from pad wells or use physics-based models. ROP modeling during formation transitions can be handled without any modeling or by using a weak Gaussian prior or a prior based on data obtained from drilling similar formations. The second kind of training data (case 2) is a combination of the cases 1 and 3, where the training set contains drilling data from other preceding formations as well as data from the formation in question. This case of training data would answer whether data from the same BHA but a different formation would help in the prediction of ROP.

Training sets can be evaluated based on the accuracy of ROP predictions of a model is built on it. The normalized error for different models is compared to ensure an unbiased comparison. Intuitively one can expect case 1 to be a better training set than case 2 – because case 1 has formation specific drilling data (or relevant data). However, case 2 contains data from other formations as well as the relevant data. This extra data (partially relevant) gets equal preference – by the algorithm – in building the data-driven model, which decreases the accuracy of models built on case 2. Case 2 is expected to be better than case 3 since it has some formation specific relevant data, whereas case 3 has data from other formations. Case 3 represents a situation where a data-driven model is trained on data collected in a different formation – not recommended. One school of thought would be to avoid using this type of data for building models since different geological formations are inherently different, using data from other formation for training a data-driven model defies the fundamental principal of geosciences. However, this case has been evaluated to show that relevant data is extremely important for model accuracy. This also goes to show that manually generated data cannot be used for ROP prediction or if used will result in poor accuracy. Additionally, it provides extra confidence in the model since random data or poor data does not yield as good a result as using pertinent data.

Three training sets are used to build a data-driven model which is evaluated for ROP prediction errors on the same test set. The best training set is determined by comparing their normalized prediction errors (Equation 2.11). Figures 2.12 shows the test set errors for the three different training sets. As expected, case 1 outperforms case 2 and case 3 for ROP prediction.

$$Normalized\ Error\ Rate = \frac{|Predicted\ ROP - Actual\ ROP|}{Actual\ ROP}, \text{(Equation 2.11)}$$

Ratcliffe is the only formation where training data from case 2 and outperforms case 1. This is hypothesized due to the thickness of the formation. This formation has 67 ft of data, which makes it a very thin formation. The sparsity of available data in the formation causes higher error rates in case 1 as compared to cases 2. These results indicate that for thinly bedded formations it is better to include training data from previous formations to augment the dataset.

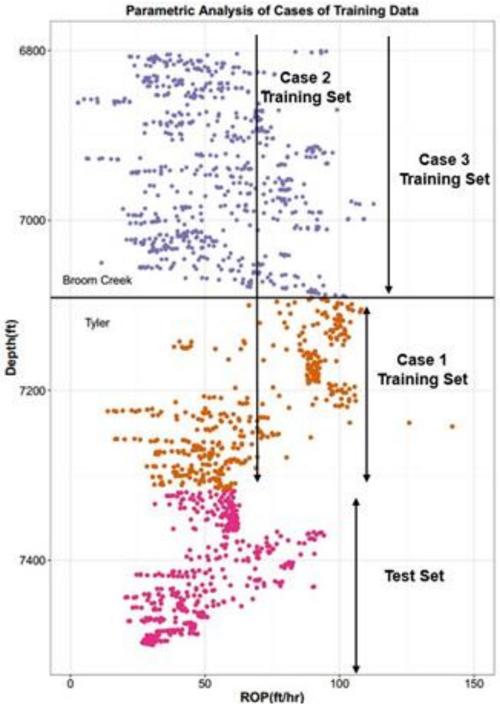

Figure 2.11: Illustration of test and training sets for parametric evaluation of nature of training set. A total of three cases have been illustrated

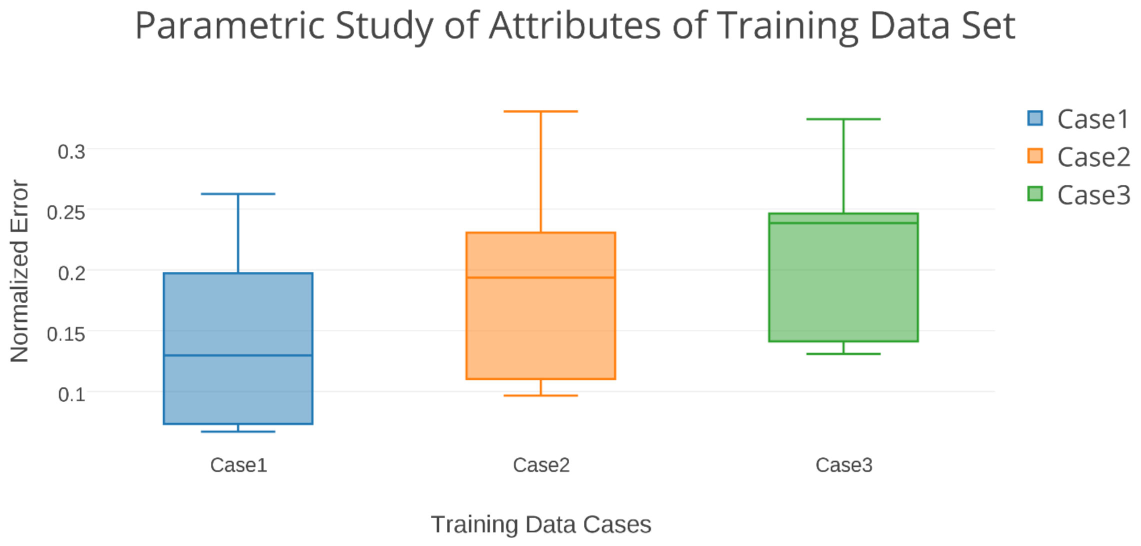

Figure 2.12: Boxplot evaluation of test set errors of models built on three different cases of training sets. As expected, overall case 1 outperforms the other two cases indicating that most accurate models require formation related data

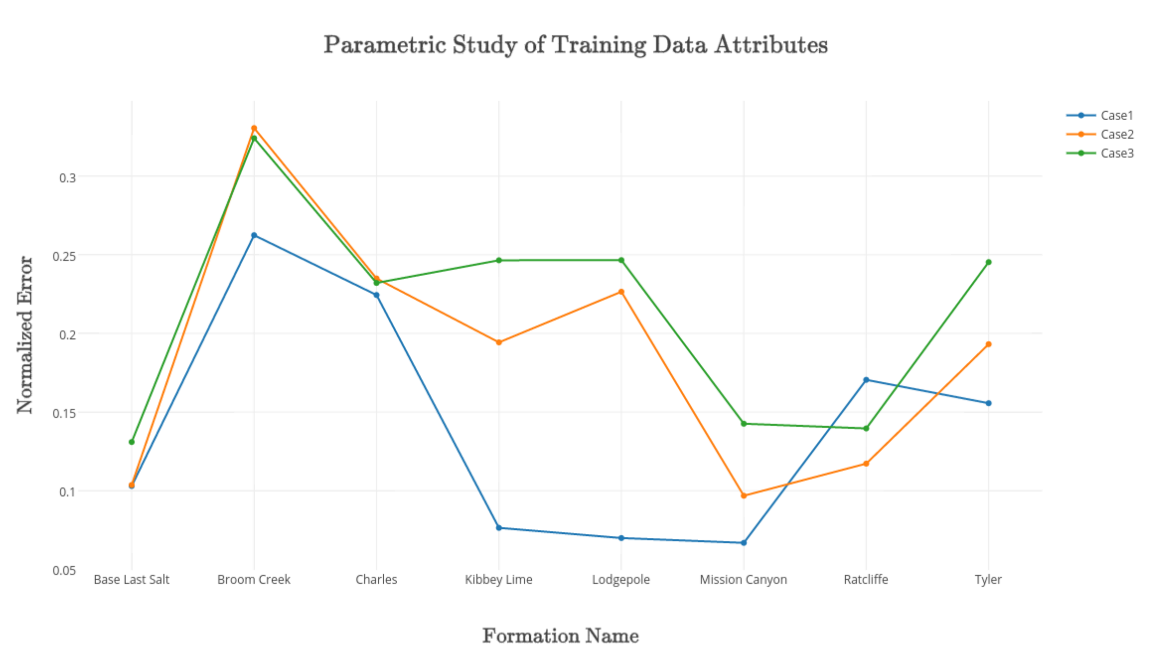

Figure 2.13: Line plot evaluation of test set errors of models built on three different cases of training sets for several different formations. In all cases (except 1) models built on case 1 outperform cases 2 and 3

2.4.2 Training-test ratio in ROP prediction

This section evaluates the relationship between the ratio of training-test sets and model accuracy. The drilling data in each formation is partitioned into training and test sets for ROP prediction (Figure 2.2). Increasing the length of the training set should ideally improve the accuracy of the data-driven model since more data would be available for learning. Data-driven models tend to produce better results with additional data. The optimum size of the training set (one that produces a reasonable test error) depends on the formation and data.

The size of the training set relative to the test set is changed for each formation; the ROP prediction error for each case on the test set is recorded. The training set is changed in size, varying its length from 10% to 90% of the size of formation. The remainder is used as the test set. The average prediction error for each case is compared. Figure 2.14 shows the results obtained from this parametric study. A data-driven model – using the random forest algorithm – is used to predict ROP for each case. Figure 2.14 shows a decrease in error with an increase in the training-test set ratio, indicating that an increase in the length of the training set produces an increase in accuracy (as expected). The accuracy desired (say a normalized error ratio of 0.2) can be computed using the plot in Figure 2.14.

2.4.3 Optimal training sets

The plots in this section 2.4.1 and 2.4.2 provide some insight into practical applications of data-driven models in drilling. Training sets are more reliable and efficient for data-driven models when constrained to the formation of interest (case 1). Optimal training-test set ratios vary depending on required accuracy and formation. If an error rate of 0.2 or 20% is assumed to be required, then a train-test ratio of 0.2 between training and test set length remains sufficient for most formations. A lower error rate requires a larger volume of training set data, pushing the training-test set ratio to 0.3-0.5 in a few cases as seen in Figure 2.14. In some cases (Tyler and Ratcliffe) higher ratios such as 0.7 may be necessary for low error rates of 10%. In one case (Broom Creek) a low error rate <10% is not possible for any ratio of training-test set data evaluated in this analysis using the random forest ROP model. The minimum training to test ratio required (for a data-driven random forest-based model) to outperform a traditional model (conditioned to the entire data set i.e. best-case scenario) are summarized in Table 2.1. The results indicate that in most cases 20-30% of the formation depth is sufficient to outperform the best traditional model for the formation in question. There undoubtedly is a relationship between model accuracy and thickness of formations. For data-driven models, the increase of data results in an increased predictive accuracy.

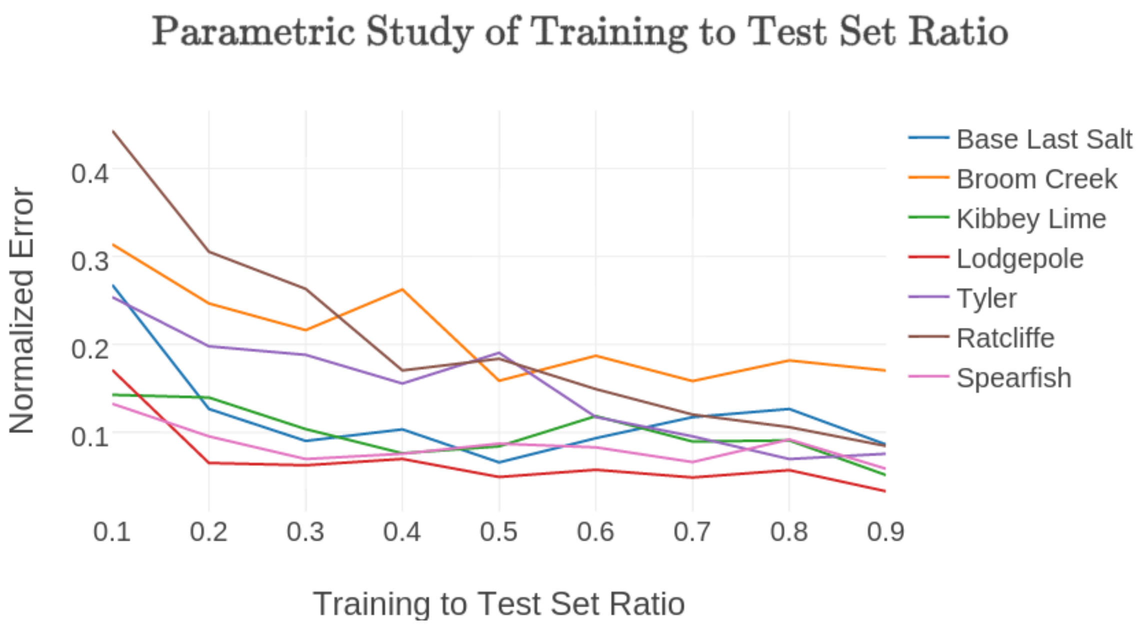

Figure 2.14: Line plot describing the relationship between model error and ratio of training to test data. The increase in the amount of training data shows a decline in model test error.

Table 2.2 shows an analysis of the amount of training data required to achieve a 10% and 20% ROP prediction error for some formations in the data set. For 80% accuracy, the amount of training data required is around 40ft on average except for Broomcreek and Tyler formations which are higher. Broomcreek and Tyler require around 80 points in training data. For 90% accuracy, the amount of training data required is around 65 points on average except for Broomcreek and Tyler. This also shows the importance of model update or retraining as drilling proceeds. The frequency of retaining and the improvement of model accuracy with an increase in training data are currently being researched.

Table 2.1: Parametric analysis to determine the size of the training set to outperform traditional models

Formation Name	Training to Test Ratio required to perform better than the best traditional model	Length of Dataset	Best Traditional Model
Base Last Salt	0.2	211	Bingham
Broom Creek	0.2	196	Hareland
Charles	0.4	348	Motahhari
Kibbey Lime	0.1	220	Hareland
Lodgepole	0.2	348	Hareland
Mission Canyon	0.3	764	Hareland
Ratcliffe	0.4	65	Hareland
Tyler	0.2	432	Hareland

Table 2.2: Amount of training data required for ROP accuracy

Formation Name	Training Points to achieve an accuracy of 0.2	Training Points to achieve an accuracy of 0.1
Base Last Salt	42	63
Broom Creek	78	NA
Kibbey Lime	22	66
Lodgepole	35	70
Ratcliffe	26	59
Tyler	86	302

2.5 ROP Predictions: Results and Discussions

Traditional models were used to model ROP based on field data. Simulations are run individually on each formation since ROP models are formation-dependent. The traditional models were used to predict ROP based on the methodology outlined in literature (Soares et al., 2016). Errors are normalized to ensure unbiased comparison with results from other formations. An ANOVA test and the high test error for linear regression models show that data modeled here is not linear. The prediction error rate has been plotted as a line and box in Figure 2.15. The test-train ratio of 0.5 was used for the experiments evaluated in Figure 2.15. Boxplots summarizing the change of test-train ratios and model accuracy as shown in Figure 2.16. The boxplot plotted here are contain the median of all values displayed with the 25th and 75th quartile as edges of the box. The whiskers extend the box plot to the extreme values. Outliers are plotted as a point outside the box plot. In order for a datum to be an outlier, it must be larger than the 75th percentile by at least 1.5 times the inter quartile range or smaller than the 25th percentile by at least 1.5 times the inter quartile range. The interquartile range is the difference between the 75th and 25th percentile. Alongside the traditional models, the accuracy of the random forest model is plotted for comparison. The random forest hyperparameters were chosen based on cross-validation error (the out of bag error would be a great alternative as well). A total of 1000 bootstrapped trees were averaged each with a feature selection subset of 3 and a minimum sample required at leaf node of 3. Since the minimum samples at the leaf is 3, the minimum data used to make a prediction is 3 points, however, the average is expected to be higher, depending on the data and the node split. The number of bootstrapped trees utilized for this algorithm is 1000, which is high considering the dimensions of the data that are being modeled. It is hypothesized that this is due to the chosen value for the minimum leaf size since the minimum samples at the leaf is small (3), there was a chance for each tree to over fit. The averaging of trees decreases is the main knob which controls variance in a random forest. Hence a larger number of trees were required to prevent overfitting.

Once the hyperparameters have been determined, the random forest model is trained in minimal time (180 ms for training a random forest model with 1000 trees) 8 GB DDR3 RAM laptop with Intel 7thgeneration i7 7500U CPU @ 2.60 GHz processor running python's scipy package (Jones et al., 2001). However, tuning the model can take longer depending of the number of hyper parameters evaluated. Implementing a random search (Bergstra & Bengio, 2012) instead of a grid search can help improve computation efficiency of this process.

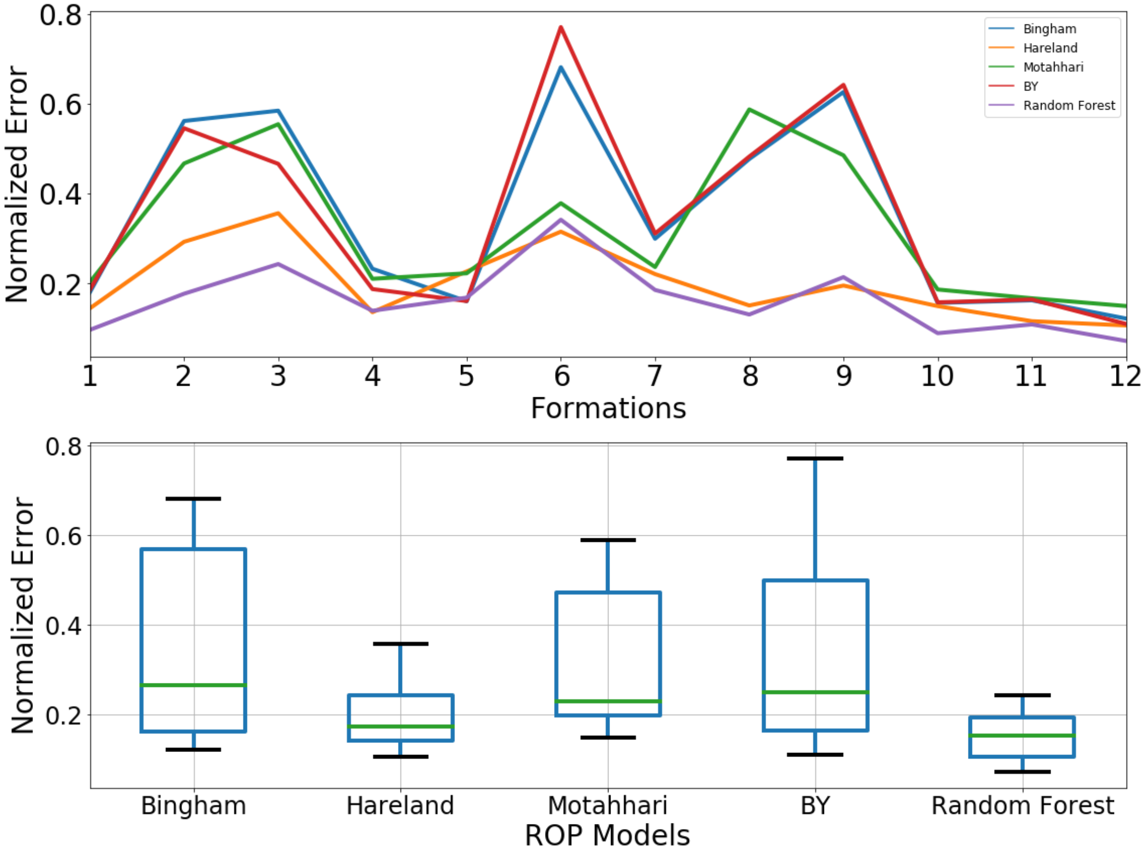

Figure 2.15: Normalized model error of traditional models compared with the random forest model for ROP prediction; A test-train ratio of 0.5 was used to train a model for each formation; (Top) The normalized error has been plotted on the y-axis against the formation; (Bottom) Box plot of the normalized errors for all formations.

Based on figures 2.15 and 2.16 it is evident that the random forest model performs better than traditional models for the prediction of ROP. As the length of the training set is increased the resultant error decreases for the random forest model. However, this does not hold for traditional models; the error decrease with an increase in the train-test ratio is not evident; in the case of traditional models where the functional form cannot be modified – only the empirical constants are changed. The prediction error for traditional models may also decrease as more data is used for training in certain cases since using more data "generalizes" the empirical constants. The empirical constants if determined at shorter intervals will lead to higher accuracy – similar to a local regression analogy (S. P. Wallace et al., 2015). The flexibility of this functional form enables the random forest model to slowly "learn".

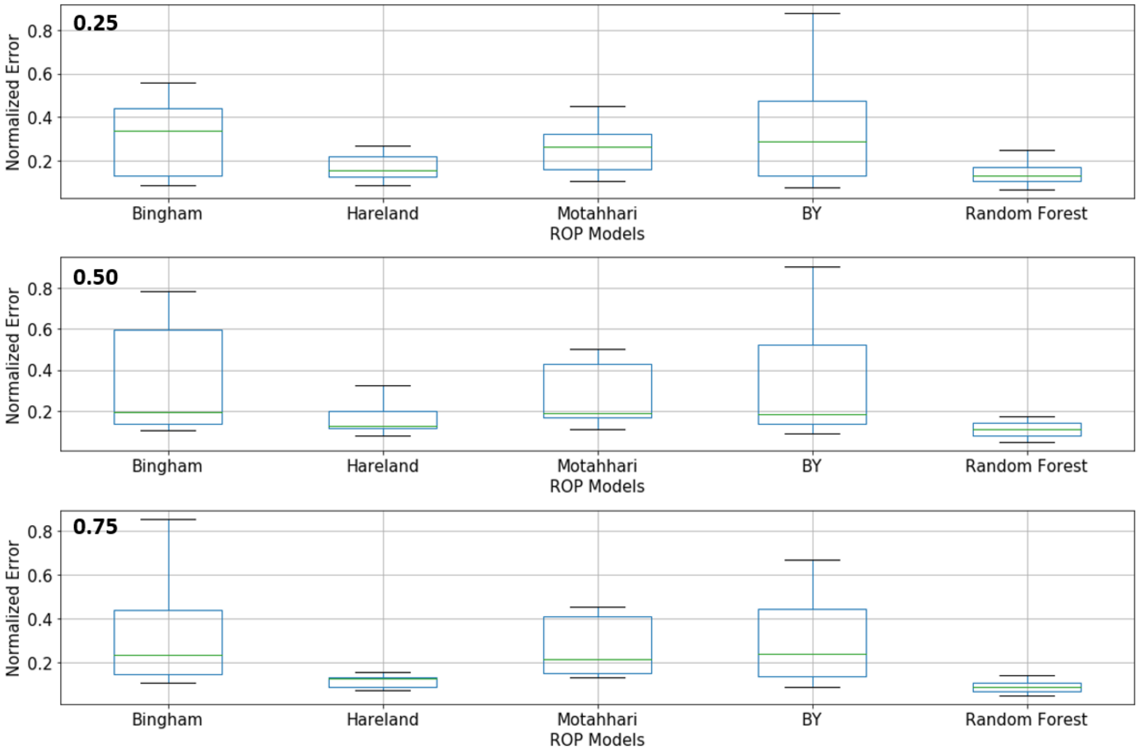

Figure 2.16: Model error with different train-test ratios (displayed at the top left corner of each image); a test train ratio is the ratio of the length of the training set to the test set; As the ratio increases the random forest model error decreases as expected. Traditional models sometimes result in a smaller error with more training data – the effect is not as pronounced as the random forest model

Several machine learning algorithms are used to model ROP based on field data. Simulations are run individually on each formation since ROP models are formation-dependent. The traditional models followed the train-test split method as discussed in section 2.2. Test errors are normalized to ensure unbiased comparison with results from other formations. The prediction error rate has been plotted as a line and box plot for data evaluated with a test-train ratio of 0.5 in Figure 2.17. The data are split into training and test set spatially – the first half of the formation is used for training and the second half to evaluate the model. Boxplots summarizing the change of test-train ratios and model accuracy as shown in Figure 2.18. The figures 2.17 and 2.18 indicate that for this dataset, random forests outperform other algorithms for ROP prediction. In specific formations, KNN or linear regression may perform better than random forests. It might be better to approach ROP prediction using a bag-of-models approach: where the best models based on cross-validation error for a given formation is used for prediction. Hyperparameters for each machine learning algorithm evaluated in this plot have been chosen based on their cross-validation error. The KNN algorithm was implemented with K=4, bagging with a total of 1000 bootstrapped trees, and linear regression keeping only statistically significant predictors with a p-value lower than 0.05. The neural networks were implemented using a learning rate of 0.001, two layers with 8 and 4 neurons each. Hyperparameter search for the neural networks included modifying the learning rate, number of layers, early stopping, number of units for each layer, loss function, and optimization algorithm using a randomized search grid (Bergstra & Bengio, 2012).

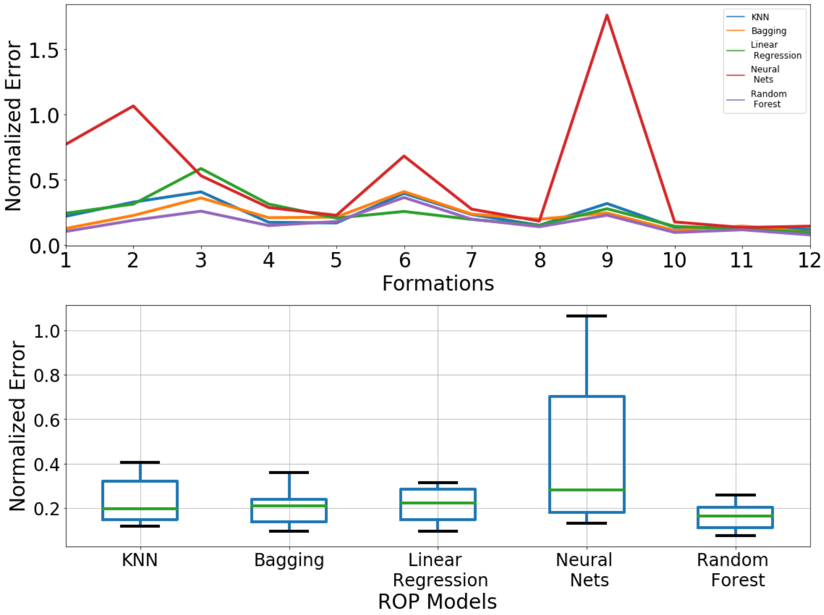

Figure 2.17: Normalized model error of data-driven models compared for ROP prediction; A test-train ratio of 0.5 was used to train a model for each formation; (Top) The normalized error has been plotted on the y-axis against the formation; (Bottom) Box plot of the normalized errors for all formations.

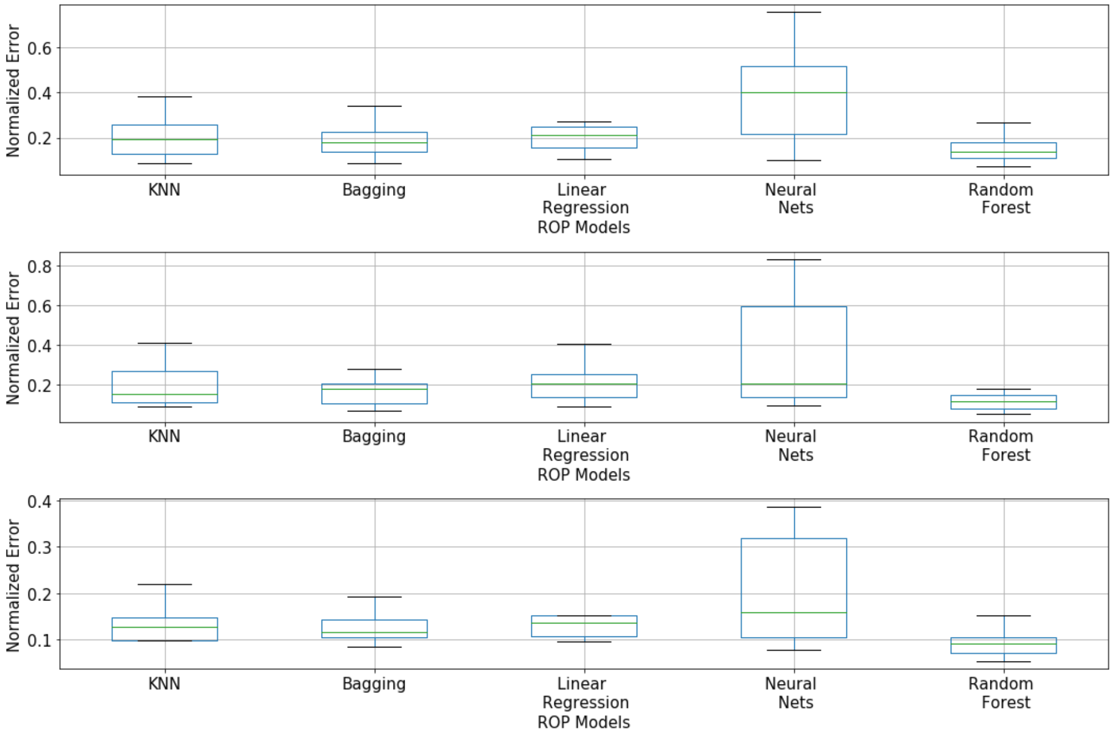

Figure 2.18: Data-driven model error with different train-test ratios (displayed at the top left corner of each image); a test train ratio is the ratio of the length of the training set to the test set; As the ratio increases (from 0.25 to 0.75 top to bottom) the data-driven models show a decrease in error as expected.

Figure 2.19 shows ROP prediction using the Bingham model and a random forest model as a means to visualize predictions and understand R^2. The R^2 is a metric which is used to determine the amount of variance explained by the model. The test data (blue), training data (orange) and ROP predictions (green) in Figure 2.19 have been colored to make a clear distinction. The machine learning model (random forests) had an R^2 of 0.97, whereas the Bingham ROP model produced an R^2 of 0.64. The ROP predictions made using the Bingham model appear to be the best fit line, not following the ROP trends seen in the data.

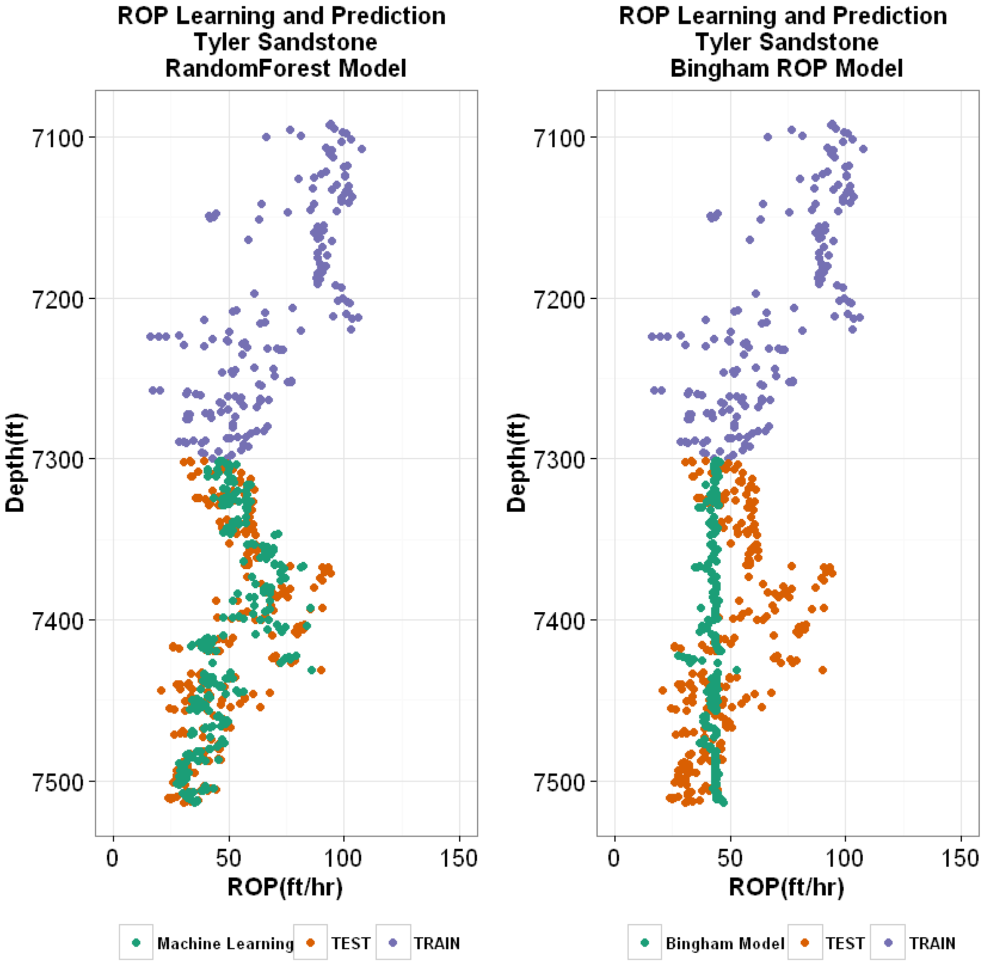

Figure 2.19: ROP predictions using random forest and the Bingham model. While the random forest is able to capture the trend of the data to predict ROP, the Bingham model produces a best fit line

Table 2.3 summarizes the results in a compact form. As previously described the random forest ROP model can be used for inference. The most important features for ROP prediction or modeling can be useful to the engineer, as it can dictate the parameters which control the ROP in that formation. Figure 2.20 provides a measure of the importance of each drilling input variable influencing ROP for a model built on Lodgepole limestone formation using the random forests algorithm. The feature importance depends on the model; hence it will change depending on the formation in question. The importance plot (Figure 2.20 (right) suggests that changing the WOB and flow-rate will change the ROP more significantly as compared to RPM for lodgepole limestone. The black vertical lines show the standard deviation of this importance metric – due to the random nature of the algorithm.

Table 2.3: Summary of ROP simulation results

Predictor	R^2	Normalized Error
Random Forest	0.84	0.14
KNN	0.72	0.19
Linear Regression	0.12	0.16
Bingham	0.46	0.34
Motahhari	0.63	0.27
Hareland	0.19	0.16
BY	0.34	0.33

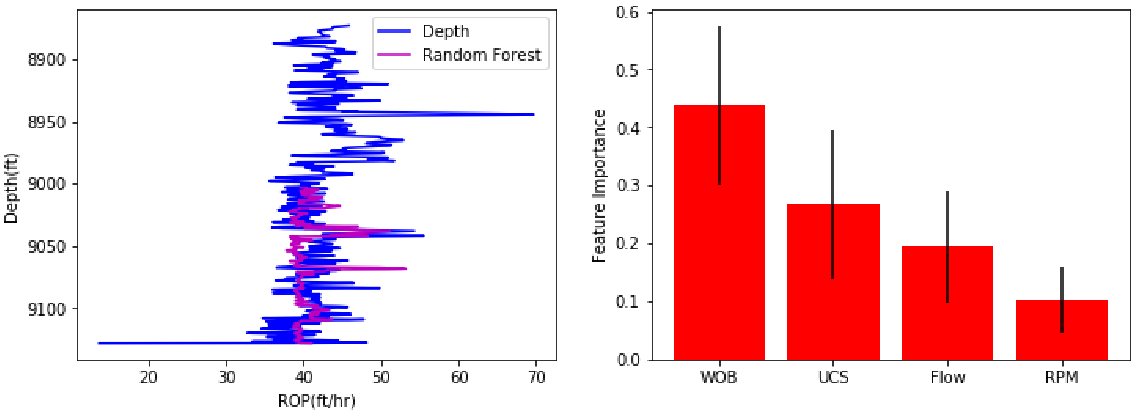

Figure 2.20: (Left) ROP prediction using the random forest ROP model; the first half of the formation is used to train the model; (Right) Feature importance of the random forest model used to model ROP in the lodgepole limestone formation; The black lines indicate the standard deviation of the input features.

2.6 Torque on bit (TOB) Models

Prediction of downhole torque during drilling can help mitigate drilling problems as well as aid in drilling optimization by improving the mechanical specific energy (MSE) of drilling. Surface torque is commonly measured on the rig and typically easy to obtain. Unfortunately, it is still rare to obtain torque on bit (TOB) measurements close to the bit, which are more important since it plays an integral role in the MSE calculations. Torque is measured on the rig surface using a sensor, most of torque and drag modeling is focused on predicting the loss of torque from the surface to the bit – estimating the torque on bit (TOB). The basic concept of TOB prediction or calculation has not changed over the past few decades. However, with increases in computational resources, there has been a surge of numerical and mathematical modeling of TOB. Models have built upon the basic model introduced in the past (Johancsik, Friesen, & Dawson, 1984). However, with increasing complexity in drilling and introduction of horizontal and extended reach wells, these fundamental TOB modeling techniques often fall short. TOB measurements are usually available post bit-run as instrumented subs are typically memory only tools, except with wired-pipe technology (Trichel et al., 2016). Downhole torque and drag modeling provide one method to estimate the TOB without downhole measurement. However, predictions in real-time require an analytical model with a closed form solution to cope with computational constraints (Gerbaud, Menand, & Sellami, 2006). Recently the pioneering work by Ertas (D. Ertas, Bailey, Wang, & Pastusek, 2014) has shown that TOB estimation is possible using surface torque by applying the transfer matrices technique. This estimation of TOB was then used for vibration control. Authors (C. Hegde, Wallace, & Gray, 2015) have used statistical learning methods to predict downhole torque using surface drilling parameters. They argued that analytical models were inaccurate and FEMbased models are not feasible in real-time. A simple solution was to use a pattern matching or machine learning algorithm to learn the torque response downhole.

Alternatively, TOB can be calculated from surface torque by measuring the difference between the torque on and off bottom during drilling: a crude estimate which assumes no contact between the drill string and borehole. As shown by Menand (S Menand & Mills, 2017), the differential pressure can be used to estimate TOB as shown in equation 2.11.

$TOB_{\Delta P\ Motor} = \frac{T_{max}}{P_{max}} * \Delta P$, (Equation 2.11)

where, ΔP is the differential pressure, T_{max} is the mud motor max related torque (ft-lbs), and P_{max} is the mud motor max related ΔP. In this chapter, TOB has been modeled as a function of drilling input parameters – RPM, WOB, flow-rate, and UCS – to measure the change in TOB with changing downhole conditions. This model will be integral to the prediction and optimization of MSE as discussed in later chapters. Modeling torque as a function of control parameters (using a data-driven approach) allows easy coupling of the ROP and TOB (Pavone & Desplans, 1994). The TOB model is fitted as a function of WOB, RPM, flow-rate and UCS using the random forests algorithm. By fitting the TOB model with the same input parameters as the ROP model it is possible to measure the change of ROP and TOB jointly (or in a coupled manner) when drilling control parameters are changed. Alternatively, a full-scale physics-based model of torque on bit (TOB) (Stephane Menand et al., 2006) can be used to model the TOB – this will not result in a coupled model.

2.7 TOB Predictions: Results and Discussions

TOB has been modeled using machine learning algorithms presented earlier in this chapter. Analytical and finite element based TOB models have not been discussed since they deviate from the main use of these models – coupling with the ROP model and TOB prediction for MSE optimization. Several machine learning algorithms were used to model TOB based on field data. Simulations are run individually on each formation since TOB models are formation-dependent. The traditional models followed the train-test split method as discussed in section 2.2. Test errors are normalized to ensure unbiased comparison with results from other formations. The prediction error rate has been plotted as a line and box in Figure 2.21. The test-train ratio of 0.5 was used for the experiments evaluated in Figure 2.21. The hyperparameters for each algorithm were chosen based on their cross-validation error. The best random forest model was built using 1000 trees, a random feature subset of 3 and a minimum leaf node size of 5. The hyper parameters search for other parameters were conducted using a random search implemented in scikit learn(Pedregosa et al., 2011). Boxplots summarizing the change of test-train ratios and model accuracy as shown in Figure 2.22. The figures 2.21 and 2.22 indicate that for this dataset, random forests consistently perform well for predictions. Since random forests perform well for both ROP and TOB prediction and use the same input feature for prediction, they can be built together using input data. TOB feature importance can be calculated paralelling the calculation performed for the random forest ROP prediction model. The importance plot (Figure 2.23 (right)) suggests that changing the WOB is the most significant input feature which controls the TOB for lodgepole limestone. The black vertical lines show the standard deviation of this importance metric – due to the random nature of the TOB prediction model.

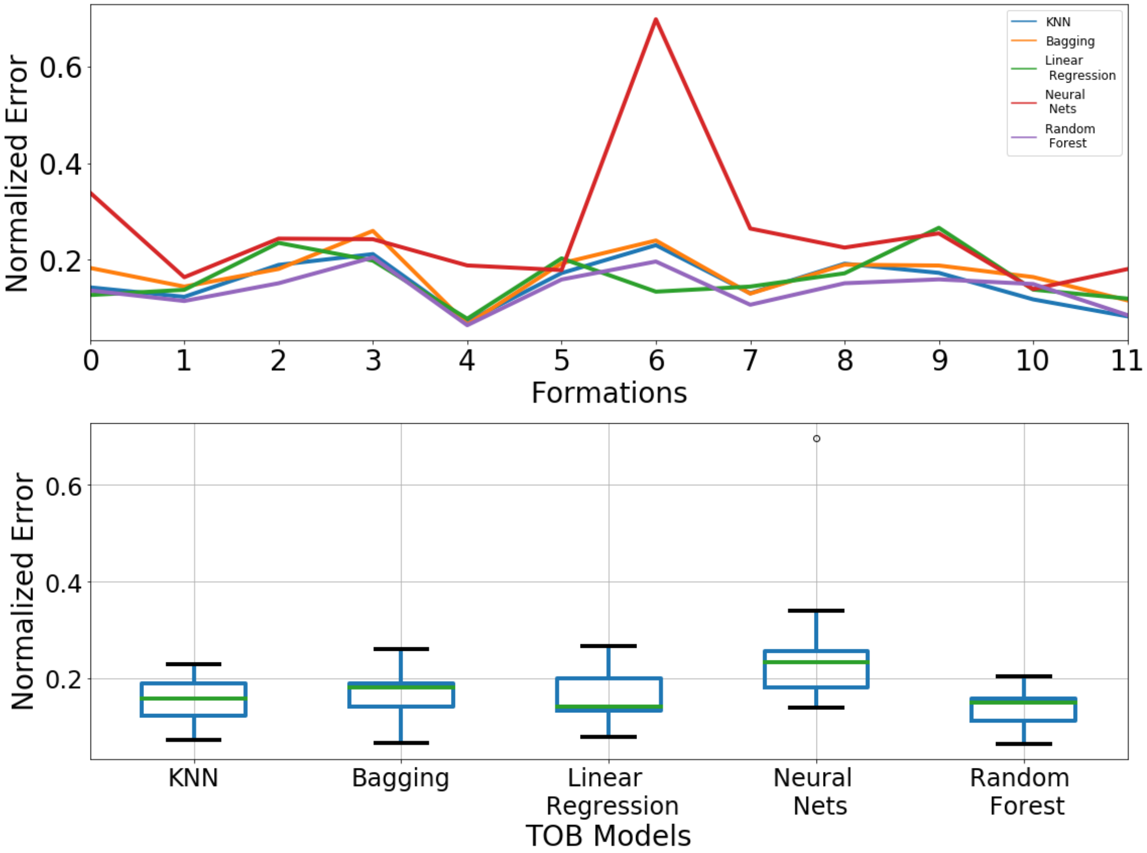

Figure 2.21: Normalized model error of traditional models compared with the random forest model for TOB prediction; A test-train ratio of 0.5 was used to train a model for each formation; (Top) The normalized error has been plotted on the y-axis against the formation; (Bottom) Box plot of the normalized errors for all formations.

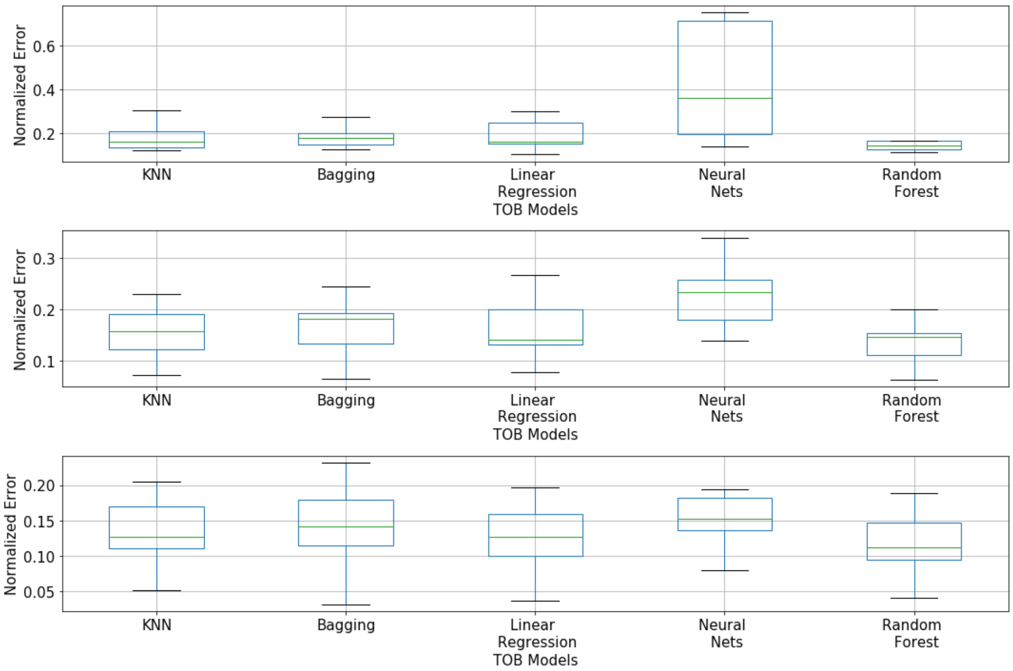

Figure 2.22: Model error with different train-test ratios (displayed at the top left corner of each image); a test train ratio is the ratio of the length of the training set to the test set; As the ratio increases the model error for TOB decreases as expected.

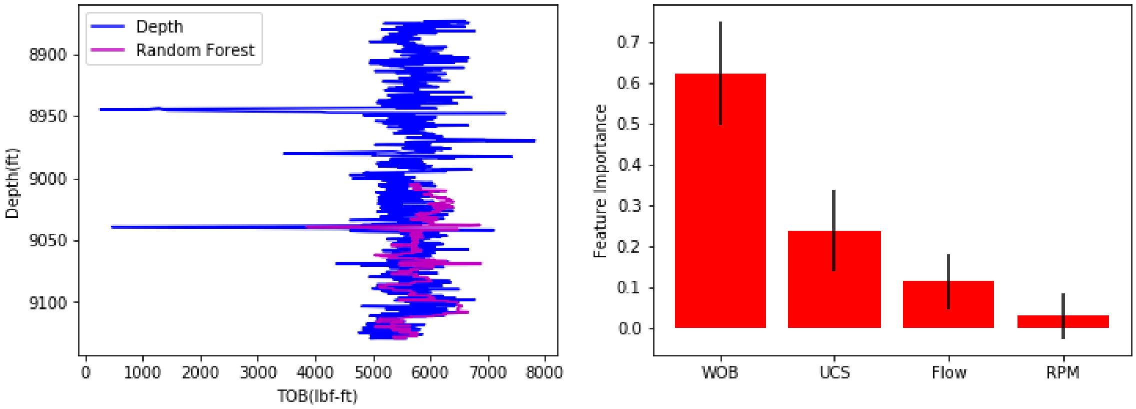

Figure 2.23: (Left) TOB prediction using the random forest model; the first half of the formation is used to train the model; (Right) Feature importance of the random forest model used to model TOB in the lodgepole limestone formation; The black lines indicate the standard deviation of the input features.

2.8 Conclusions

This chapter introduced modeling of important drilling optimization metrics such as ROP and TOB. Traditional models are most often the industry standard for ROP prediction. Although sound in physics, these models introduce many empirical coefficients which restrict their functional form. Empirical coefficients are determined by conditioning the models to the data – the training set. Data-driven models, on the other hand, rely purely on the data to build the model, but the inclusion domain expertise improves the quality of the model. The data-driven models described in this chapter used input features – WOB, RPM, flow rate, and UCS – derived from the physics-based models for ROP prediction. Traditional and data-driven models were used to predict ROP on the test set. Results showed that some data-driven models outperformed traditional models. The random forest algorithm performed the best among the data-driven models resulting in an average 12% error on the test set and an R^2 of 0.84. The random forest algorithm also improved prediction accuracy as more data was used to train the model whereas the traditional models did not. In some formations, ROP was better predicted by other machine learning algorithms and even by traditional models, however, the best overall predictor of ROP is the random forest algorithm. A bag-of-models approach can be used where the best model is utilized for ROP prediction ahead of the bit for a given formation based on cross-validation error on the training set.

Data-driven models were used to model the TOB during drilling using WOB, RPM, flow-rate and UCS as input features. This modeling scheme allows joint modeling of ROP and TOB using the same input features – coupling them. Of all the algorithms, the random forests model outperformed all others evaluated with an average error of 16% on the test set. This error can be reduced with the use of more data for prediction.

Chapter 3: Rate of penetration (ROP) modeling using hybrid models: deterministic and machine learning[3]

The previous chapter discussed modeling of ROP using deterministic and data-driven models. Deterministic models impose constraints on functional form but provide a means to interpret the model whereas the data-driven models were unconstrained in form – leading to an increase in prediction accuracy at the cost of interpretability. There is a middle ground between the two models, where the perks of both models can be combined resulting in a model which provides trade-offs between accuracy and inference – the hybrid model.

3.1 INTRODUCTION

Data-driven modeling techniques provide high-performance, reproducible, and scalable solutions. Their unknown functional form – is it exponential, polynomial, etc.? – makes their interpretation difficult. They use drilling parameters – RPM, WOB, flowrate, and UCS – as inputs, while not constraining functional form. This allows them to model the data more closely, resulting in higher accuracy. This increase in accuracy is a result of sacrificing model inference or interpretability. Knowing the functional form of a ROP model can be used to derive insights (the most influential input drilling control parameter). Hence, finding a middle ground between these two models can result in a good trade-off between accuracy and interpretation (or model inference). The goal of this chapter is to evaluate the performance of hybrid models – a combination of deterministic and machine learning models – in modeling rate of penetration (ROP) of drilling. This modeling

[3] Hegde, C., Soares, C., & Gray, K. (2018). Rate of penetration (ROP) modeling using hybrid models: deterministic and machine learning. *Unconventional Resources Technology Conference*. Houston, Texas, USA: Unconventional Resources Technology Conference

The author of this thesis is the primary author of the paper

methodology has been carried out with success in climate science (Goldstein, Coco, Murray, & Green, 2014), where, combining the strengths of inductive (data-driven) and deductive (physics-based) approaches into a hybrid model was successful.

This chapter analyzes different ROP modeling strategies. Bingham, BY, and Motahhari models constitute the applied deterministic ROP models. The best machine learning algorithm from Chapter 2 (random forests algorithm) is used as the data-driven model for ROP prediction. A novel way to formulate hybrid models has been discussed by presenting two broad strategies: ensembles of a single deterministic model (hybrid-One) and ensembles of several deterministic models (hybrid-N). Both types of hybrid models are evaluated for ROP predictive accuracy and model inference on twelve different formations by running experiments on data measured while drilling a well in the Williston Basin, North Dakota. A case study for Mission Canyon limestone shows the application of each drilling model in detail – for prediction and inference. Hybrid models result in a higher ROP prediction accuracy when compared to deterministic models. Additionally, they can be used as effective inferential tools, unlike data-driven models.

3.2 Dataset and Experiments

The dataset used for validation follows the same field-based data as presented in Chapter 2. Experiments are conducted to test the accuracy of different models for ROP prediction on measured field data. The same methodology as presented in Chapter 2 (section 2.2) has been used for ROP prediction and evaluation.

3.3 Theory: ROP Modeling

Deterministic models used for ROP modeling – the Bingham, Motahhari and BY model – have been described in Chapter 2 (section 2.3). Data-driven models used for ROP prediction have been analyzed in Chapter 2 (section 2.3).

3.3.1 Hybrid Models

These models combine deterministic and machine learning models; their objective is to provide a trade-off between interpretability and prediction accuracy. Two approaches are utilized to build a hybrid model: ensembles of a single deterministic model (hybrid-One) and ensembles of several deterministic models (hybrid-N).

3.3.1.1 Hybrid-One Model

This model uses a single deterministic ROP model within an ensemble algorithmic framework. Deterministic models have fitting parameters (or constants) which are determined based on the geology and drill bit design. The Hybrid-One model uses an ensemble machine learning algorithm to effectively fit these parameters in batches.

The ensemble algorithm combines many versions (or "realizations") of the same deterministic model to yield a hybrid model. Figure 3.1 shows a flowchart of this algorithm by using Bingham's model as a base. 'N' versions of Bingham's model are combined using an ensemble algorithm such as regression, bagging or random forests. A "version" of Bingham's model is a model with unique fitted coefficients. These coefficients are selected by fitting the deterministic model on a subset (or batch) of the training set. For example, a Bingham model can be built using a random one-tenth of the training set. This constitutes a form or version of the model as shown in Figure 3.2. This is repeated 'N' times. All 'N' models are fed into an ensemble algorithm which is used to combine all versions of the model to a single predictive model – the Hybrid-One model. This algorithm works on the premise that many input features can be underrepresented in the final model while fitting the entire formation at once. Alternatively, this two-layered fitting procedure guards against outliers and increases the influence of underrepresented input features. The analogy used in this algorithm is similar to ensembling machine learning algorithms; ensembling many algorithms work on the premise that some models correct or capture deficiencies of other models better. Since deterministic models are not stochastic in nature, the training set is partitioned into 'N' parts; a model trained on each "Nth" part acts a "version" of the deterministic model. These N parts can be obtained by splitting the data set into "N" parts or with the use of bootstrapping. The overall flavor of this algorithm follows bagging closely.

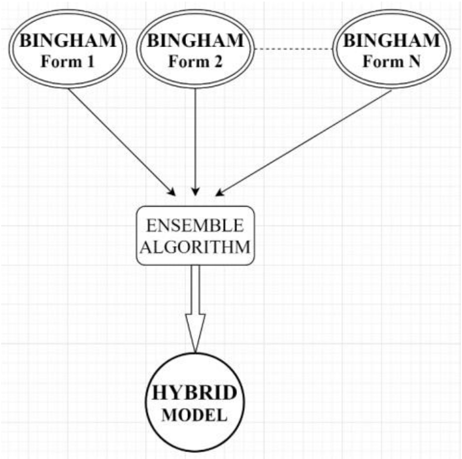

Figure 3.1: Schematic for building a Hybrid-One model using Bingham's model as the base deterministic model; Several versions of the Bingham's models are combined using an ensembling algorithm

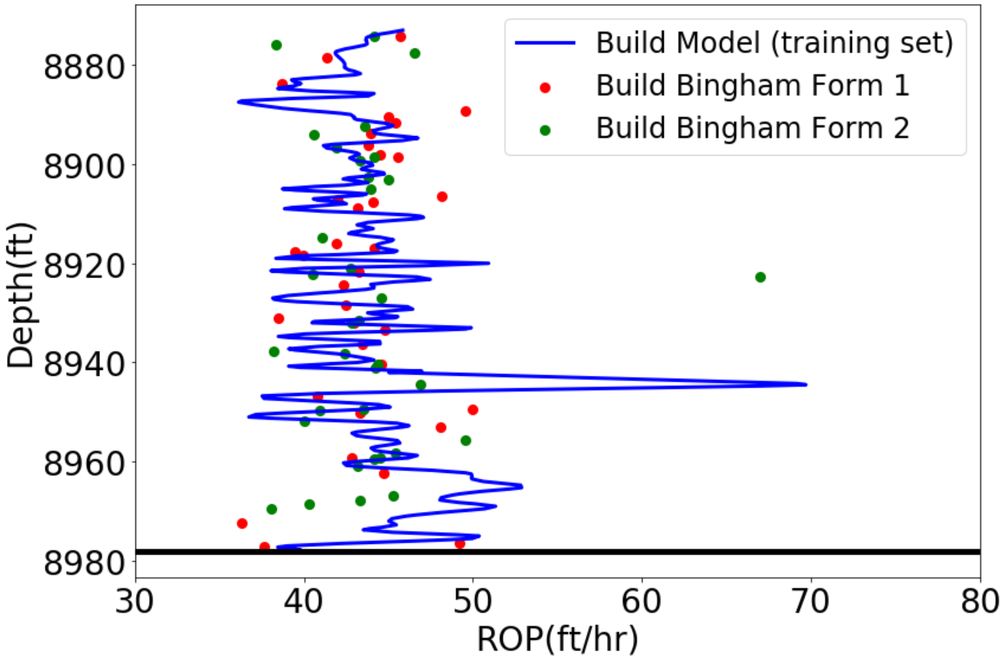

Figure 3.2: Building a form of a Bingham model; Training set is randomly sampled as shown in the figure for data, which is used to fit a Bingham model. This fitted Bingham model is called a "version" of a Bingham model.

3.3.1.2 Hybrid-N Model

All three previously described deterministic ROP models are combined to provide better ROP predictions. This is based on the premise that certain downhole conditions which are not explained by one model can be explained using another model – similar to intuition behind an ensemble model containing many machine learning algorithms. Therefore, it is hypothesized that the deficiencies of models are canceled out by the other models in the ensemble – work together mimicking a team effort. Each model is assigned a weight (w_i) determined mathematically to reduce the overall error in ROP prediction. Machine learning algorithms are used to determine the weight since they account for correlation between the input ROP models and account for overfitting. Figure 3.3 shows a schematic of the hybrid-N model where three deterministic models – Bingham, Hareland

and BY model – are combined using an ensembling algorithm to yield a predictive model. This template can be easily extended to numerous other oil and gas applications such as prediction of production using different models, decline curve analysis for better accuracy, history-matching, etc. A constraint is often beneficial for ensembling different algorithms to ensure that the optimization problem is well defined (Friedman et al., 2001). The weights of each model are set to only take on values between 0 and 1. Additionally, the sum of all 'N' model weights are set to 1 (this makes it a geometric programming problem, making the optimization easy (Friedman et al., 2001)). There can be cases where some model weights are driven to zero, indicating that the model in question does not contribute to the final model.

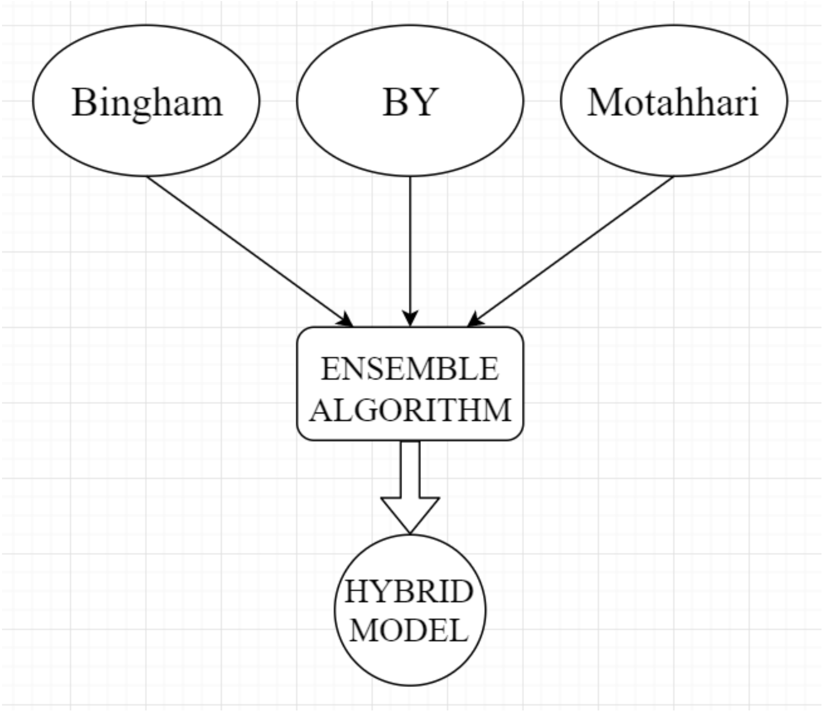

Figure 3.3: Schematic for building Hybrid-N ROP model using three deterministic models; Compared to the Hybrid-One algorithm, the hybrid-N algorithm combines different models using an ensembling algorithm

3.4 ENSEMBLING ALGORITHMS

This section describes algorithms which are used for the development of hybrid and data-driven ROP models. KNN, trees, bagging, neural networks and random forests can also be used as ensembling algorithms by feeding them meta-data (data already processed by models) instead of field measured data. Since the application and theory of these algorithms remain the same, the reader is referred to the previous chapter (section 2.3). New algorithms are presented in this section.

3.4.1 Mean

This algorithm would be applied to build the hybrid-N model; it predicts ROP by using an n^{th} of the 'N' inputs (Equation 4). For the case shown in Figure 3.3, since three input models were used, ROP is predicted using a third of each model's prediction. While not a machine learning algorithm, this method has been presented given its sheer simplicity. The premise is that the 'N' ROP models used are dependent on different drilling parameters and can compensate for each other. For example, Bingham's ROP model highly depends on ROP and WOB, whereas Motahhari's model is the very sensitive to rock strength.

$$\text{ROP} = \frac{1}{N}\sum_{n=1}^{N} D_n, \text{ (Equation 3.1)}$$

where, 'N' is the number of input deterministic models and D_n is the ROP prediction of the n^{th} model.

3.4.2 Stacking

Regression ensembling is the simplest machine learning technique which assumes that the data are linearly related. For the hybrid-N algorithm, it assumes that the metadata - predictions of ROP models themselves - can be combined linearly to predict the response

(Equation 3.2). Regression assigns each deterministic model with a weight, which measures its relative contribution to the final ROP model. The least squares algorithm is used to minimize the l_2 norm. The premise of using linear regression is that ROP predictions from different models are sufficiently non-linear; these non-linear metadata can be combined linearly to predict the response.

$$\text{ROP} = \sum_{n=1}^{N} \mathrm{w}_n D_n, \qquad \text{(Equation 3.2)}$$

where, w_n is the weight of each prediction, and D_n is the ROP prediction of the n$^{\text{th}}$ deterministic model. Since these models are prone to overfitting, adding constraints often helps improve the efficacy of regression. By constraining the weights (w_n) (or squares of weights) to sum to 1 and remain non-negative the problem is better formulated (called stacking). This ensures that stacking does not assign higher weights to more complex models and converts the least squares regression to a quadratic programming problem (Friedman et al., 2001).

3.4.3 Ridge Regression

This algorithm adds constraints to the regression equation by adding a regularization term (Equation 3.2).

$$\text{ROP} = \sum_{n=1}^{N} \mathrm{w}_n D_n + \lambda \sum_{n=1}^{N} w_n^2 \ , \quad \text{(Equation 3.3)}$$

$$\text{ROP} = \sum_{n=1}^{N} \mathrm{w}_n D_n + \lambda \sum_{n=1}^{N} w_n \ , \quad \text{(Equation 3.4)}$$

where, w_n are the weights of each prediction, and D_n is the ROP prediction of the n^{th} deterministic model, λ is a tuning parameter selected using cross-validation. The extra term in Equation 3.3 as compared to Equation 3.2 is the regularization term; it adds a constraint on the weights which helps prevent overfitting by penalizing large weights (making the model less flexible). The use of the $l_1$constraint instead of l_2 is called lasso regression (Equation 3.4). The lasso acts as a feature selector and drives weights to zero; the lasso is particularly useful in cases with large a number of features resulting in a sparse model (Efron & Hastie, 2016).

3.5 Results and Discussions

Models are evaluated in this section based on their test errors. 40% of the data in each formation are used for building the model (training set) and the rest to evaluate accuracy (test set). Model hyperparameters are finetuned using cross-validation on the training set; trained models are then evaluated on the test set for an unbiased test accuracy. Deterministic models and machine learning algorithms were used to model ROP based on field data. Figure 3.4 summarizes the performance of using deterministic models for ROP prediction. Results indicate that the random forests algorithm outperforms deterministic models for ROP prediction. Among the deterministic models, the Motahhari model performs better than Bingham and the BY model. In a few formations, the deterministic models yield acceptable results. However, in most cases, they underfit and result in low test accuracy.

A random forests model built using WOB, RPM, flow-rate and UCS as input features produced lower error rates than all other model explored so far. Different machine learning algorithms have been evaluated for ROP prediction (Figure 3.5) of which random forests outperformed all others in test accuracy paralleling results seen in literature (C.

Hegde, Daigle, Millwater, & Gray, 2017; Chiranth Hegde & Gray, 2017; Chiranth Hegde et al., 2015).

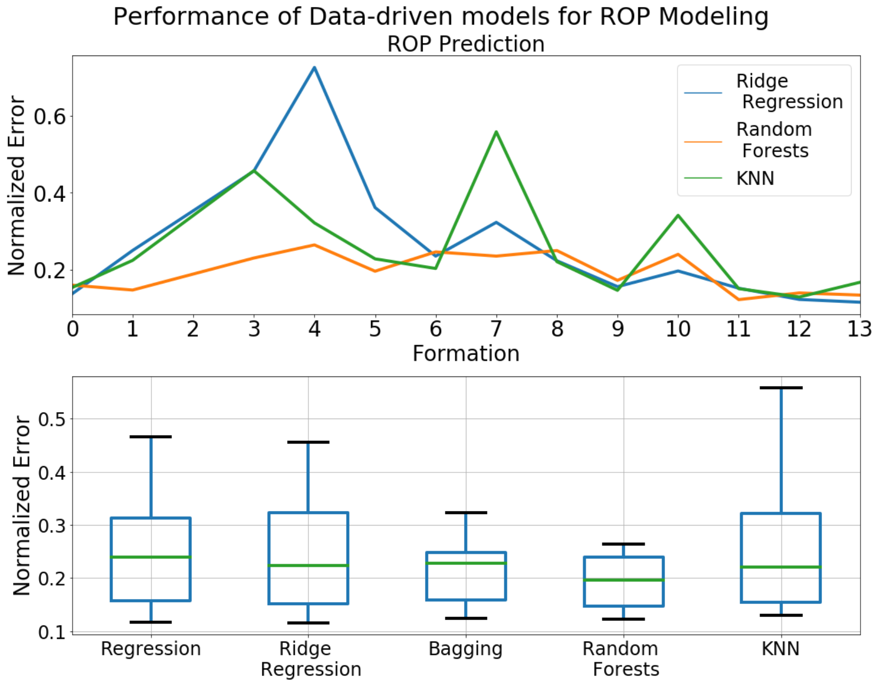

Figure 3.4: ROP prediction using data-driven models (40/60 split). Random forests ROP prediction has the lowest error as compared to all other statistical learning algorithms

Regularized (or ridge) regression, KNN, and bagging were used as ensembling algorithms to combine each deterministic model using the hybrid-One algorithm. Since the input models are used as metadata for the ensemble, this model is prone to overfitting (especially if the different "realizations" of the deterministic models are similar). Ridge regression and bagging gives some protection against overfitting and serves as good candidates for ensembling. Figures 3.6, 3.7, and 3.8 show the results of using a Hybrid-One model for ROP prediction. Ridge regression produces the best hybrid models using

Bingham and BY as a base deterministic model resulting in a significant increase in test accuracy over their vanilla deterministic counterparts. For the Motahhari model, using bagging works well and improves accuracy. Overall, the results show the hybrid-One model results in better ROP prediction as compared to purely deterministic models as hypothesized. While these models still do not always outperform machine learning models, most hybrid-One models come close.

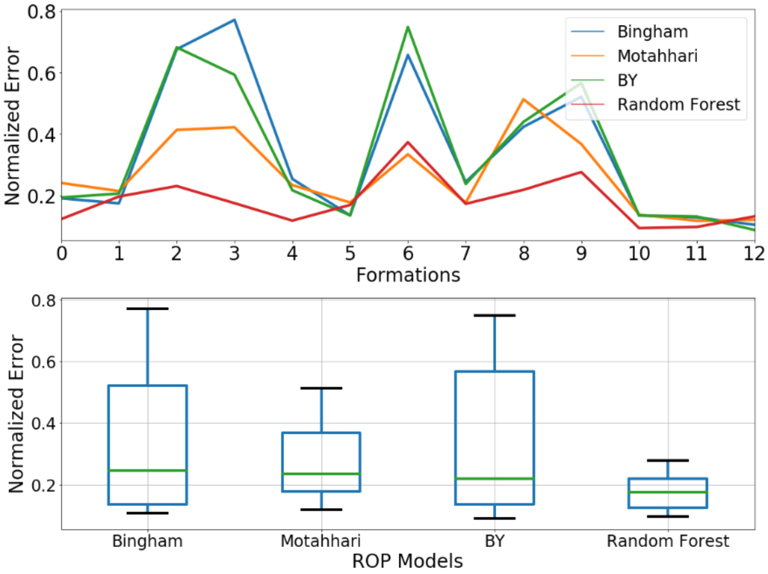

Figure 3.5: Performance of ROP models on test data for the entire dataset (40/60 split). Three deterministic models and one machine learning algorithm have been evaluated for test set accuracy.

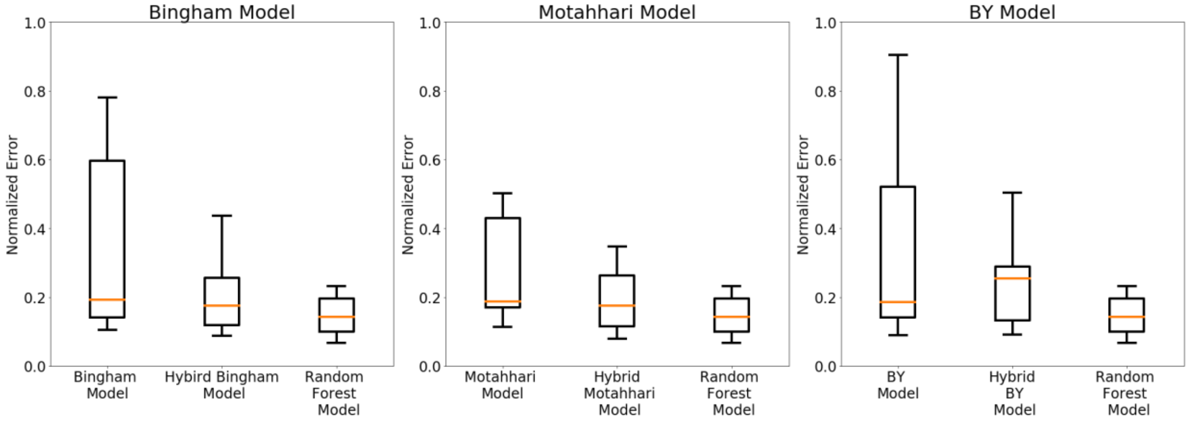

Figure 3.6: Hybrid-One models using the bagging algorithm. Bingham and Motahhari hybrid-One models perform better than the deterministic models; the hybrid-One BY model has a higher mean error than the deterministic model itself; all models perform worse than machine learning models.

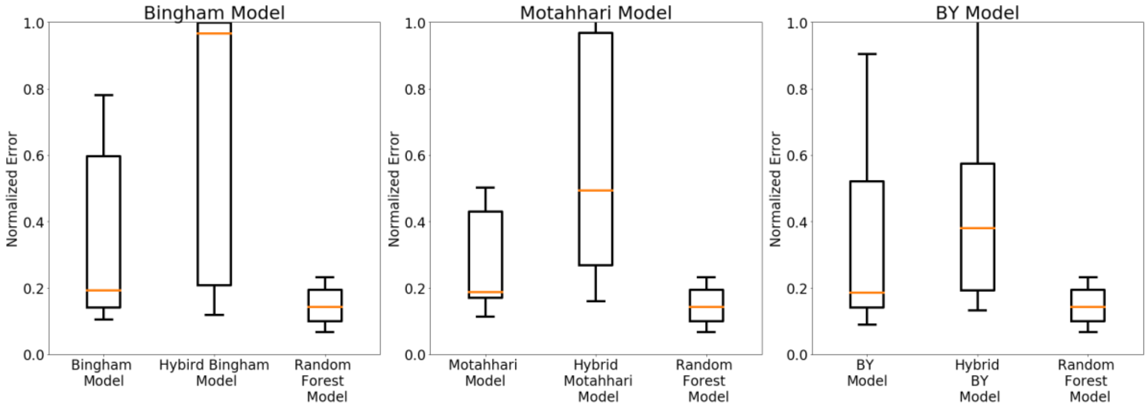

Figure 3.7: Hybrid-One models using the random forest algorithm. All three hybrid-One models perform worse than the deterministic models and machine learning models.

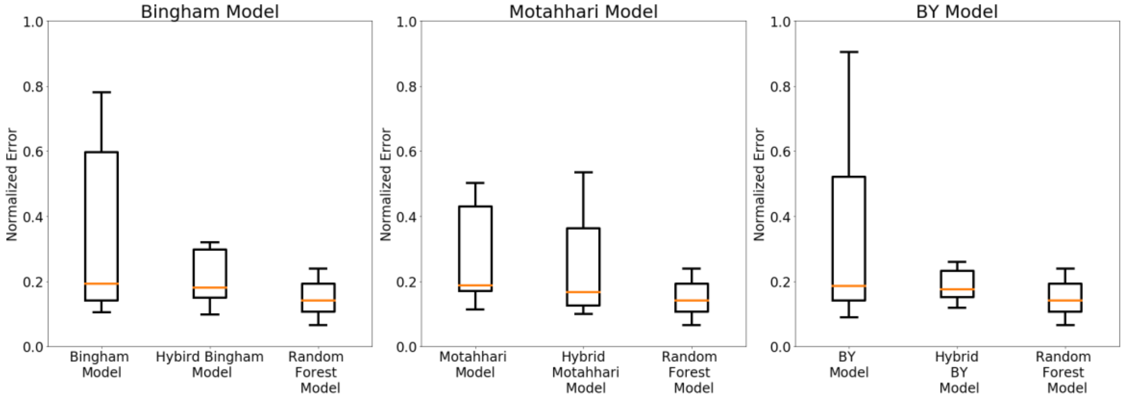

Figure 3.8: Hybrid-One models using the ridge regression algorithm. The hybrid-One with Bingham and BY models perform much better than the deterministic models. The hybrid-One Motahhari model does not perform as well as the deterministic model overall. Whereas the machine learning model outperforms all models.

Using the hybrid-N algorithm, models were built and evaluated against test data. A total of seven algorithms – mean, regression, bagging, random forests, KNN and stacking – were used to combine deterministic models in an effort to identify the best hybrid-N algorithm (Figure 3.9). Results indicate that the stacking algorithm outperforms other evaluated algorithms. Additionally, stacking is a linear algorithm which makes its interpretation easy; Table 3.1 summarizes the weights assigned to each deterministic model by the stacking algorithm. These weights can be considered to be a proxy for the importance of each deterministic model in the final ROP model.

Data-driven models perform well in most cases for ROP prediction. In certain instances, with a shortage of data, it may be more feasible to use a deterministic model. Hybrid models provide an alternative way to fit deterministic models which often boosts accuracies and still provides some interpretability. In general, for a field application, a bag-of-models approach should be preferred – the best model for that specific formation based on cross-validation error is used as opposed to an overall best model for the entire well.

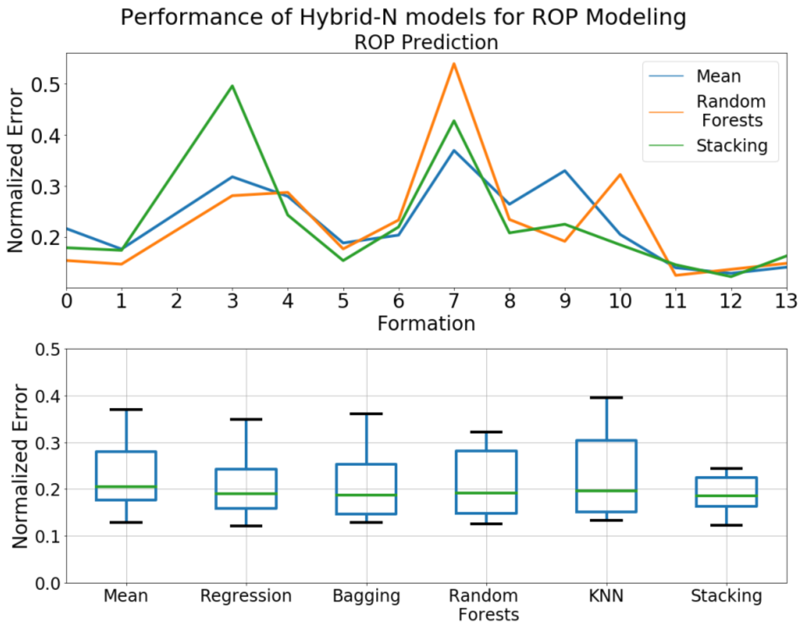

Figure 3.9: Performance of hybrid-N models for ROP modeling. A total of seven algorithms have been evaluated to determine the best algorithm for the use of the hybrid-N model.

Table 3.4: Weights assigned to each formation based on stacking algorithm used to create a hybrid-N ROP model

	Hybrid-N Model Contribution		
Formation	**Bingham**	**Motahhari**	**BY**
Piper Limestone	0	0.35	0.65
Spearfish Sandstone	0.59	0.21	0.2
Pine Salt Sandstone	0	0.77	0.23
Broom Creek Sandstone	0	0.64	0.36
Tyler Sandstone	0.1	0.76	0.15
Kibbey Lime Limestone	1	0	0
Kibbey Lime Shale	0	0	1
Charles Sandstone	0	0.6	0.4
Charles Limestone	0.05	0.76	0.19
Ratcliffe Sandstone	0.65	0.35	0
Base Last Salt Limestone	0.04	0.91	0.04
Base Last Salt Sandstone	0.25	0.26	0.49
Mission Canyon Limestone	0.11	0.89	0
Lodgepole Limestone	0	0.24	0.76

3.5.1 Model Analysis and Interpretation

Deterministic models provide utmost transparency in the ROP modeling process. The exact form of the equation is known, which makes model interpretation (or inference) easy. The parameters which affect the ROP – RPM, WOB, UCS, and flowrate – are built into the equation; hence, once the constants are determined it is easy to carry out an inferential analysis. For example, if the best fit Bingham model had RPM raised to 0.5 and WOB raised to 1.5, then WOB would be the most influential parameter which affects ROP while drilling that formation. The problem with these models is their accuracy.

Alternatively, machine learning based ROP models result in much higher predictive accuracy, however, they have low inferential capabilities (except in the case of linear regression-based methods). The random forest algorithm allows the ranking of input features (or compute the importance of each input feature) in the model which can be used as a proxy for inference. An example of this feature importance has been shown in Figure 3.10. It is calculated by measuring the increase in information (or Gini index) with a split in the decision tree for a given input feature. This is averaged across all trees and plotted (Figure 3.10). Since the random forest and bagging algorithm builds trees using a greedy algorithm the feature importance may not be representative of the true physical phenomena; they merely represent the feature importance in the model. Additionally, it does not provide a direction of change but only a magnitude of importance.

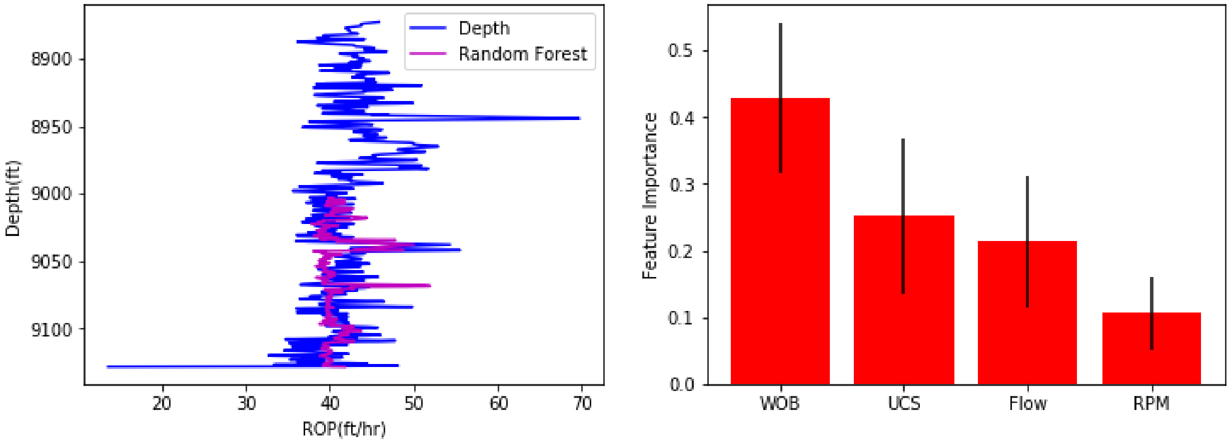

Figure 3.10: (Left) ROP prediction using random forests in Lodgepole Limestone; (Right) Drilling parameter interest using random forests for ROP prediction in Lodgepole Limestone

Hybrid models provide a tradeoff between accuracy and ability to perform model inference. The hybrid-One algorithm introduced an alternative method to fit and tune hyperparameters using deterministic models. Since it is an additive model, it can be used to determine the most influential sub-models which can be used for inference (since these sub-models are “realizations” of a deterministic model). The hybrid-N algorithm provides a simple method to evaluate the importance of a deterministic model for each formation. This can be traced back to the most influential input parameter of the dominant model. The Bingham model calculates ROP based on the change in RPM and WOB. The most dominant term added in Motahhari’s model is UCS (when compared to Bingham’s model); BY similarly adds flow-rate. Hence, it can be hypothesized that the weight of a deterministic model (Tab1e 3.1) in the hybrid-N algorithm is representative of the dominant drilling operating parameters for that formation (note that WOB and RPM are combined into one parameter). This provides a proxy for model importance and thereby input feature importance. This model importance can be compared to the feature importance which is calculated using the random forests algorithm (Table 3.2).

Table 3.5: Comparison of random forest feature importance and hybrid-N model weights. Important random forest features are highlighted in blue and green for high hybrid-N weights

	Random Forest Importance				Hybrid-N Model Contribution		
Formation	**WOB**	**RPM**	**Flow**	**UCS**	**Bingham**	**Motahhari**	**BY**
Piper Limestone	0.35	0.2	0.11	0.34	0	0.35	0.65
Spearfish Sandstone	0.28	0.15	0.31	0.25	0.59	0.21	0.2
Pine Salt Sandstone	0.08	0.43	0.31	0.18	0	0.77	0.23
Broom Creek Sandstone	0.37	0.19	0.07	0.36	0	0.64	0.36
Tyler Sandstone	0.61	0.07	0.04	0.28	0.1	0.76	0.15
Kibbey Lime Limestone	0.17	0.24	0.02	0.58	1	0	0
Kibbey Lime Shale	0.5	0.16	0.13	0.22	0	0	1
Charles Sandstone	0.22	0.09	0.15	0.54	0	0.6	0.4
Charles Limestone	0.27	0.19	0.14	0.41	0.05	0.76	0.19
Ratcliffe Sandstone	0.58	0.12	0.15	0.14	0.65	0.35	0
Base Last Salt Limestone	0.32	0.19	0.14	0.35	0.04	0.91	0.04
Base Last Salt Sandstone	0.47	0.15	0.1	0.28	0.25	0.26	0.49
Mission Canyon Limestone	0.26	0.12	0.09	0.53	0.11	0.89	0
Lodgepole Limestone	0.43	0.1	0.2	0.26	0	0.24	0.76

A higher feature importance of UCS is associated with a higher weighting of the Motahhari model in all cases. In a few cases where the random forest model associates high feature importance to RPM and WOB, the Bingham model is dominant or most relevant for the Hybrid-N model. Moderate importance of all features is associated high weight of the BY model – the BY is made of eight input parameters and is not overly dominant in one feature, making this is a plausible result (as opposed to the initial hypothesis that BY is a flow-rate dominant model). This also shows that the modeling schemes presented in chapters 2 and 3 are essentially moving in the same direction; some methods are more flexible than others which may result in a better fit.

3.6 Case Study: Mission Canyon Limestone

This section applies previously discussed ROP analysis on the Mission Canyon formation which is made up of limestone. The dataset was separated into training and test sets with a 40/60 split.

3.6.1 ROP Prediction

ROP modeling is undertaken on the training set. The Motahhari model is used as the base deterministic model for this formation since it was most accurate among the deterministic models. The random forests algorithm is built using 1000 trees and a feature subset of 2; WOB, RPM, flow-rate, and UCS are used as input parameters. All deterministic models are stacked using the stacking algorithm for the Hybrid-N ROP predictor; the model's weights are displayed in Table 3.2. 100 Motahhari models are ensembled using the random forest algorithm to create the hybrid-One model. The test errors and ROP predictions of these models have been plotted in Figure 3.11; all models perform well producing low test error; both hybrid models perform better than the deterministic and random forest model.

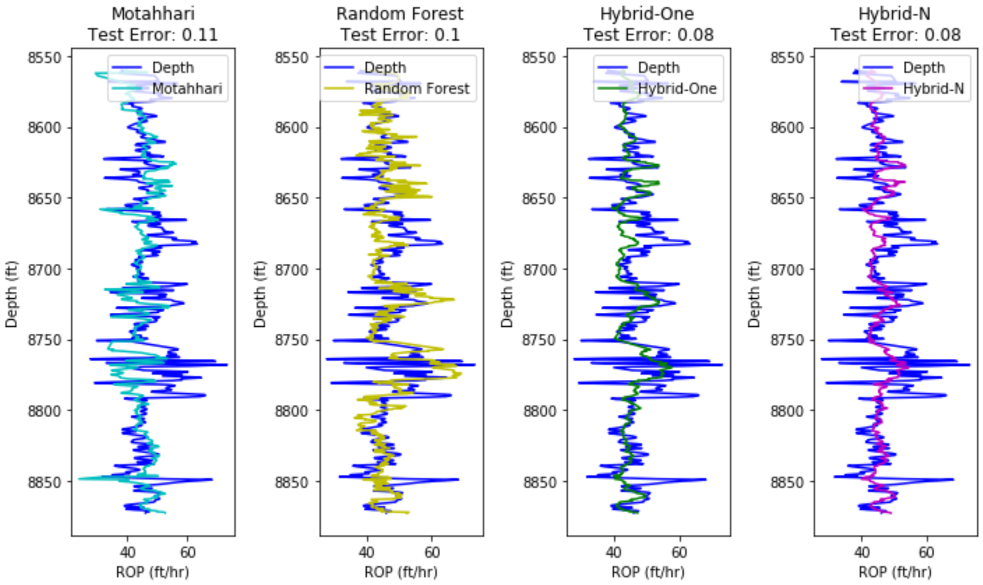

Figure 3.11: ROP predictions of models developed for Mission Canyon Limestone; (Left) ROP predictions of deterministic Motahhari model; (Left -middle) ROP predictions of data-driven random forest model; (Right-middle) ROP predictions of the hybrid-One model; (Right) ROP predictions of the Hybrid-N model;

3.6.2 ROP Inference

ROP models built in the previous section are used for inference. The deterministic model used – Motahhari's model – has three formation related constants. α and γ represent the exponents of WOB and RPM in Equation 2.5. For this formation, α is 1.064 and γ is 1. This shows that a percent change in WOB will have a higher impact on the ROP than a percent change in RPM providing field intuition.

Random forest's feature importance plot (Figure 3.12) emphasizes that UCS is the most important parameter influencing ROP followed by WOB, RPM, and flow-rate. Hence, to change ROP while drilling this formation, WOB is the key controlling parameter, since rock strength is not a controllable parameter. This method does not provide the direction of change like the deterministic model.

Hybrid-N model weights are 0.11 for Bingham, 0.89 for Motahhari, and 0 for the BY model. Zero weight assigned to a model implies that the BY model does not help improve ROP predictions. From an interpretation perspective, the dominant drilling feature of the excluded deterministic model – flow-rate in this case – is not an important feature influencing the ROP. This is supported by the random forest model's feature importance plot shown in Figure 3.12. The weights of the hybrid-N model imply that rock strength is the key feature which affects ROP (followed by WOB and RPM). These results align perfectly with the previously discussed models. However, in this case, unlike random forests, it is possible to derive the exact relationship between ROP and the input features. The model is composed of two ROP models – 0.11* Bingham + 0.89*Motahhari; each of these models has a relationship between ROP, WOB, and RPM which can be used for inference in the form of an additive model as shown in Equation 3.5.

$$ROP \propto 0.11 * RPM * WOB^{0.5} + 0.89 * RPM * WOB^{1.064}$$, (Equation 3.5)

The hybrid-One model combines different versions of a deterministic model. Bagging was used for ensembling Motahhari models (since it gave the best ROP predictions); the feature importance of the bagging predictor can be used to determine the most prominent versions of Motahhari's models (Figure 3.13). For Mission Canyon formation, the most influential version of the Motahhari model is the 4th version; for which the empirical constants α and γ are 1. Other models have decreasing amounts of influence on the model; the empirical constants for these versions can be calculated to form an additive equation for ROP inference (similar to Equation 3.5). However, in this case extracting the empirical constants of versions 1,6, and 0 provide no additional information. This provides the same level of inference as the deterministic model but fits the data better:

resulting in higher accuracy. This method of model fitting and inference bears the same flavor as the boosting algorithm. In this case, the best deterministic model is fit to the data and the portion of data left unexplained (residuals) are refit with the same deterministic model to reduce prediction error. Inference can be performed similarly on the hybrid-N model; the terms in Equation 8 for this method would all be composed of the different "realizations" of the same model.

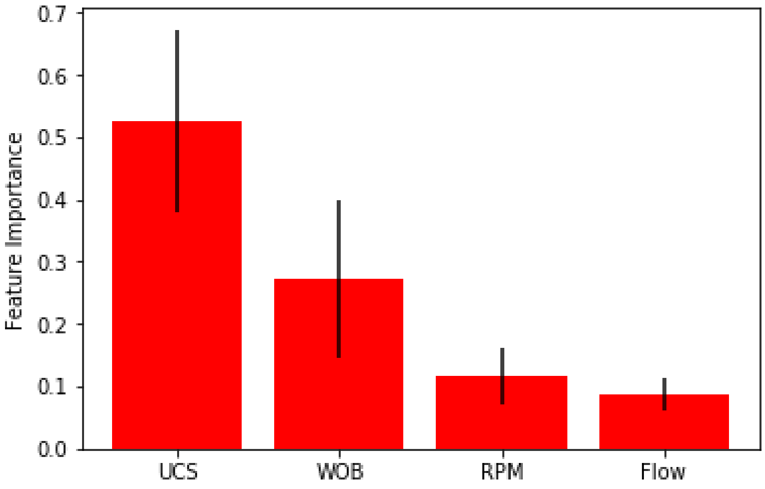

Figure 3.12: Random forest ROP model input feature importance for Mission Canyon Limestone.

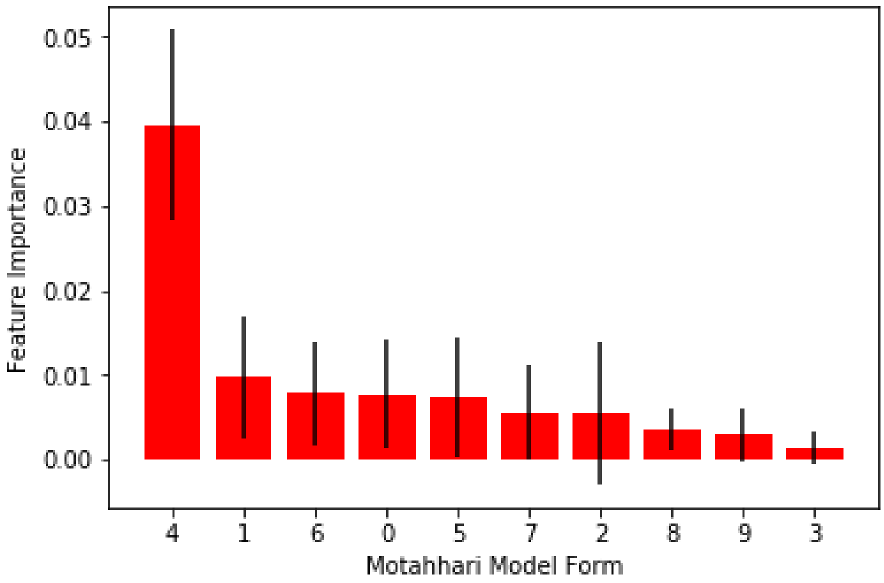

Figure 3.13: Hybrid-One model feature importance for Mission Canyon Limestone

3.7 Conclusions

Deterministic models are often the industry standard for ROP prediction. Although sound in physics, these models introduce many empirical coefficients and functional constraints for fitting which often lead to poor results. Machine learning models, on the other hand, utilize data to predict ROP; machine learning models are more flexible allowing them to be better ROP predictors.

Empirical coefficients in deterministic models are calculated by conditioning the models to the data – the training set. Data-driven models, on the other hand, rely purely on the data. Hybrid models combine the two analogies. Two algorithms for building hybrid ROP models have been discussed: hybrid-One and hybrid-N. The hybrid-One model combines different versions (or realizations) of a single deterministic model – providing an alternative method to determine empirical constants. The hybrid-N model combines predictions from different deterministic models using an ensembling algorithm. This is based on the premise that one model can compensate for the weaknesses of other models present in the ensemble. The hybrid model algorithms introduced in this paper can easily be extended to other applications in oil and gas.

Traditional, hybrid, and data-driven models were simulated to predict ROP in different formations. Models were evaluated using normalized ROP test error. The machine learning model – using random forests – outperformed all other models in terms of predictive accuracy on average. However, adopting the bag-of-models analogy can be more beneficial – this method selects the best model for each formation based on the cross-validation error. It produces better results as opposed to selecting an overall best algorithm.

In general, ROP data are non-linear; deterministic models attempt to estimate this nonlinear data using power law or exponential functions and as a result underfit the data. However, they provide great inferential insight into ROP modeling, which is not available using machine learning. The hybrid models can also provide inferential insight using additive ROP equations. For the hybrid-One model, the "version (s)" of a deterministic model with the highest weight(s) (or that which explains the highest variance) can be used for inference. The hybrid-N algorithm provides valuable insights: the weight of a deterministic ROP model in the hybrid-N algorithm can be used to gain information about the dominant drilling parameter which influences ROP. Feature importance of random forests and the weights of the hybrid-N model showed synergy – indicating that both modeling techniques were converging to the same solution. A case study implements these ideas to model ROP and gain insight into the drilling process for Mission Canyon Limestone formation.

Chapter 4: Optimization of drilling models[4]

The previous chapters covered the modeling of parameters such as ROP and TOB using deterministic and data-driven algorithms. Metrics such as ROP, TOB, and mechanical specific energy (MSE) are of key interest in drilling since they relate to the productivity of the rig. The purpose of modeling these metrics is to better understand the drilling process. This understanding can lead to better decisions and improvement of these key metrics.

4.1 Introduction

The drilling models themselves have been modeled as a function of drilling control parameters (which can be controlled on the surface of the rig); these models can be used to find key drilling control parameters which improves drilling metrics. For example, since ROP was modeled as a function of RPM, WOB, flowrate, and UCS – the ROP model can be inverted to determine the optimal settings of ROP, WOB, and flowrate which results in the highest ROP. This optimization methodology is prolific in science and engineering; drilling optimization by specifying an objective function can be traced back decades – where ROP was shown to improve by solving an optimization problem in order to maximize ROP (Lummus, 1970). The objective function – namely, the ROP – was maximized for economic benefit in drilling since ROP is inversely proportional to the cost of the well. While many papers have discussed the optimization of ROP, the same concept can be easily extended to other metrics to optimize TOB, reduce drilling vibrations,

[4] Hegde, C., Daigle, H., & Gray, K. (2018). Performance comparison of algorithms for real-time rate of penetration optimization in drilling using data-driven models. *SPE Journal*.

The author of this thesis is the primary author of the paper

optimize MSE and so on. This chapter will discuss the optimization of ROP using physics-based and data-driven models; the key concept of optimization remains the same, hence the methodology and algorithms discussed here can be applied for the optimization of other drilling metrics such as TOB or MSE.

Most discussions of optimization algorithms have been geared towards post-drilling analysis or planning tools and not real-time applications. Modeling the ROP in real-time (concurrent with drilling operations) is more complicated given the dynamic conditions, data collection, data quality, data transmission, and computational speed requirement (Bybee, 2011). However, real-time optimization of ROP can be more effective in saving operational costs. It is common practice to try and change drilling control parameters on the rig to improve ROP in real-time. This optimization is commonly carried out based on experience and intuitions of the driller: in the absence of any mathematical framework.

Optimization of analytical or deterministic models in a real-time scenario does not suffer from computational run-time constraints since the equations (objective functions, at least in drilling) are generally convex or smooth and in general can be solved for optimal parameters (Meng, Haige, Jinying, Liu, & Zhixue, 2015). A more complicated response equation can be solved using meta-heuristic algorithms such as swarm algorithms (Self, Atashnezhad, & Hareland, 2016) or the shuffled frog algorithm (Yi, Kumar, & Samuel, 2015).

Research related to ROP optimization of data-driven models have been discussed for post-drilling analysis without considering any computational time constraint (Chiranth Hegde & Gray, 2017). Data-driven models have been known to be disadvantageous for real-time applications since they can be non-linear functions whose functional form is unknown; making it difficult to optimize data-driven models in real-time. The nonlinearity is essential for an accurate ROP model.

This chapter covers an approach to optimize drilling models. A strategy to optimize ROP models – physics-based and data-driven – has been discussed. Of late, literature seems to consist of two classes of ROP models – physics-based and data-driven (especially using neural networks); however, the integral part of the problem, which is the utilization of these models for determining optimal drilling control parameters are generally not covered. To address these gaps in literature this chapter delves into the optimization – algorithms, cost functions, and computational constraints – of these drilling models so that they may be used for effective simulation and analysis. Additionally, applications of these algorithms for real-time optimization have also been addressed.

This chapter introduces three types of optimization algorithms one for optimizing traditional models – gradient ascent, and two types of algorithms for data-driven models: simple and metaheuristic algorithms. The algorithms are used to invert the ROP model – use the model to determine drilling control parameters which maximize the ROP. The data-driven model used in this chapter is equivalent to the model introduced in chapter 2. Optimization of a deterministic model is easy and computationally effective. For data-driven models since the optimization is not straightforward, five algorithms are evaluated based on ROP increase (percentage of ROP improvement with respect to the base case) while noting their computational efficiency (algorithm run-time on drilling data). A computationally efficient algorithm is not required to make recommendations each second but to have a continuous supply of optimal parameters on the screen with minimal data-lag, i.e. a model built using the most recently available data. Operators must choose the most suitable algorithm based on available resources. Practically making a change every stand would be more viable than every minute or 10 minutes. The algorithms are evaluated by running simulations on data measured during drilling a well in the Williston Basin, North Dakota as described in chapter 2. Results indicate that the particle swarm algorithm produces the best set of drilling parameters to maximize the objective for the data-driven models. However, given its computational inefficiency and its effect on data-lag, for a real-time drilling scenario, it is better to opt for the simplex or random search method. The simplex and random search methods perform worse in terms of ROP improvement but are computationally efficient for real-time closed-loop drilling optimization and result in an improved ROP. The tradeoff between ROP increase and computational run-time can be important for real-time implementations.

4.2 Overview of Optimization Algorithms

Optimization algorithms are used to optimize a function (target, response or objective). This involves selection of the optimal element (maximum or minimum) from the entire solution space (made up of different variables). Simple optimization problems can make use of calculus to find the optimal solution. For example, if the objective function is : $\mathrm{f(k)} = \mathrm{k}^2 - 5$, the minimum of the function (-5) can be obtained by taking the derivative and setting it to zero. The optimization problem may be constrained (where constraints are placed on variables) or unconstrained. In the case of ROP optimization, the optimization is constrained by drilling control parameters. Additional constraints can be imposed based on the objective function to be optimized or for vibration control (covered in chapters 6 and 7). The selection of the algorithm used for optimization of the objective function – referred to hereby as the optimization algorithm – is crucial since the quality of the solution depends on it. The selection of the optimization algorithm depends on the type of the objective function being optimized. Knowing the functional form of the objective function – polynomial, exponential, log, linear, affine – aids in the selection of the best algorithm.

A function is convex if its epigraph – the set of points on or above the plot of the function – is a convex set. In other words, a function twice differentiable, where the second derivative is always greater than or equal to zero for its entire domain is a convex function. A convex function is special since it can be always be optimized to find the global minimum (Boyd & Vandenberghe, 2010). Convex functions have been studied in science and engineering for decades, and detailed (easily implementable) solutions exist (Boyd & Vandenberghe, 2010). However, the optimization of a non-convex function can pose many difficulties and can be time-consuming. Additionally, there is no guarantee that a global minimum is indeed found. A function can be smooth – gradient is defined everywhere and is a continuous function – or non-smooth (Figure 4.1). Optimization of a smooth function is generally easier since gradients can be computed.

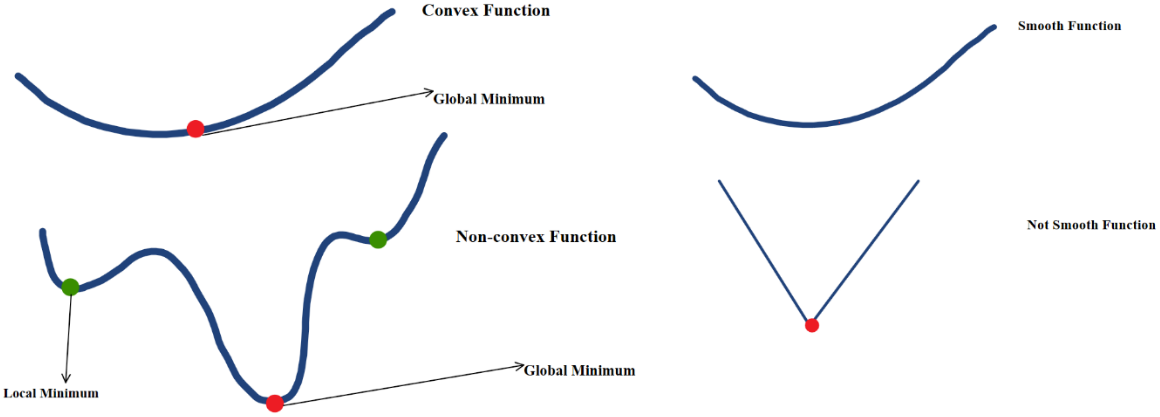

Figure 4.1: Examples of convex, non-convex, smooth and non-smooth functions

Equation-based ROP models are generally convex in nature (or can be converted to convex functions with a transformation) and can be easily optimized to attain a global maximum. ROP can be optimized by finding the maximum of the ROP objective function. For example, in the case of Bingham's ROP model the equation for ROP $-ROP = aRPM((\frac{WOB}{c})^b)$ – is estimated using a power law model. The function is monotonically increasing; the maximum can be obtained when the argument is maximum. In the case of Bingham's model, the maximum ROP occurs when RPM and WOB are both at their maximum values. Data-driven ROP models are unknown in functional form: pre-determined knowledge of the function aiding in optimization is not possible.

Algorithms can be classified as gradient-based (slope is used to determine next step eventually leading to the optimal value) or non-gradient based. Gradient descent is a popular algorithm where small steps are taken in the direction of the gradient of the function (or steepest ascent). This method works well for convex functions. However, in case of non-convex functions, there are possibilities of being stuck in a local minimum. One viable way to use gradient descent for the data-driven model is to re-run the algorithm in a large loop: repeatedly find the optimal parameters (starting each time at a random position), using this algorithm until convergence for many iterations; then compare all converged local minimums to find the global minimum. This method is time-intensive and is not the best way to approach this problem. A meta-heuristic solution has been adopted instead. Meta-heuristics is a subfield of stochastic optimization where a degree of randomness is used to find the optimal solution. It is apt for optimizing data-driven models since a brute-force approach is infeasible. Meta-heuristic algorithms are useful for inverse problems where the function itself is unknown (like the data-driven ROP model in this case). The reader is referred to literature (Boyd & Vandenberghe, 2010; Luke, 2009) for convex-, non-convex-, and meta-heuristics-based optimization literature and further reading.

4.2.1 Solution Space (or optimization space)

The solution space for the optimization algorithms is constrained based on operational, instrumental and field limits. For the purposes of this paper, the solution space was limited by the drilling parameters observed in the training set. For example, the WOB observed in the training set of the lodgepole limestone was between 26-42 klb. The WOB bounds for the algorithm's search was set between these values. However, in practice, there are other considerations which influence the solution space (Chapman et al., 2012).

4.2.1.1 WOB Limits

WOB limits occur due to PDC bit design, tool design, BHA design, where a maximum WOB is specified to prevent damage to the bit. WOB considerations due to buckling of the drill string in case of a high axial load.

4.2.1.2 Surface RPM Limits

Surface RPM limits arise due to PDC bit design, downhole tool design, and motor constraints. Top drive manufacturers specify a maximum torque vs RPM curve which can be utilized to set these limits.

4.2.1.3 ROP Limits

ROP constraints are generally constrained physically due to limitations of hole cleaning and well stability.

4.2.1.4 Pump Limits

Pump limits will influence the maximum flow rate in a well. Pump limitations will be specified by the manufacturer. This can also affect the differential pressure at the bit which will influence the ROP. Differential pressure limits can be calculated for rig equipment in cases of high standpipe pressure to prevent damage to the rig equipment. Other factors include formation fracture and association with lost circulation resulting from high friction imposed due to high flow rates. Additionally, the risk of washouts due to excessive flowrate must also be considered.

4.2.2 Algorithms

Broadly, the optimization algorithms used in this chapter are classified as gradient-based, simple and meta-heuristic algorithms. Gradient-based algorithms are used to optimize traditional ROP models. Simple algorithms are the easiest algorithms to optimize the ROP function. Eyeball method and the random search algorithm fall in this category. Meta-heuristic algorithms implement a form of stochastic optimization. The simplex algorithm, differential evolution-based algorithms, and swarm optimization algorithms have been evaluated.

4.2.2.1 Calculus-based optimization

A simple optimization problem can be solved using calculus analytically. For example, if the objective function is of the form: $\mathrm{f(k)} = \mathrm{k}^2 - 5$, taking the derivative of the function and setting that equal to zero yields a minimum at 0. If functions take on such simple forms, it is possible to evaluate them analytically using calculus. The Bingham ROP model is one such function, which can be evaluated using calculus.

ROP = $aRPM((\frac{WOB}{b})^c)$, (Equation 4.1)

It can be assumed that the constants 'a', 'b', and 'c' are >=0. This is a valid assumption since 'a' and 'b' are dimensionless constants which characterize the drillability of the formation. Taking the derivative of the equation with respect to RPM yields equation 4.2.

$$\frac{d\,ROP}{dRPM} = \frac{a}{b^c} WOB^c, \text{ (Equation 4.2)}$$

Setting equation 4.2 to 0 yields *WOB = 0*. Since the second derivative of Equation 4.1 is 0, it cannot be concluded to be a maximum or minimum. Equation 4.3 is obtained by taking the derivative of ROP w.r.t to WOB.

$$\frac{d\,ROP}{dWOB} = \frac{ac}{b^c} WOB^{c-1} * RPM, \text{ (Equation 4.3)}$$

Setting equation 4.3 to 0 implies that either *WOB* or *RPM* should be 0. The second derivative shown in equation 4 is always greater than 0 in operation since *RPM* and *WOB* are greater than 0. Hence this is a minimum.

$$\frac{d^2\,ROP}{dWOB^2} = \frac{ac(c-1)}{b^c} WOB^{c-2} * RPM, \text{ (Equation 4.4)}$$

The results of this optimum show that a minimum of ROP is achieved when WOB and RPM are set to 0 respectively. The maximum of the function – which is of interest – cannot be obtained by this method.

However, by evaluating the function in a given range (30 <= *RPM* <=60, and 10000 <= *WOB* <= 20000) the maximum ROP can be calculated by looking at the end range of the function since it is monotonically increasing in both variables. In this case, the maximum ROP is achieved at 60 *RPM* and a *WOB* of 20000 lbs as seen in Figure 4.2.

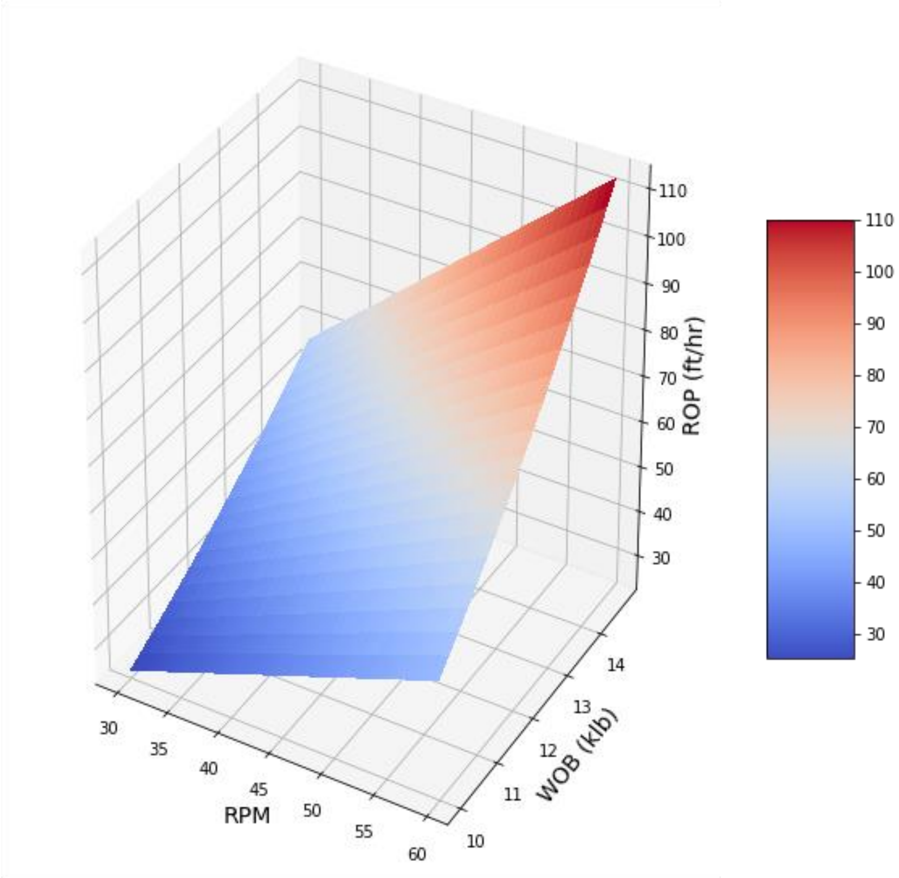

Figure 4.2: Optimization of a Bingham's ROP model using numerical methods. The Bingham's model used in this figure has 'a' of 0.6, bit diameter of 8.75 in, and 'b' of 2.2

4.2.2.2 Gradient ascent

Gradient ascent (or descent) is a simple algorithm that is very useful in searching for the argmax (arguments of a function which lead to a maximum value of the function) or argmin (arguments of a function which lead to a minimum value of the function) of a function. The gradient (or Jacobian) at a point indicates the direction of steepest ascent. By traveling in the direction of steepest ascent, it is possible to eventually reach the maximum. The negative of the gradient is taken to find the minima. In a three-dimensional world, the intuition of gradient ascent can be best explained with a hiking analogy. It is similar to standing in a valley and trying to hike towards the peak of a mountain. By moving in the direction of the steepest, the hiker would eventually reach the mountain peak. This can be expanded to any number of dimensions and the Jacobian replaces the role of the gradient. This has been used very successfully for fitting purposes in neural networks where gradient descent is by far the best algorithm for training (Rumelhart et al., 1988). The speed or time it takes to reach the peak of the mountain depends on the size of the steps that are taken in the direction of steepest ascent. This is often referred to as the learning rate in machine learning, where the step size α can be varied. If α is small, the algorithm will eventually converge towards a maximum, but it may take a long time. If α is large, the algorithm may converge faster, but it may also overshoot and never find the maximum. Gradient ascent is also known as a first-order method because it requires calculation of the first derivative at each iteration. Gradient ascent does not ascertain that the given argmax indeed refers to a global maximum; it is possible to get stuck in a local maximum. One simple way to avoid this is to use random restarts and average the results which work well in practice. An example has been shown in Figure 4.3 where gradient descent has been used to find the minima of a ‘sinc’ function.

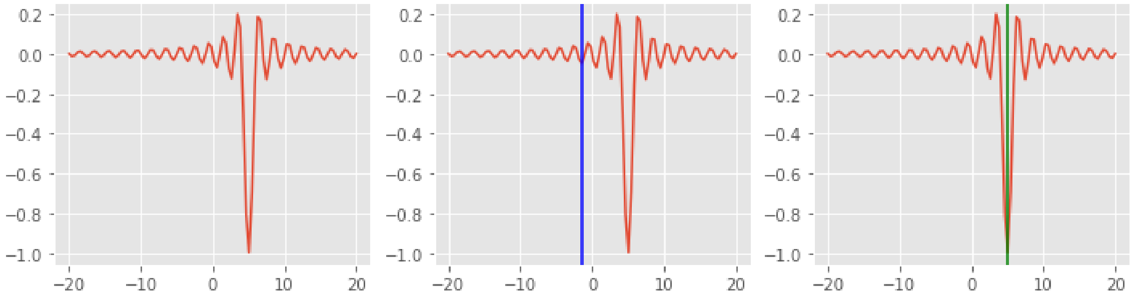

Figure 4.3: (Left) Sinc function plotted between -20 and 20 on the X-axis and 0.2 and -1 on the Y-axis; The objective of the gradient descent algorithm is to find the minima of the function displayed; (Middle) The use of vanilla gradient descent results in a local minima as opposed to a global minima which is evident from the plot; (Right) The use of gradient descent with random return the global minima as opposed to a local minima.

4.2.2.3 Eyeball algorithm

This is a simple algorithm which can be used to improve ROP ahead of the bit for any model. It is similar to field-based rules which are employed in drilling. The training set is evaluated for the best (90th percentile) of ROP data points (drilling parameters which result in the highest ROP) as shown in Figure 4.4. The results of these points are used to determine the optimal settings ahead of the bit. The input settings (RPM, WOB, and flow rate) that produced the highest ROP in the training set are then used ahead of the bit. The best input settings of the training set – drilling parameters that produced the highest ROP in the training set – are then applied ahead of the bit to yield a higher ROP as shown in Figure 4.5. Pseudo code for eyeball method has been described in the Appendix. It is better to choose the 90th – 100th percentile and evaluate a few points as opposed to a single best setting since it decreases the variance in picking drilling parameters. However, it must ne ensured that these points are in the same proximity. In an ideal case, the average of these optimal points in the training set would converge to the global minimum. In the average case, they converge to values which lie in between the local minima and global minima.

This method is extremely simple and will yield an improved ROP if the training data are efficiently sampled i.e. the driller changes RPM, WOB and flow rate to evaluate effects of each input parameter on ROP in the training set. It is extremely easy to implement and bears no computational constraints. A variation of this method is used in drilling pad wells, where the driller or engineer uses intuition based on earlier drilled wells. This is a mathematical approach to rig-based decision optimization (Fred E Dupriest & Koederitz, 2005). This method may not always work since it relies heavily on the assumption that the rock strength, the geology and other drilling conditions remain unchanged ahead of the bit. However, given its simplicity, it would be recommended in drilling long horizontal shale sections where the formation type does not change, and the bit wears gradually.

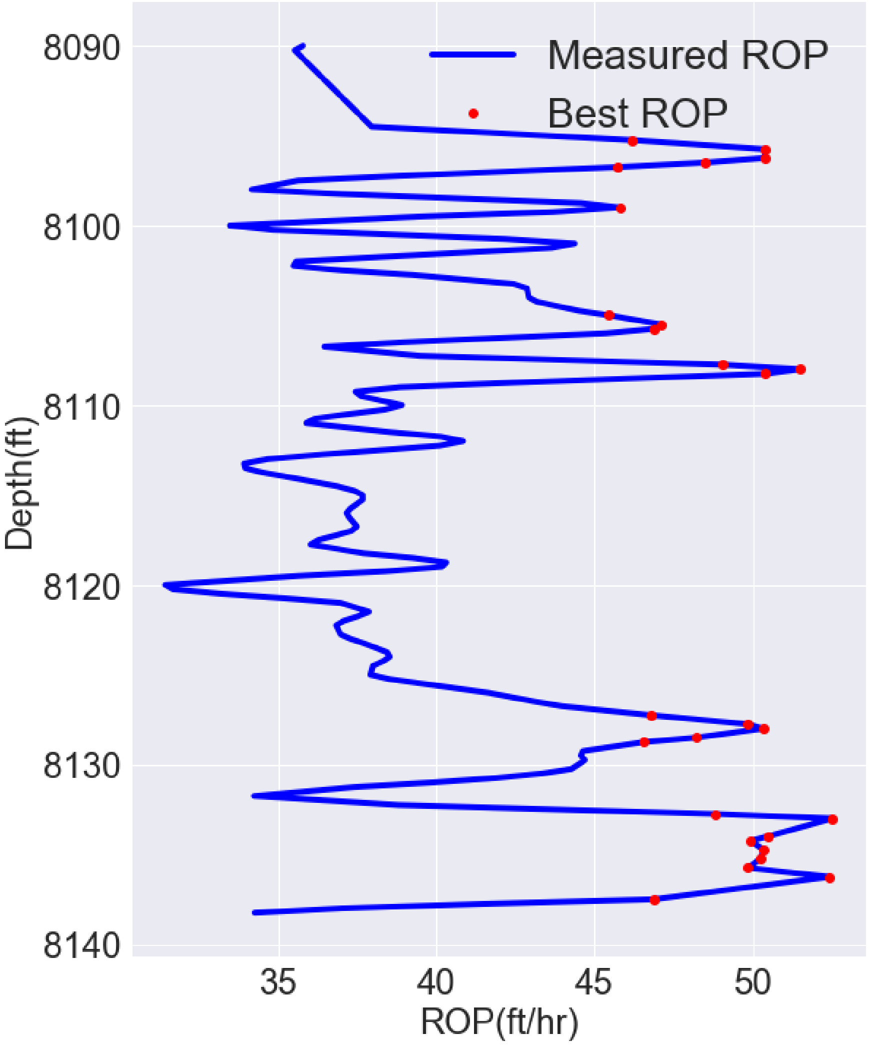

Figure 4.3: Eyeball method schematic: picking 90th percentile of ROP; Data collected by varying different parameters are used to gain an intuition of the well; The parameters which led to the best ROP in the training data can be used ahead of the bit for improved ROP.

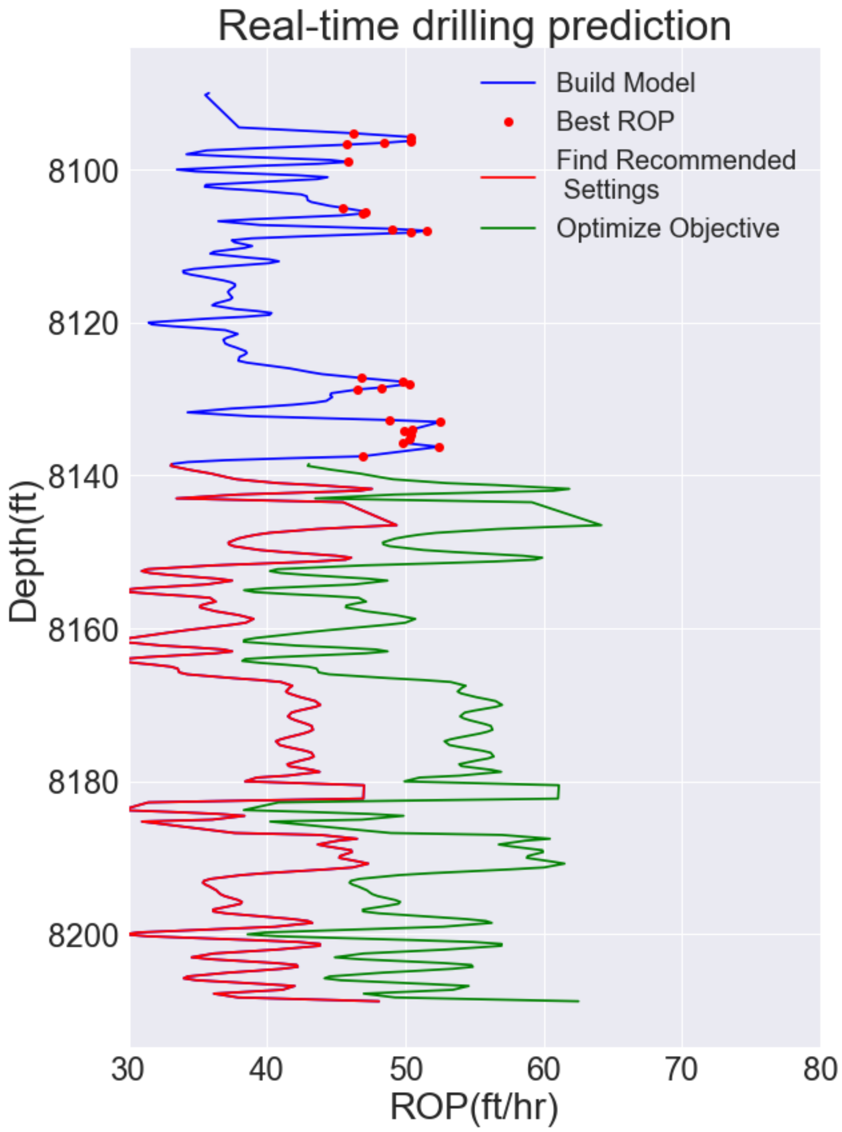

Figure 4.5: Application of eyeball method schematic on test data ahead of the bit. The data colored in blue are the training data: data collected as soon as we enter a new formation for the purposes of modeling. The data ahead of the bit in red is the original data observed while drilling without any optimization. The data colored in green is the improved ROP when optimization is employed. The green data plotted here is a hypothetical case.

4.2.2.4 Random search (Latin hypercube initialization)

Random search is the simplest meta-heuristic algorithm. Random samples from the solution space are evaluated. If the solution space is heavily sampled, this algorithm will approach the brute force solution and find a global optimum. The sampling method is important because it determines the samples evaluated. Latin hypercube sampling is a technique of sampling which is useful for sampling high dimensional spaces. It guarantees better coverage of high dimensional (>3 dimensions) spaces as compared to random or Monte Carlo based sampling. Latin hypercube sampling ensures that samples will provide information when some variables are dominant over others (Stein, 1987).

To perform the LHS, the cumulative probability (100%) is divided into multiple segments, one for each iteration. A probability is randomly picked within each segment using a uniform distribution, and this is mapped to the correct representative value in of the variable's actual distribution. For example, a simulation with 1000 iterations would split the probability into 1000 segments, each representing 0.1% of the total distribution. One sample is chosen from each segment. This is repeated for each variable. After all the variables have been sampled, variables are randomly grouped, ensuring that each variable's value is used only once. Pseudo code for random search has been described in Appendix. The random search algorithm builds on the eyeball algorithm; rather than just limiting the evaluation to explored values, the random search algorithm searches the optimization space more thoroughly.

4.2.2.5 Simplex (or Amoeba) method

The simplex algorithm is commonly called the amoeba method or Nelder-Mead algorithm. It is named after its creators (Nelder & Mead, 1965) who modified the original work of Dantzig (circa 1945) to make it better suitable for non-linear optimization. The amoeba method uses three solutions (or three random samples) and labels them best, intermediate and worst based on the response evaluation of each sample.

At each step, it attempts to replace the worst solution with a new, better solution among three candidates: a reflected point, an expanded point and a contracted point (as shown in Figure 4.6). Each of these new samples lies along a line from the worst point through the centroid of the simplex (a point that is in the middle of all points except the worst point). The best solution among the reflected, contracted, and expanded point is chosen. This process is repeated till convergence.

If neither the reflected point, nor the expanded point, nor the contracted point is better than the current worst solution, the amoeba shrinks itself by moving all points (except for the best point) halfway toward the best point. The algorithm terminates when the vertices are close to each other in value, and the best vertex is considered the optimal solution. Pseudo code for simplex has been described in Appendix.

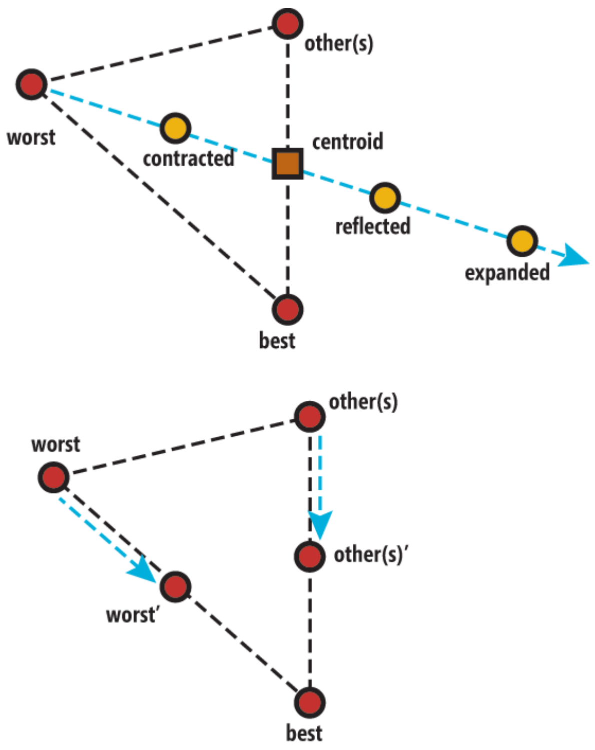

Figure 4.6: Visualization of the amoeba method for optimization of a data-driven model

4.2.2.6 Differential evolution

A subset of the popular genetic algorithms, a population-based method: many random solutions (called a population) are initialized. Each solution is evaluated for its fitness (how optimal it is) before it is discarded or retained in the next iteration. The retained solutions are tweaked (variables are changed slightly) based on certain rules borrowed from evolutionary biology. The tweaked solutions are then added to the population. This methodology is iterated until convergence.

Differential evolution (Storn & Price, 1997) is a variant of genetic algorithms with two major changes. First, the tweaked solutions are compared to their original form (from which they were tweaked), and the better solution is retained. Second, the number of samples tweaked in each round is calculated based on the variance of the initialized solutions (or population). The second major change is advantageous since the tweaking will be large when the variance is high; otherwise, it will remain low, making it an adaptive genetic algorithm.

The tweaking of variables employs vector calculations in an N-dimensional space (where N is the number of drilling parameters being optimized). There are multiple tweaking operations (commonly called a mutation in optimization circles) that can be performed on variables. For example, pick three random points ($\vec{a}$, $\vec{b}$, $\vec{c}$) in the population and tweak away from one of them ($\vec{a}$) by performing vector addition on the other two points ($\vec{a}$+ $\vec{b}$-$\vec{c}$). The tweaked point is compared to a member of the population to retain the better solution. This process is repeated until convergence. The locations of new points are based on the locations of the previous points; however, the vector addition and subtraction allow exploration of the global solution space. Pseudo code for differential evolution has been described in the Appendix.

4.2.2.7 Particle swarm optimization (PSO)

This algorithm is a stochastic optimization technique modeled after swarming or flocking of animals (Kennedy, 2011). This is like the simplex algorithm; however, more than three samples are used, and each sample is called a particle. PSO operates well in a multi-dimensional setting where at each iteration the solution is tweaked towards the current best solution.

Each particle in the PSO has a location and velocity. The velocity determines the direction and speed of travel at the next time step. Put another way, if $x(t-1)$ and $x(t)$ are the locations in space of the particle at times t-1 and t, then at time t: $v = x(t) - x(t-1)$. Each particle starts off with a random location and a random velocity.

At each time step, the velocity vectors of each particle are updated based on the global optimum discovered till that time step. The velocity vector is modified to point at a specified magnitude towards to global minimum. The particle is tweaked based on the velocity vector and some noise. The pseudocode for this algorithm has been described in the Appendix.

4.3 RESULTS AND DISCUSSION

ROP models built using equation-based methods as well as data-driven methods have been evaluated for optimal control parameters. In both cases, models were built individually on each formation with half the dataset used for training and the rest for testing (as explained in Chapter 2).

The feasibility of an algorithm to be used in real-time drilling depends on the accuracy of the algorithm in finding the global optimum ROP. Each algorithm has been evaluated for an increase in ROP and the run-time is noted. The runtime refers to the time for execution of optimization running on an 8 GB DDR3 RAM laptop with Intel 7thgeneration i7 7500U CPU @ 2.60 GHz processor running python's scipy package (Jones et al., 2001). Runtime for algorithms is important since it directly relates to the amount of data-lag of the ROP model. With a fast algorithm, a continuously updating ROP model can be utilized which is built using the most recent data collected as opposed to a ROP model built using older data or a model with "data-lag". This has been addressed further in section 4.4.1.

Each optimization algorithm was used to find optimal settings of drilling parameters to obtain maximum ROP ahead of the bit (test set). Figure 4.7 is a schematic of the entire optimization process. The top left panel in Figure 4.7 is a density plot of WOB in the training set encountered in Lodgepole formation. The panel shows the WOB used in the training set(blue), as well as the recommended weight-on-bit (orange and green) based on the employed algorithm's recommendation. The recommended WOB values are suggested values (for the driller or engineer by the algorithm) which if used would result in an improvement of ROP ahead of the bit. RPM and flow-rate settings have been plotted in a similar manner in the top right and bottom left panels in Figure 4.7. The bottom right panel describes the predicted change in ROP based on implementing the recommended settings for each input drilling parameter. The density plot in blue is the current ROP, in green and orange are the improvements in ROP that can be expected. Figure 4.8 shows the effect of using the recommended drilling parameter settings ahead of the bit. The new (or simulated) ROP based on optimal drilling parameters for the test set was calculated using the data-driven model (hereby referred to as the simulated test results). The simulated test results are compared to the original ROP to evaluate its improvement. A higher increase in ROP would indicate a better solution, making the algorithm more accurate (to the actual global maximum). Table 4.1 summarizes ROP improvement and computational run-time statistics of the evaluated algorithms. It is important to note that in Figure 4.8, the predicted ROP improvement has some variance (it is not a straight line). This randomness comes from the UCS values of the data. ROP was modeled as a function of four drilling control parameters – RPM, flow-rate, WOB, and UCS. Only three parameters are controllable on the rig; UCS remains unchanged. The continuously changing value of UCS over the formation causes the predicted improvement in ROP to exhibit some variance. If ROP was modeled as a function of RPM, flow-rate, and WOB only, the predicted improvement in ROP for the rest of the formation would be a single value, or when plotted it would be a straight line (as it is for the case of Bingham and BY Models).

Table 4.1: ROP improvement and algorithmic run-time statistics

ROP Increase	**Mean**	**Median**	**Lower Quartile**	**Upper Quartile**	**Range**
Gradient Ascent (Hareland)	**0.28**	**0.184**	**0.056**	**0.464**	**0.408**
Eyeball	**0.234**	**0.192**	**0.161**	**0.2724**	**0.111**
Random Search	**0.2877**	**0.218414**	**0.197**	**0.41494**	**0.218**
Simplex	**0.3008**	**0.218366**	**0.177**	**0.3917**	**0.215**
PSO	**0.4528**	**0.259**	**0.2122**	**0.5508**	**0.339**
DE	**0.4082**	**0.275**	**0.20529**	**0.5414**	**0.336**
Runtime (s)					
Gradient Ascent (Hareland)	**<0.1**	**<0.1**	**<0.1**	**<0.1**	**<0.1**
Eyeball	**<0.1**	**<0.1**	**<0.1**	**<0.1**	**<0.1**
Random Search	**<0.1**	**<0.1**	**<0.1**	**<0.1**	**<0.1**
Simplex	**5.93**	**5.48**	**4.93**	**5.51**	**0.58**
PSO	**196.41**	**64.43**	**27.71**	**120.34**	**92.63**
DE	**18.475**	**12.06**	**11.9**	**21.73**	**9.83**

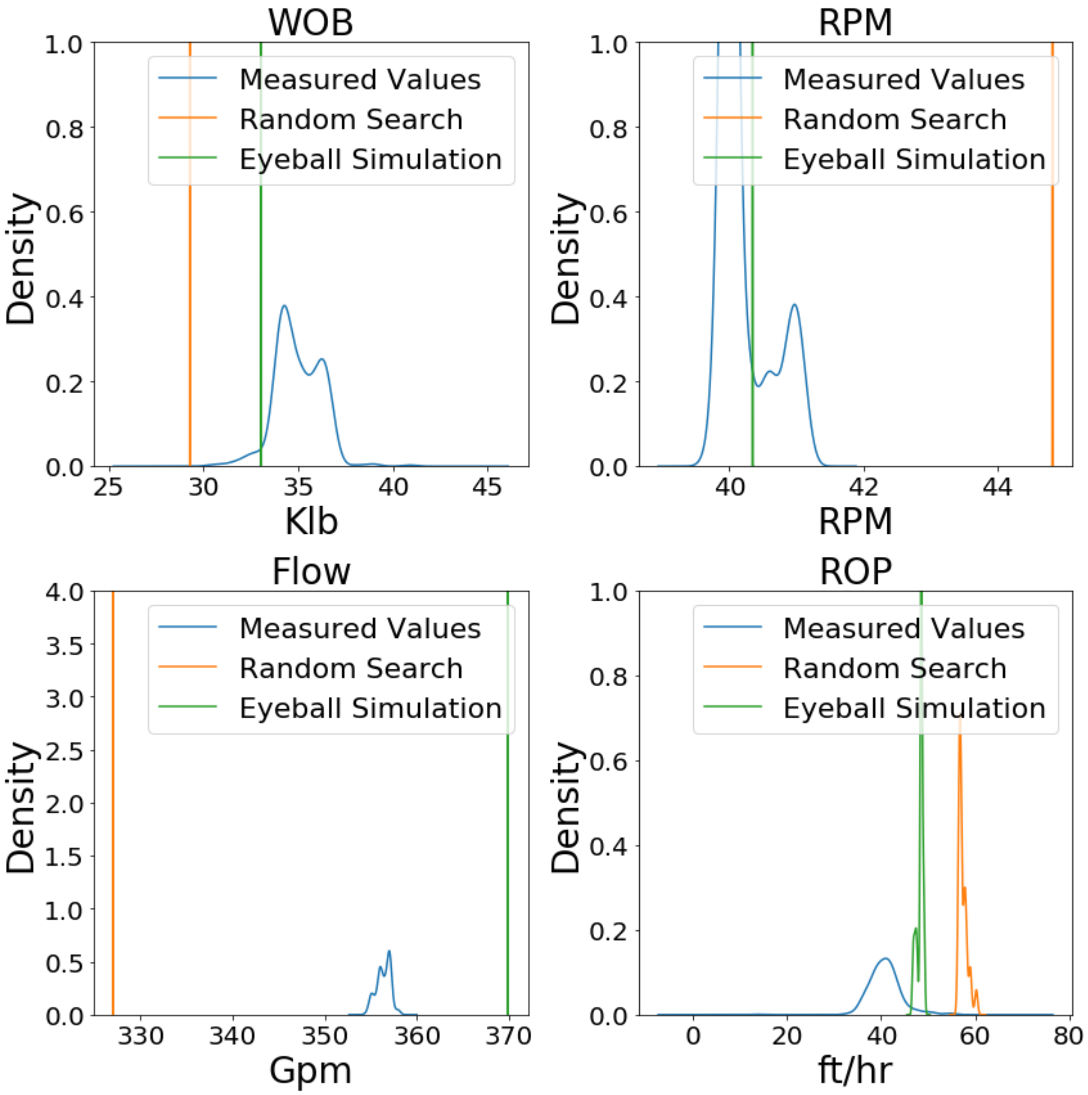

Figure 4.7: Recommended settings by eyeball and random search algorithms for lodgepole limestone formation

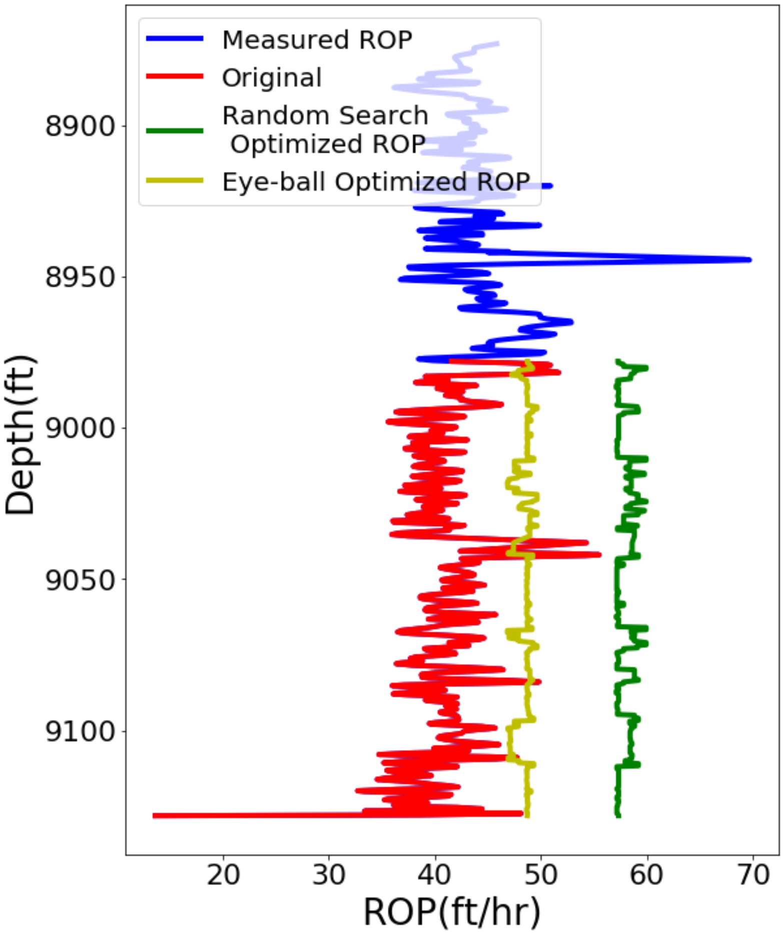

Figure 4.8: Increase in ROP based on recommended settings by eyeball and random search algorithms in the Lodgepole limestone formation

From a practical standpoint, operational parameters can be changed after drilling each stand. This will provide sufficient time – the time taken for a drilling connection – to run the algorithm to determine optimal drilling control parameters ahead of the bit. The complexity of random search and eyeball methods are linear. Simplex runs in polynomial time if implemented well; DE and PSO are highly dependent on the hyper parameters such as the number of populations, number of swarms, maximum number of iterations. However, with good implementation practices these can be reduced to polynomial time (Luke, 2009). Meta-heuristic methods have often been known to be time consuming.

Similar results have been reported while using the swam algorithm for optimizing an ROP model built using neural networks (Gandelman, 2012). Thinking about it intuitively, a ROP model built using a simple tree with 2 input parameters (or 2 dimensions) should look like a three dimensional plane (James et al., 2013). The ROP model used in the paper is more complex since it's an average of 100 randomized trees (with a randomization step in building the tree), where each tree has a block like structure but in five dimensions forming a hyperplane. Hence, navigating this space to find the best parameters will be time consuming. The time difference between the algorithms is due to the number of function evaluations (Simplex (39) << DE (137) << PSO (1075)) – analyzed using a code profiler.

The amount of time taken to run these algorithms will depend on the computational power available on the rig (if performed at the rig). An alternative option is to run these analyses using cloud computing capabilities if data are continually processed in a real time operating center. In the case of the latter, the computational run time is not an issue, and the algorithm of choice can be purely based on its performance. Computational runtime is not an issue while using gradient ascent (for equation-based models) or eyeball or random search.

4.3.1 Practical Implementation

The use of these algorithms would be primarily in an advisory mode where optimal parameters can be computed based on drilling data on the rig. Drilling operational parameters cannot be changed every second or every minute due to human constraints (such as response time of the driller), operational constraints (the changes may take time to show effect on the ROP), and unwanted oscillatory response of the system.

The frequency of drilling parameter change can be in the range of every 50-100ft of drilling or 10 minutes of drilling. A minimum of 30-40 ft of drilling or 5 minutes may be required to evaluate the changes induced in drilling speed due to drilling operational changes. The reason optimization algorithms must run computationally efficiently (or in real-time) is to maximize the amount of data used for the calculation or prevent a "data-lag". This also ensures that the latest model is computed and used for ROP analysis. Two cases are presented to help explain this concept.

4.3.1.1 Drilling parameters set every stand

After drilling one stand, the data collected is used to update and re-train the ROP model. An optimization algorithm is used to find the best parameters for drilling the next stand. In this case, 120 seconds exist (say, the connection time plus any overhead in data collection and processing) to re-calibrate the model and run the optimization algorithm to obtain real-time parameters. If the calculations are performed on the rig, simplex or random search may be used to calculate the best parameters. If the results are calculated in an offsite real-time processing center or using cloud computing, the particle swarm algorithm can be used to determine the best operating parameters. These operating parameters are then used to drill the next stand, and this process is repeated until the entire formation is drilled.

4.3.1.2 Drilling parameters changed on-the-fly or when drilling is too slow

Data is being collected real time during drilling. Assume the time when drilling the new stand was 12:00:00 PM GMT and the average ROP was 60 ft/hr. After drilling for a period of 15 mins (current time: 12:15:00 PM GMT; 15 ft drilled), the parameters are re-evaluated to check for changes. In this case, if the PSO algorithm is used, and the PSO algorithm takes 240s of run time (including any overhead costs), the data used for such a computation will lag by 4 minutes, i.e. only the first 11 minutes of real-time data will be used to make any suggestions. However, if the simplex method were used instead, the data-lag would be around 0.5 minutes. These run-times correspond to the use of a fast computer (or a cloud processing unit). If an on-site rig computer is used, depending on the processing time this value can range from 10x-200x, given the complexity of the algorithm. In this case, using PSO would not be recommended since the operational parameters are lagging several hundreds of feet and cannot be considered real-time.

4.3.2 Traditional ROP Models

Traditional models are optimized using the gradient ascent algorithm. The data are whitened to a mean of 0 and standard deviation of 1; this ensures that the problem is well conditioned (Ng, 2000). A learning rate of 0.01 was used to successfully obtain stable solutions for all four equation-based models. The optimal parameters were used in the corresponding traditional model to predict the ROP ahead of the bit (on the test set).

Results of using improved drilling control parameters are shown in the form of a box-plot in Figure 4.9. The mean ROP before optimization was 50 ft/hr for the entire well; by implementing optimal parameters, the predicted improved average ROP: 77 ft/hr for Bingham's model, 75 ft/hr for Motahhari's model, 64 ft/hr for Hareland's model, and 74 ft/hr for BY model. These are very encouraging results; however, the assumptions of the model must be taken into account. It has been shown that increasing WOB indefinitely does not result in an increase in ROP due to an inefficient cleaning of cuttings at the bit (called floundering of rock (Maurer, 1962)). In the case of Bingham's model, the result shown in Figure 4.9 is obtained by implementing the argmax, which includes the maximum WOB permissible. This result may not be an accurate representation of the improvement of ROP since the prediction of ROP itself had a high error as discussed in Chapter 2. The prediction of ROP using Hareland's model was the most accurate among all equation-based model used, hence, the predicted increase in ROP and optimal parameters returned with this model would be most trustworthy of actual improvements in ROP.

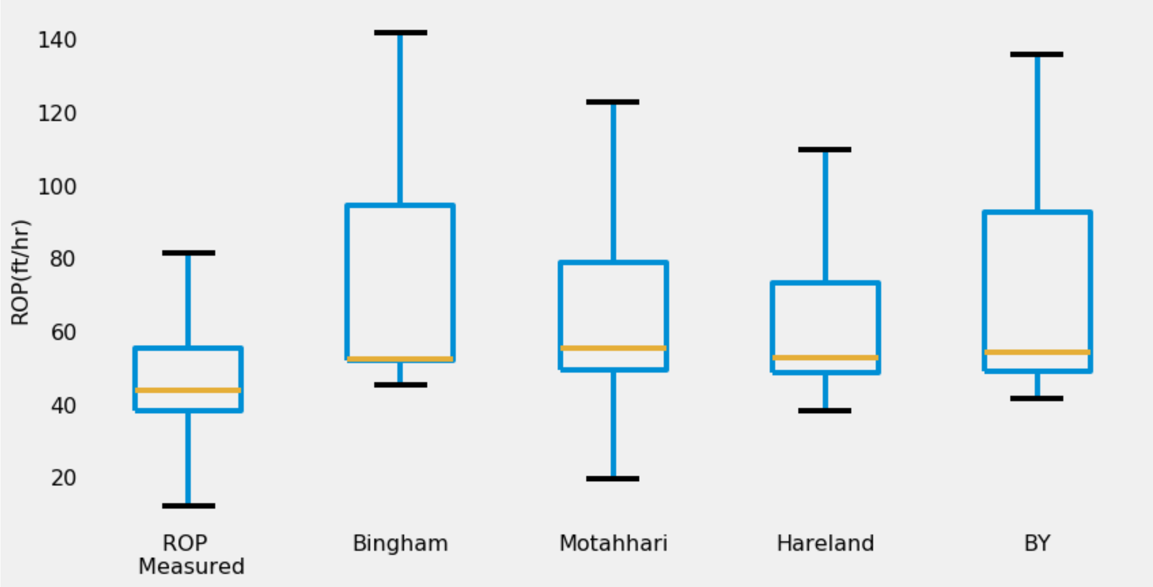

Figure 4.9: Predicted ROP on the test set implementing optimal parameters obtained using the gradient ascent optimization algorithm.

These results have been presented anyway since the methodology behind their implementation is important. In cases where these models do provide an accurate representation of drilling, the use of the gradient ascent algorithm to find optimal drilling parameters and implement them ahead of the bit will result in increased ROP. The change in ROP in each formation has been plotted in Figure 4.10.

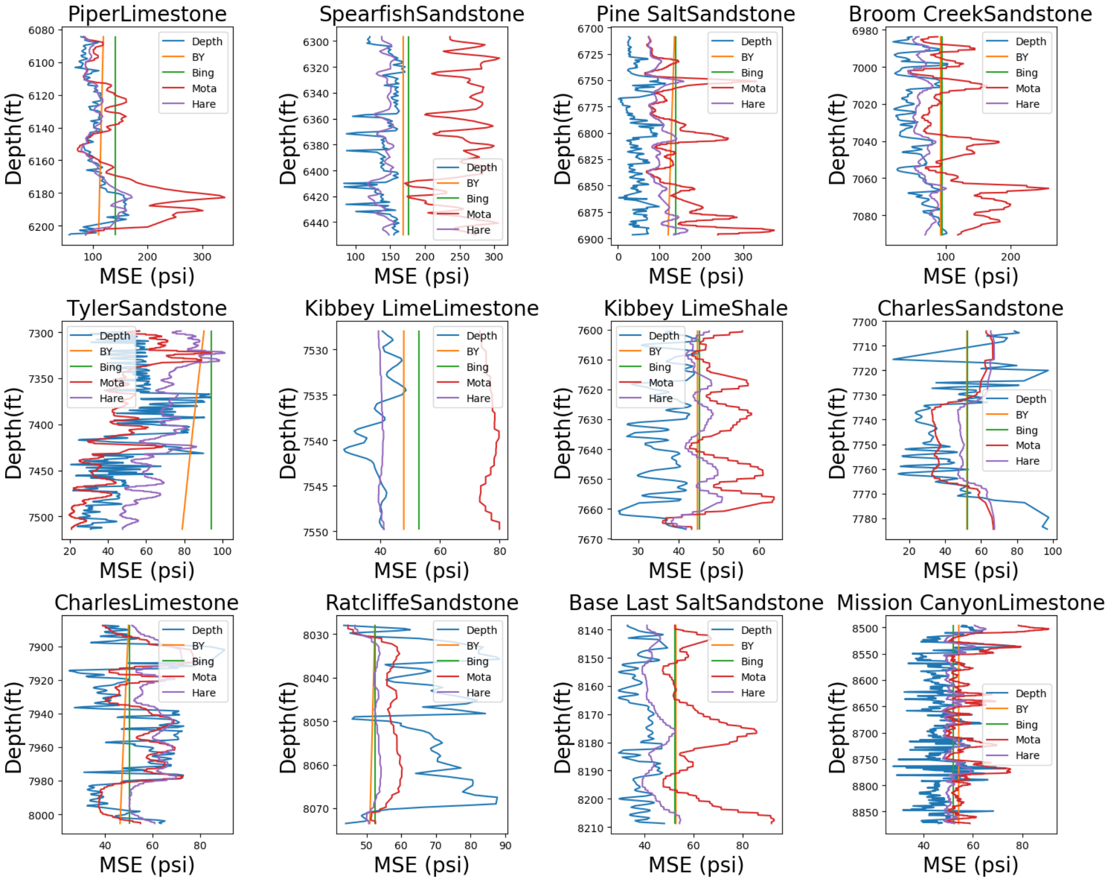

Figure 4.10: ROP maximization using equation-based ROP models; Optimal drilling control parameters are determined using the gradient ascent algorithm and implemented on the test set; Predicted ROP is returned and shown in the form of a line plot.

4.3.3 Data-driven ROP Models

In the case of data-driven ROP models, different algorithms are evaluated based on the ROP improvement (the computational runtime is noted as well). Figure 4.11 shows a line and boxplot comparisons of the simple optimization (random search and eyeball) algorithms. The run-time has not been plotted since it was trivial (<100 ms). A total of 500 samples were drawn using the LHS algorithm and evaluated in the random search.The line plots show that random search performs marginally better than the eyeball method. Box plot comparison shows an increased ROP with the random search. The eyeball method only samples from the best points; this may result in a situation where a portion of the sample space which is more optimal is missed. LHS has better sampling regime – it ensures that the entire solution space is sampled. The improved performance of LHS can be attributed to a more robust sampling and a broader range of the solution space.

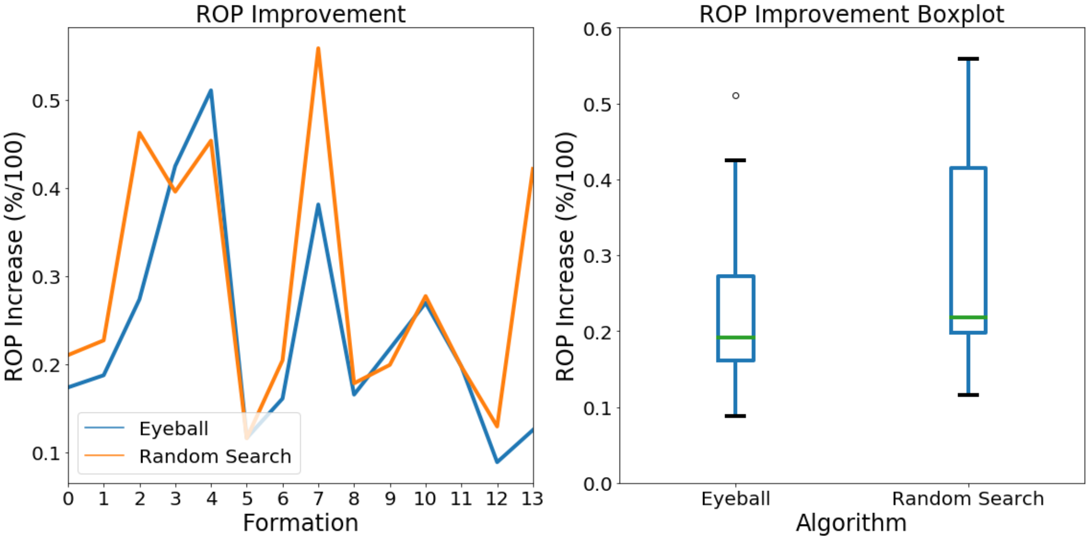

Figure 4.11: Comparison of ROP optimization for data-driven models using eyeball and random search algorithms

Figure 4.12 shows a line and boxplot comparison of the advanced meta-heuristic optimization algorithms. The tolerance of meta-heuristic algorithms was set to 1 ft/hr– if an improvement of atleast 1 ft/hr of ROP is not achieved in 100 iterations the algorithm is terminated. The tolerance is based on practicality if the solution. Differential evolution was implemented with an initial population size of 30, mutation (a proxy for search radius) of [0.5,1), recombination constant (probability of producing an offspring) of 0.7, and the population was initialized using the latin hypercube algorithm. The particle swarm algorithm was initialized with a population of 100, velocity scaling of 0.5, and scaling to search away from the best location of 0.3 implemented using the pyswarm package in python (Miranda, 2018). The hyperparameters for the meta-heuristic algorithms were calculated using manual tuning. A grid-based search was used to calculate the best hyper parameters for each algorithm using the normalized increase in ROP as well as the computation runtime as metrics for evaluation.

Based on the line plots (Figure 4.12 left top) for ROP increase it can be concluded that PSO > DE > simplex. The difference in median ROP increase between PSO and simplex is ~ 4% based on the box plots (Figure 4.12 left bottom). The computational run-time for these algorithms are in the same order as ROP increase with PSO >> DE >> simplex (Figure 4.12 right top and bottom). Simplex algorithm can be executed fast enough to avoid major effects of data-lag (this depends on the available computational resources on the rig). Optimizing ROP for higher depth does not change the functional form of the ROP model being optimized, hence making the algorithmic run-time invariant. If the model is retrained with additional data, as drilling proceeds, the run-time would depend on the complexity of the new ROP model.

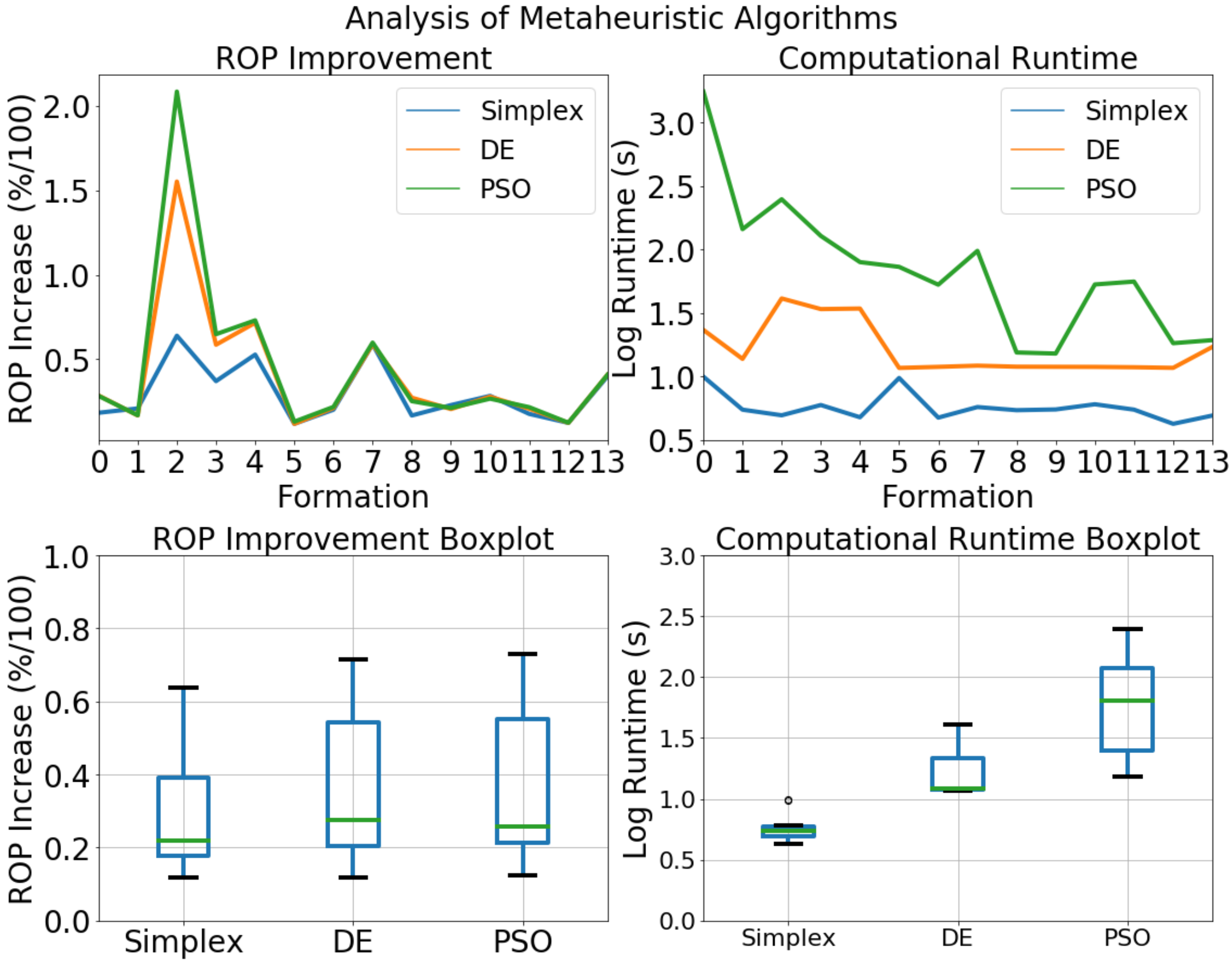

Figure 4.12: Comparison of advanced optimization algorithms ROP increase and computational run-time (logarithm of base 10 has been used in the plot)

A parametric analysis was carried out to evaluate the effect of repeatedly re-training the model while drilling as shown in Figure 4.13. Changing the length of the training set will change the ROP model and expand the solution space. Since the solution space increases with continuously increasing training set, both computational time, as well as ROP improvement, may change.

The algorithmic runtime and increase have been plotted for optimizing different models by re-training the model 5 times within a formation in Figure 4.13. New ROP models were built at different training set lengths (0.4,0.5,0.6,0.7,0.8) for each formation. The box plots in Figure 4.13 are similar to Figure 4.12, indicating algorithmic runtime is invariant of training set sizes for this dataset (logarithm of base 10 has been used in the plot).

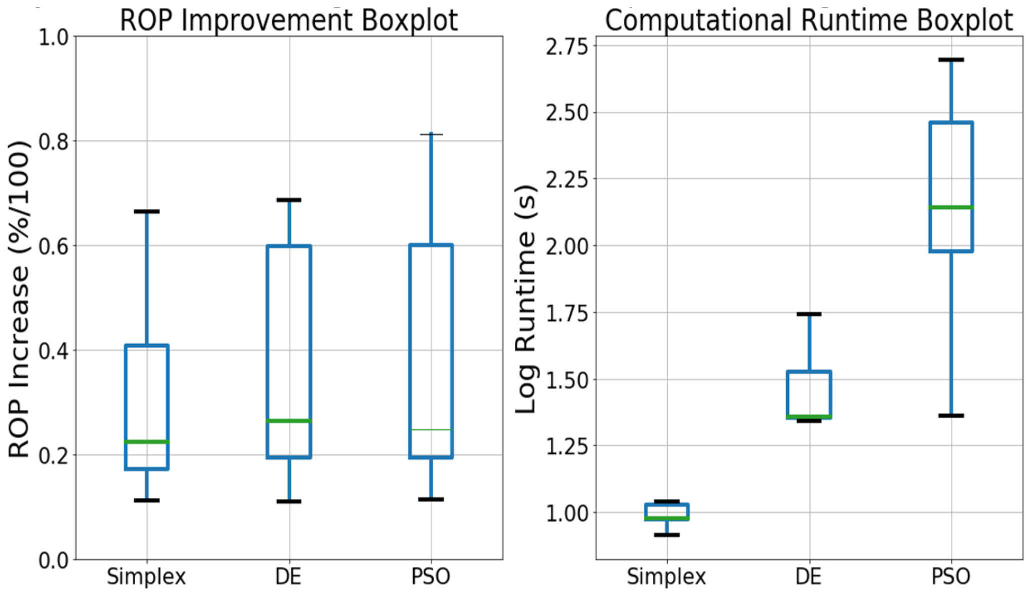

Figure 4.13: Analysis of algorithm performance for varying the length of the training set or continuous model updating during drilling (logarithm of base 10 has been used)

4.3.4 Uncertainty Analysis

Uncertainty analysis of the expected increase in ROP for each formation evaluated can be calculated for the data-driven ROP models given the probabilistic nature of the models and algorithms used. The equation-based ROP models are deterministic – an uncertainty interval is not possible. The optimization results in the previous section described the improvement of ROP on 14 different formations. In this section, an uncertainty analysis is conducted to calculate the 95% confidence interval for ROP improvement on the test set. The 95% of the time, the range defined by this (confidence) interval will contain the possible improvement in ROP. This analysis is commonly practiced in reservoir engineering to aid the decision-making processes.

In each simulation, the training and test set are changed by randomly choosing data points to act as the test and training set respectively. The test-train split – the ratio of training set length to test set length – is kept fixed at 0.5. At each iteration, 50% of the data points selected in a formation are chosen to be part of the training set, and the rest will be a part of the test set. 1000 Monte Carlo based simulations are evaluated to calculate the 95% confidence interval. Each time a model is built on the training set, optimization algorithms are used to evaluate the best drilling parameters, and these drilling parameters are used in a drilling simulation to test the improvement in ROP. The projected improvement in ROP for the eyeball method and LHS random search algorithm has been plotted in Figure 4.14. Results show that ROP optimization based on the eyeball method and LHS random search is consistent with the previous simulation shown in Figure 4.11. An average ROP improvement of 22.5% and 32% can be expected from the eyeball and random search simulations. The projected improvement in ROP for the simplex, particle swarm, and differential evolution algorithms are plotted in Figure 4.15. Results parallel those seen in Figure 4.12 with some variance around the ROP improvement in each formation. Tabulated statistics are shown in Table 4.2. A low-variance is indicative of a consistent algorithm.

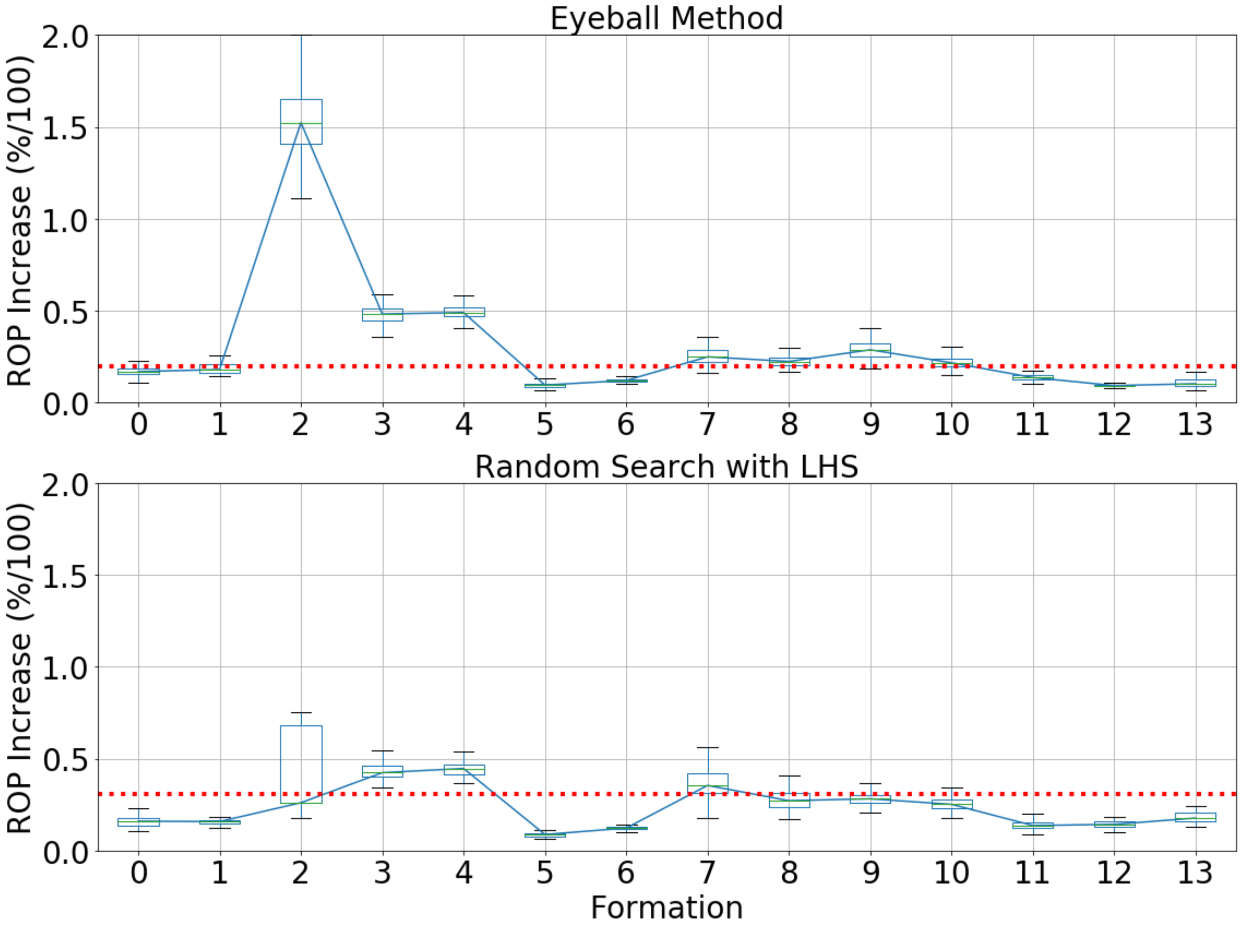

Figure 4.14: 95% confidence interval for ROP improvement using eyeball method and LHS random search based on 1000 Monte Carlo simulations. A line plot joins the medians of each box-plot. The red dotted line corresponds to the average improvement in ROP for all the formation over the length of the well.

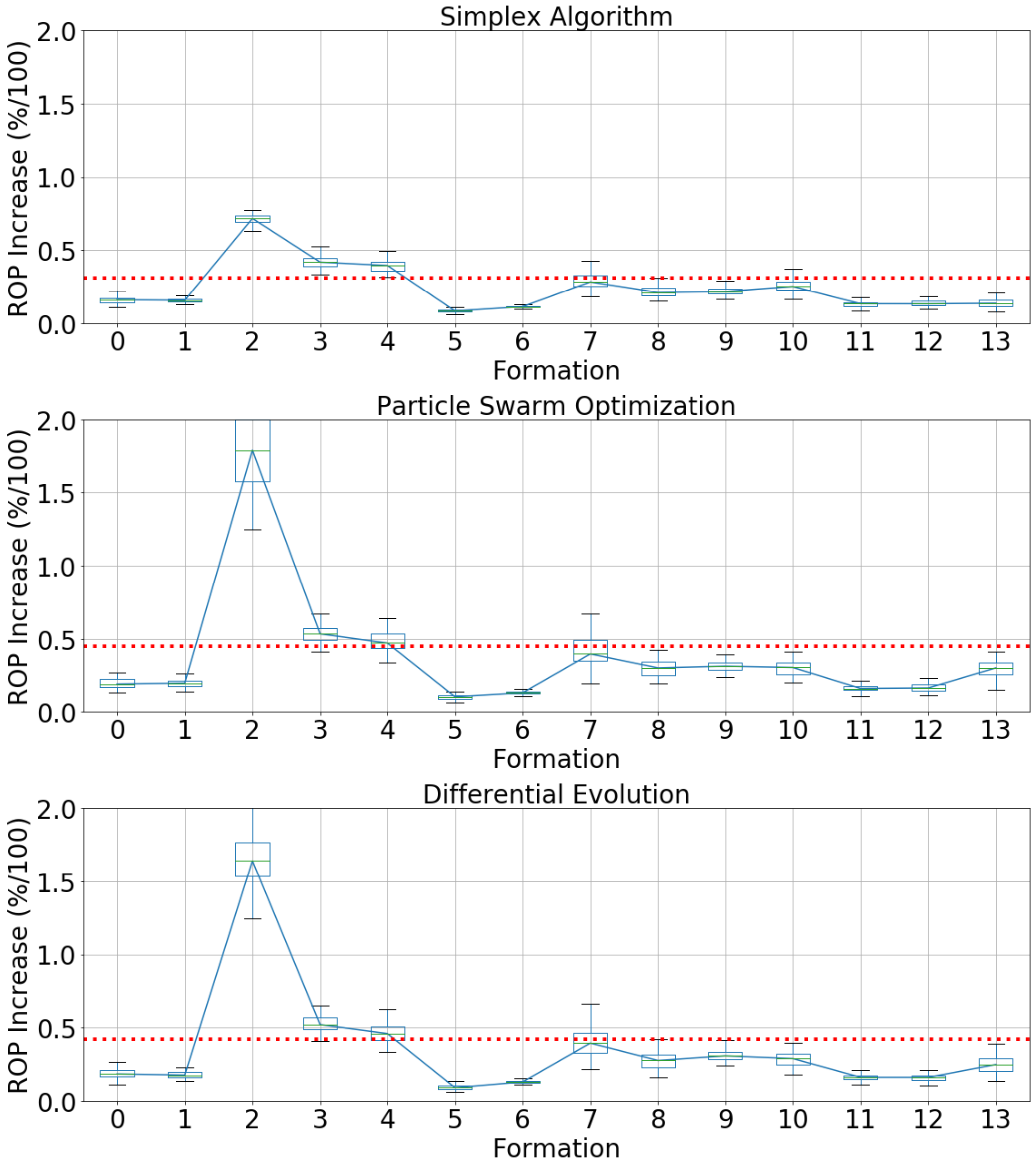

Figure 4.15: 95% confidence interval for ROP improvement using eyeball method and LHS random search based on 1000 Monte Carlo simulations. A line plot joins the median of each box plot for all formations. The line plot can be compared to the line plots in Figure 4.12 to evaluate consistency in the ROP optimization with each algorithm. The red dotted line corresponds to the average improvement in ROP for all the formation over the length of the well.

Table 4.2: Statistics for Monte Carlo uncertainty analysis for ROP optimization simulations

	Eyeball	LHS	Simplex	PSO	DE
Mean	0.225	0.32	0.3089	0.44	0.42
Variance	0.125	0.0525	0.024	0.17	0.14

4.3.5 Why do data-driven algorithms work?

Following the discussion of the convexity of equation-based ROP models, it is interesting to note that they are even monotone increasing, in most cases: if you increase the WOB and the bit rotational speed, the ROP will forever increase. This has been shown to be a bad method of modeling ROP since it does not account for rock floundering and bit cleaning effect. Too high a WOB may result in phase III where inefficient cleaning of broken rock at the bit will hinder the ROP from increasing (Detournay, Richard, & Shepherd, 2008). Ideally, during drilling, ROP should behave in a manner that can be modeled by a convex function. However, the wear of the drilling bit, vibrations, borehole assembly (BHA) may modify this response and make it more complicated. ROP, WOB and RPM data have been plotted in Figure 4.16 to better understand the relationship between these drilling parameters. The scatter shown in Figure 4.16 does not follow an easily observable trend, making it difficult to fit a convex function well. This is precisely why traditional ROP models have lower accuracy when compared with data-driven ROP models. It is difficult to fit ROP data when the function to be fit is highly constrained, especially when the drilling conditions makes the ROP behave in a nonconvex manner. Data-driven models have performed better because of their ability to represent non-linear data more effectively as compared to traditional functions forms (like polynomials or exponential functions). The scatter shown in Figure 4.16 can be fitted with a convex function with high error. A data-driven model can fit the scatter more effectively, however, the fitted function will not be convex since the data does not follow a convex trend.

Looking at the first two windows in Figure 4.16 it is possible to gain an intuition of the success of the data-driven models. The plot in the first window shows a scatter of ROP, WOB, and RPM. Suppose Bingham's model is fitted to this data – the model is a power law model and would have an appearance similar to Figure 4.2 – it would not fit well. The model would attempt to fit a smooth plane curve to the data, which would result in a lower accuracy. Looking at the second window, it's possible to discern a paraboloid which would be fitted well by a power law model. Equation-based models work well when such situations are encountered where the data follow convex functions and can be fitted easily. However, in most other cases as seen in Figure 4.16, the scatter cannot be fitted well by a power law model or a smooth convex surface. The random forest model (which would appear similar to Figure 2.8 yields a better fit. It is pretty obvious why: in its simplest form, the algorithm partitions the scatter into smaller manageable regions and returns the average of this scatter as the ROP prediction. The splitting of data into small regions, and fitting these small regions individually yields an accurate ROP predictor. On the surface this is prone to overfitting (since the scatter is split into small regions); this is combated by the averaging of trees in a random forest. Chapter 2 covers a thorough error analysis on modeling ROP data using data-driven models as opposed to traditional models.

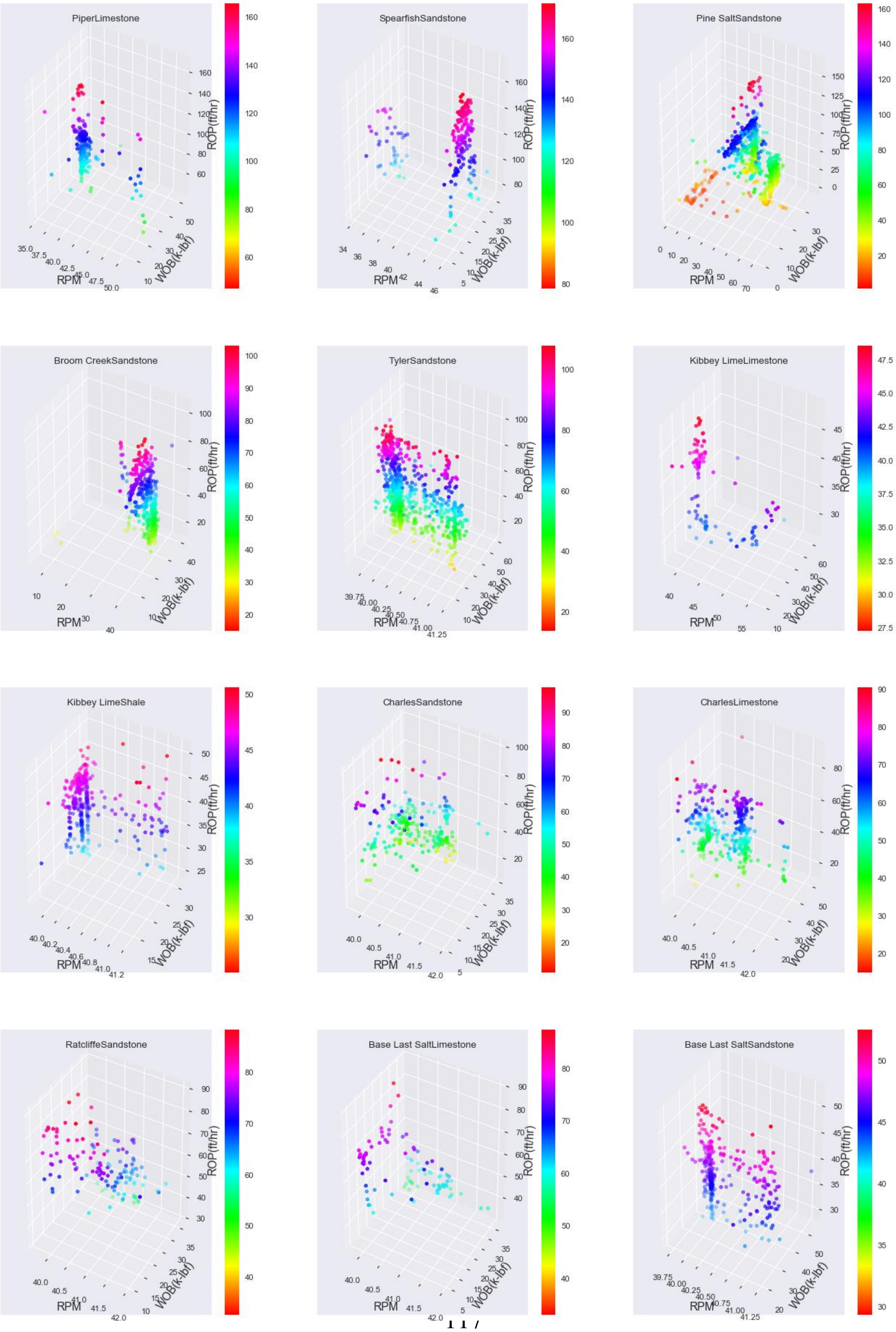
PiperLimestone
RPM
WOB(k-lbf)
ROP(ft/hr)
SpearfishSandstone
RPM
WOB(k-lbf)
ROP(ft/hr)
Pine SaltSandstone
RPM
WOB(k-lbf)
ROP(ft/hr)
Broom CreekSandstone
RPM
WOB(k-lbf)
ROP(ft/hr)
TylerSandstone
RPM
WOB(k-lbf)
ROP(ft/hr)
Kibbey LimeLimestone
RPM
WOB(k-lbf)
ROP(ft/hr)
Kibbey LimeShale
RPM
WOB(k-lbf)
ROP(ft/hr)
CharlesSandstone
RPM
WOB(k-lbf)
ROP(ft/hr)
CharlesLimestone
RPM
WOB(k-lbf)
ROP(ft/hr)
RatcliffeSandstone
RPM
WOB(k-lbf)
ROP(ft/hr)
Base Last SaltLimestone
RPM
WOB(k-lbf)
ROP(ft/hr)
Base Last SaltSandstone
RPM
WOB(k-lbf)
ROP(ft/hr)

Figure 4.16: Three-dimensional scatter of ROP, WOB, and RPM data in 12 different formations. Plots indicate that ROP relationship within each formation is non-linear and non-convex

4.3.6 Effect of ROP model on optimization

The algorithm used to determine the optimal parameters are crucial, however, the model used for ROP prediction itself affects this result. Recalling that traditional models are functionally power law models they may not be able to capture complex coupling, unlike the data-driven models. An example is the effect of rock floundering, where an increase in WOB indefinitely does not increase but leads to a decrease in ROP due to inefficient cleaning at the bit (Maurer, 1962). This has been reported and confirmed by field-based studies (F E Dupriest, Witt, & Remmert, 2005). The traditional models always predict higher ROP with increased WOB and higher RPM given their functional form. Hence, there is a chance that the optimal parameters for RPM and WOB will lie in the middle and not edges of the input parameter sample spaces – in which case using a traditional model will not lead to an ideal solution. Contour plots can be used to efficiently visualize this phenomenon and have been plotted in Figure 4.17. The contour plots (Figure 4.17) for traditional as well as data-driven models have been plotted for the Lodgepole Limestone formation in a reduced two-dimensional sample space (WOB and RPM). As expected, the traditional models predict an increase in ROP with an increase in WOB and RPM (similar to Figure 4.2). The field-data are evaluated for the maximum ROP measured and has been plotted as a red star on the plot. Based on the plot it is evident that the random forest model captures the ROP variation with changes in RPM and WOB. However, all traditional models predict an increase in ROP with increases in WOB and RPM which does not yield the best ROP and may lead to rock floundering. The contour of ROP variation for

Hareland (the best traditional model) and random forest model (the best data-driven model) have been plotted in Figures 4.18, and 4.19 for each formation.

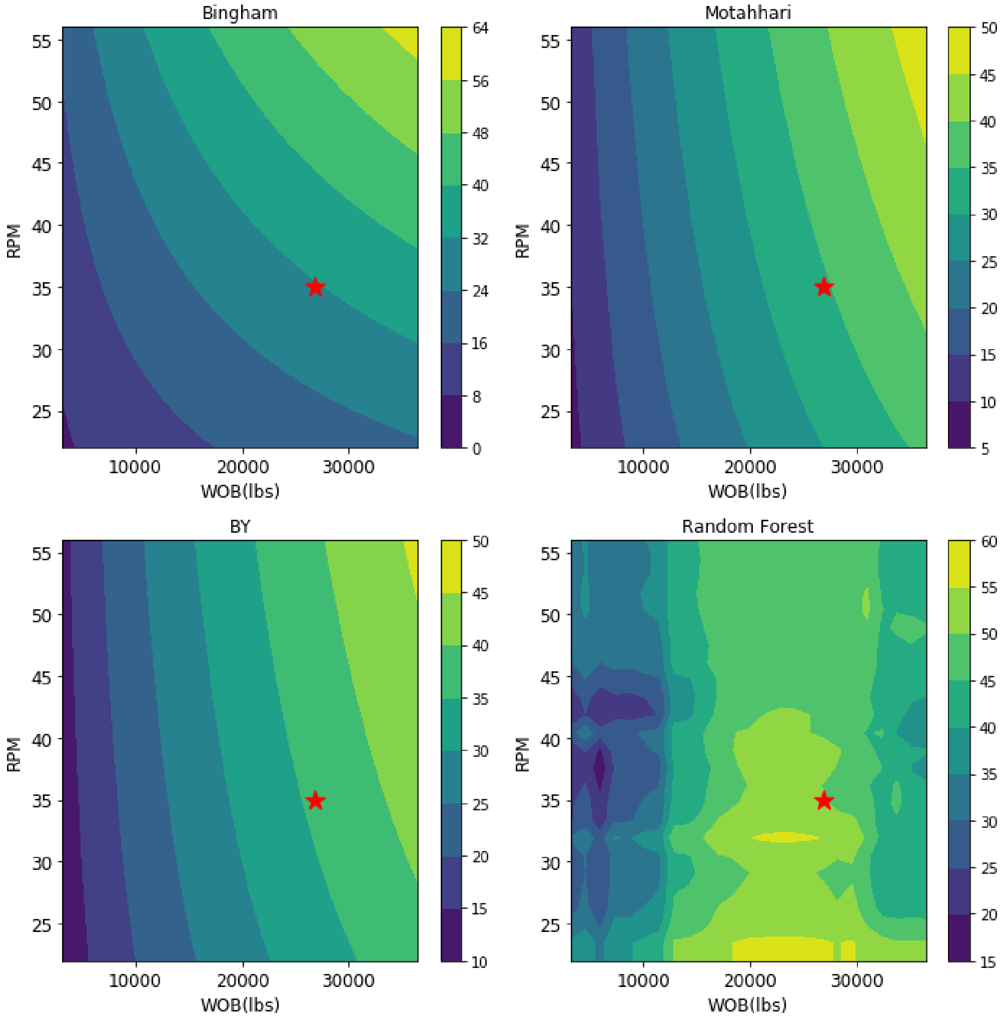

Figure 4.17: Contour plots of the WOB and RPM sample space for different ROP models – Bingham, Motahhari, BY and Random Forest. The true maximum ROP as determined by field-based measurements has been plotted as a red star. The traditional models move away from the true maximum and would lead to reduced ROP whereas the random forest more accurately captures the ROP variation

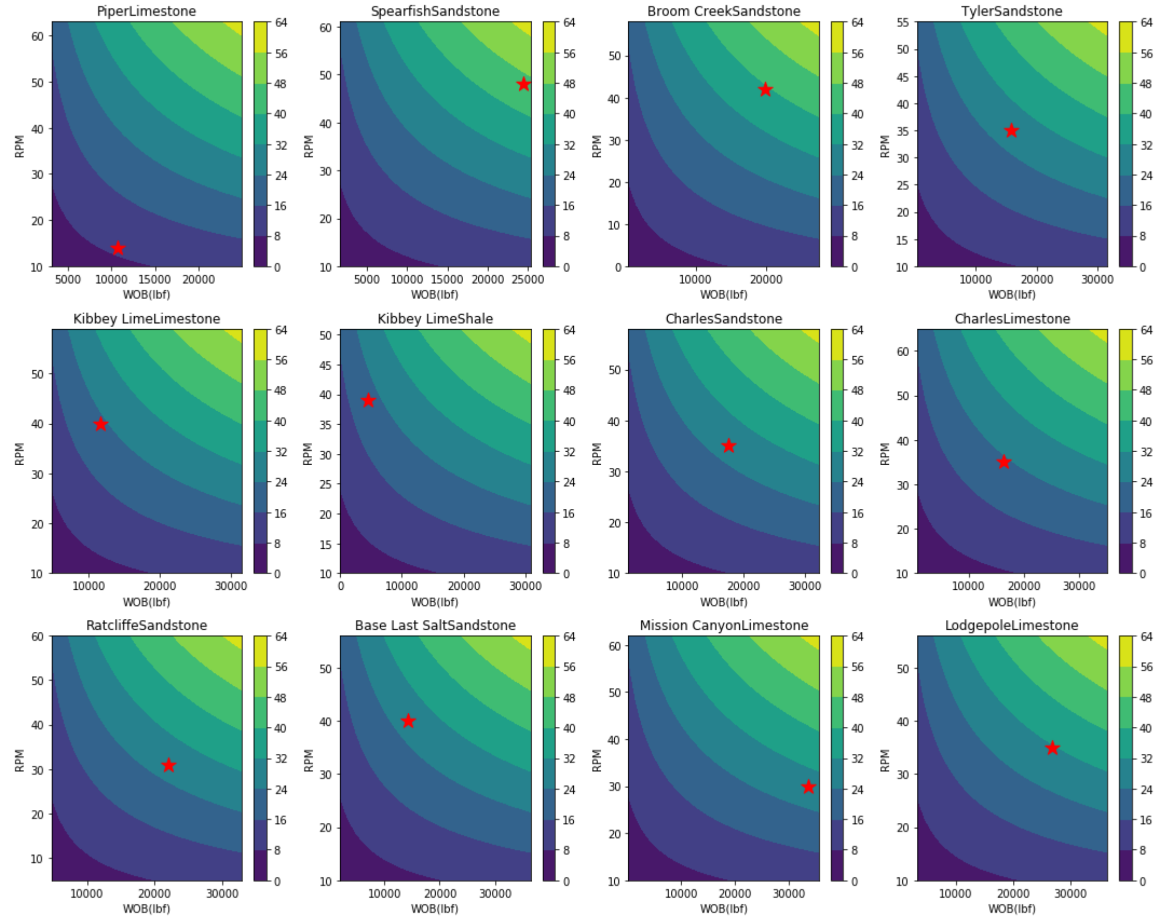

Figure 4.18: Contour plots of the WOB and RPM sample space for the Hareland ROP model for all formations. The true maximum ROP as determined by field-based measurements has been plotted as a red star.

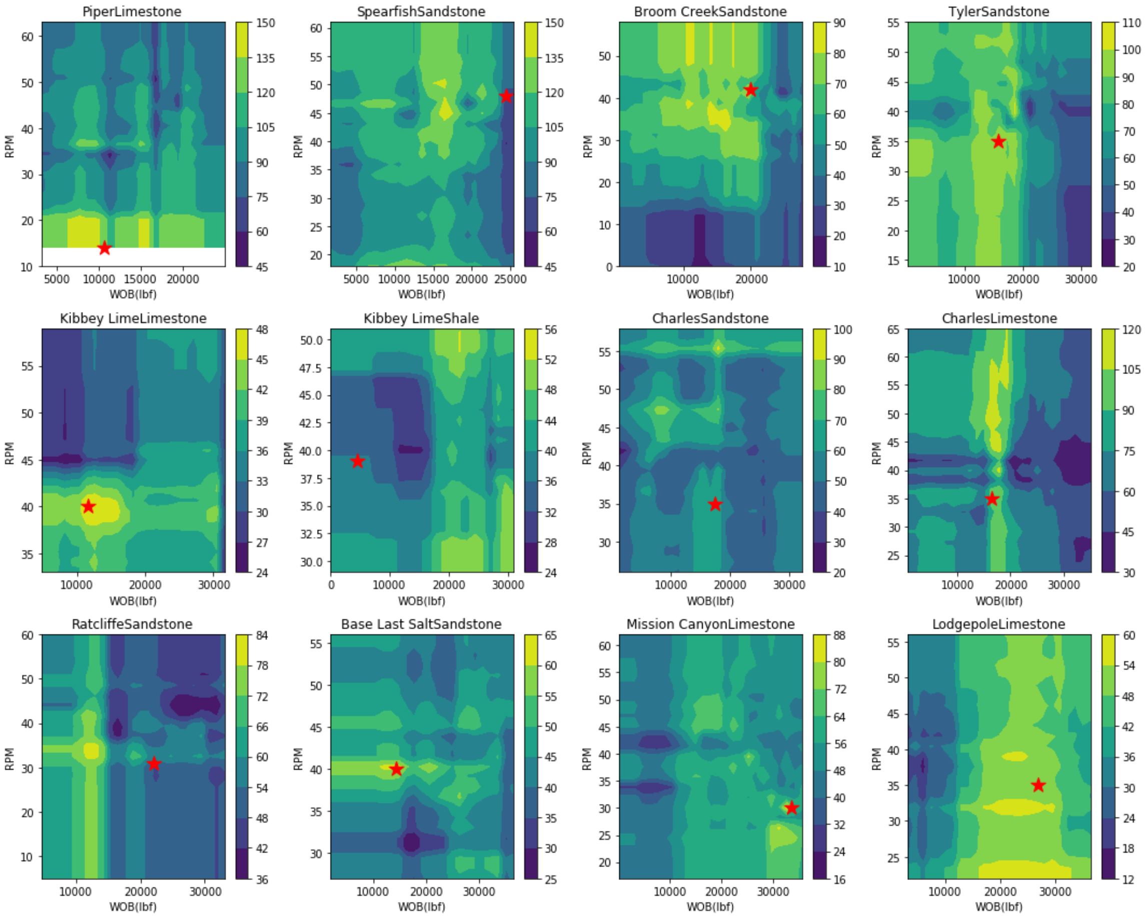

Figure 4.19: Contour plots of the WOB and RPM sample space for the Random Forest ROP model for all formations. The true maximum ROP as determined by field-based measurements has been plotted as a red star. This model does a good job of capturing the variation of ROP using input parameters.

4.4 Conclusions

Traditional or deterministic models are most often the industry standard for ROP and MSE optimization. Data-driven models are commonly more accurate than deterministic models, however, inverting them to determine optimal parameters is more difficult. The deterministic models can be functionally convex in nature (like the Bingham model) and/or smooth analytical functions making their optimization easy with the use of analytical methods or algorithms such as gradient ascent.

The optimization of traditional models showed an average improved ROP of over 28% for all models. However, the ROP predictions themselves were not accurate using these models, hence the ROP predictions using optimal values cannot be fully trusted for this dataset. Additionally, the assumptions of some models are not represented in these optimized results. If a well being drilled can be accurately modeled using traditional models, this methodology can be used to improve ROP in real-time or for drilling an adjoining well. The most accurate equation-based model for this dataset (Hareland's model) showed an average increase of 28% on the test set with the implementation of optimal drilling control parameters.

Data-driven models pose a problem for real-time optimization since their functional form is essentially unknown. This inverse problem is commonly solved using meta-heuristic algorithms. Two simple and three meta-heuristic algorithms were evaluated in this chapter based on their ROP improvement ahead of the bit (higher the better) and run-time. Simple algorithms, like the eyeball method and random search, performed well, improving the ROP by 20 and 21% on average for 14 formations tested in this chapter. Their run-time was trivial (<100 ms) making them effective algorithms for real-time use with zero data-lag. Advanced meta-heuristic algorithms used in this paper: simplex method, differential evolution and particle swarm method improved the ROP by 30%, 40% and 45% (on average). The computational run-times (mean) were 5.93 s for simplex, 18.475 s for DE and 196.41 s for PSO. These run-times can have adverse effects on the amount of data used to train the model in real-time (or data-lag). An uncertainty analysis was carried out using a Monte Carlo based simulation to calculate the 95% confidence intervals for the improvement in ROP. None of these confidence intervals contained zero: making these results statistically significant (Casella & Berger, 2002). The results of the Monte Carlo analysis bolster the claims of the proposed algorithms.

Analysis showed that data-driven models can be used for real-time drilling despite the unknown functional form. Algorithms should be carefully selected to provide a satisfactory ROP improvement ahead of the bit and run with feasible computational power to minimize the data-lag of recommendations. All algorithms evaluated in this paper worked well, however, the simplex algorithm performed with the best tradeoff. The simplex algorithm resulted in a 30% improvement in ROP ahead of the bit on average with a low variance of 2.5% and can be executed within 6 seconds. A Monte Carlo based simulation was used to calculate the 95% confidence interval of ROP optimization ahead of the bit to aid in algorithmic selection and decision-making. Algorithm selection will depend on rig conditions, operator preferences and computational feasibility which vary. This chapter can be used as a template and a starting point for algorithmic selection for data-driven model optimization.

The optimization of ROP models provides an explanation of the superior performance of data-driven ROP models. The data captured in the field are often non-increasing, non-convex functions which are difficult to model using power-law or analtyical models. The random forest algorithm captures this data more efficiently by splitting the drilling parameter space to fit small portions of the data. Additionally, the effect of using different ROP models for optimization was evaluated. Analysis through contour plots showed that the optimal parameters calculated using traditional models are often wrong and nowhere close to the actual solution measured out in the field. The random forest algorithm performs well and captures most of the variation of ROP in the WOB and RPM parameter space giving a solution which closely matches observed field data.

Chapter 5: Optimum Drilling Metrics and Parameters[5]

Drilling optimization is generally approached by optimizing ROP. Other metrics in drilling such as MSE, TOB, and cost-per-foot of drilling are not as prolific as ROP. The earlier chapters in this dissertation provided tools to model and optimize ROP. However, these tools can easily be extended to other drilling metrics such as MSE, TOB, or cost-per-foot of drilling. Structured steps towards optimization were introduced for ROP in drilling by defining an objective function, and optimizing it thereafter (Tansev, 1975). However, the functions being optimized to improve drilling performance in literature has always been ROP. ExxonMobil's drilling advisory system (DAS) is a popular system which incorporated the effect of vibrations for ROP optimization (Chang et al., 2014; Gregory S. Payette et al., 2015) moving towards the ideas laid out by Tansev (Tansev, 1975). The objective of this chapter is to investigate the effect of not only using ROP but also using torque on bit (TOB), cost-per-foot (CPF), and MSE as the objective (metric cost functions) for drilling optimization. Drilling is optimized from a mathematical perspective: choose an objective function, determine best control parameters, and evaluate their effect on drilling.

A procedure to optimize engineering operations using machine learning (using the data-driven models developed in earlier chapters) has been laid out. Models (discussed in Chapter 2) can be optimized based on drilling control parameters – drilling parameters which can be controlled or changed on the surface of the rig. Heuristic algorithms are used to determine optimal settings of control parameters (to optimize machine learning functions as discussed for ROP in Chapter 4). These algorithms are then used (on fitted models) to determine the best control parameters ahead of the bit. Different objective functions are

[5] Hegde, C. M., & Gray, K. E. (2018). Evaluation of coupled machine learning models for drilling optimization. *Journal of Natural Gas Science & Engineering*.

The author of this thesis is the primary author of the paper

evaluated to determine the effect of choosing a given objective function for drilling optimization. Model evaluation is performed by running experiments on field measured data. Results show that choosing MSE as an objective function may be more suitable than purely optimizing ROP. Only data-driven models are evaluated in this chapter since they are coupled.

5.1 Drilling Metrics

A metric (cost or objective function) represents the entity or function which is being optimized. The objective is modeled in terms of controllable parameters. This allows determining an optimal set of controllable parameters to optimize this objective. For example, when ROP is the objective function, the aim would be to maximize ROP. ROP is modeled as a function of WOB, RPM, flow-rate, and strength of rock (UCS). An optimization algorithm is used to determine optimal control parameters – WOB, RPM, flow-rate – so that ROP can be maximized ahead of the bit.

5.1.1 ROP

The ROP of drilling has long attracted researchers and engineers. It is by far the most commonly optimized metric in drilling. Since the rate directly corresponds to the time taken, improving the rate can lead to time saved. However, this metric has many disadvantages: higher rates are associated with higher vibrations which lead to drilling dysfunctions. Drilling dysfunctions or tool failures would incur replacement costs of tools and a trip (which can be very time intensive) – offsetting any time saved due to higher ROP.

5.1.2 TOB

The TOB objective function would seek to minimize TOB which draws less power from the motor and has shown to reduce drilling vibrations or stick-slip (D. Ertas et al., 2014). Traditionally torque and drag are modeled during the well design process with the objective of minimizing drag for drilling wells (especially extended reach wells). There is a loss of energy along the wellbore, hence the torque measured at the bit is always lower than the torque measured on the surface. Minimum energy used to rotate the drill string providing the same ROP is more advantageous.

5.1.3 MSE

The MSE (Equation 5.1) objective functions seek to minimize MSE which will minimize the amount of energy used to extract rock. MSE is a calculated parameter and measured as a function of ROP and TOB. MSE is not directly modeled but calculated using ROP and TOB models (or measurements).

$$MSE = \frac{WOB}{A_b} + \frac{120\pi * RPM * TOB}{A_b * ROP}, \text{ (Equation 5.1)}$$

5.1.4 Cost

Cost (or cost-per-foot) of drilling has often been used as a metric for bit selection (Ebrahimi & Novieri, 2010; Gjelstad, Hareland, Nikolaisen, & Bratli, 1998; Nygaard et al., 2002; Rashidi, Hareland, & Nygaard, 2008) but can be modified as a drilling optimization metric. Intuitively, the cost-per-foot of drilling should be the key metric to optimize, since the main objective of drilling optimization is to improve its economics.

Costs are broken down into two categories: fixed and variable operation costs. Fixed costs include costs such as bit cost, rig cost, equipment cost. Variable costs depend on the time taken to drill, salaries and so on. A back-of-the-envelope calculation can be performed by simplifying the cost-per-foot using Equation 5.2.

$$CPF = \frac{Bit\ Cost + (Drilling\ Time + Trip\ Time) * Rig\ Cost}{Feet\ drilled\ in\ bit\ run}, \text{(Equation 5.2)}$$

Unless the well is tripped due to routine operations such as casing the well or running tools, the effect of the fixed costs is minimized as the hole gets deeper. When a new bit is run in hole, the initial cost-per-foot for that bit will be infinite given the nature of the equation. However, this cost decreases as the well is drilled and variable costs increase. Eventually the time-related costs overtake the fixed costs (as drilling proceeds). A cost-per-foot plot has been shown in Figure 5.1 on both linear and log scale. Figure 5.2 plots cost-per-foot for different formations.

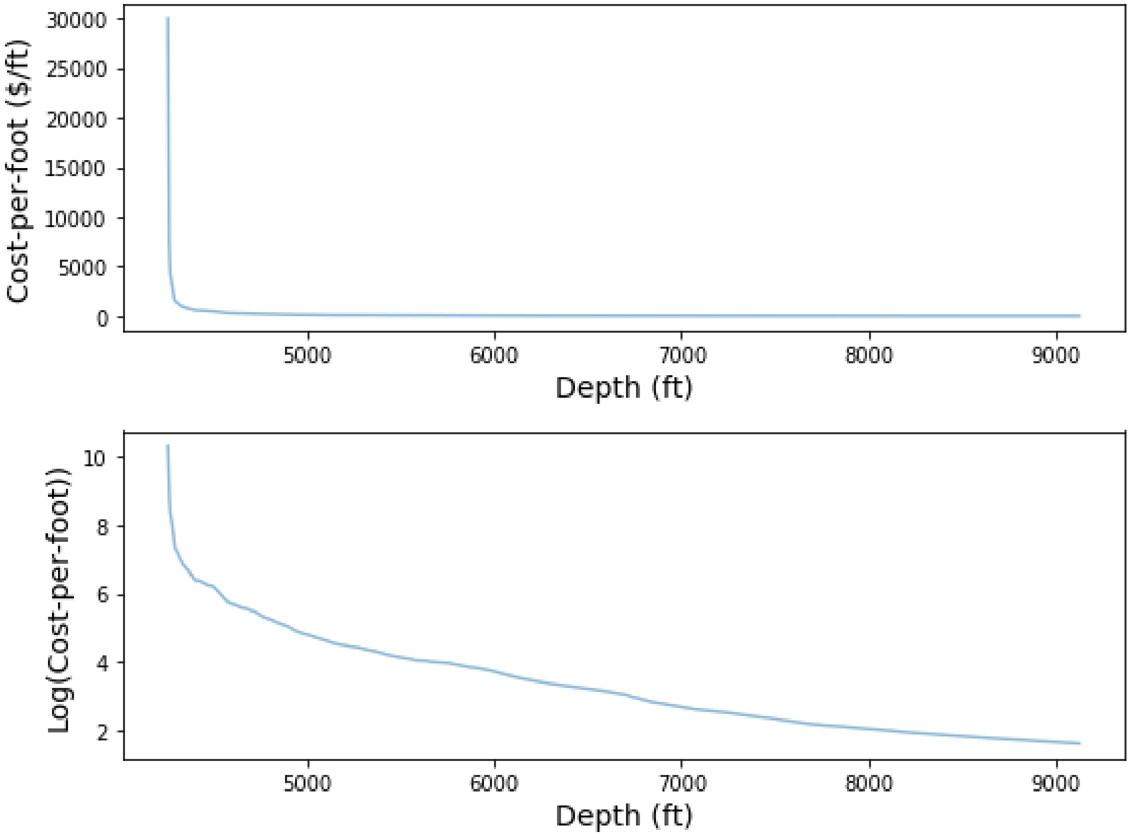

Figure 5.1: Cost-per-foot of drilling the well on linear (top) and log (bottom) scale. A linear decline is seen after the first hundred feet of drilling on the linear scale and thousand on the log scale. A deviation from this linear decline is often used as a signal for bit change.

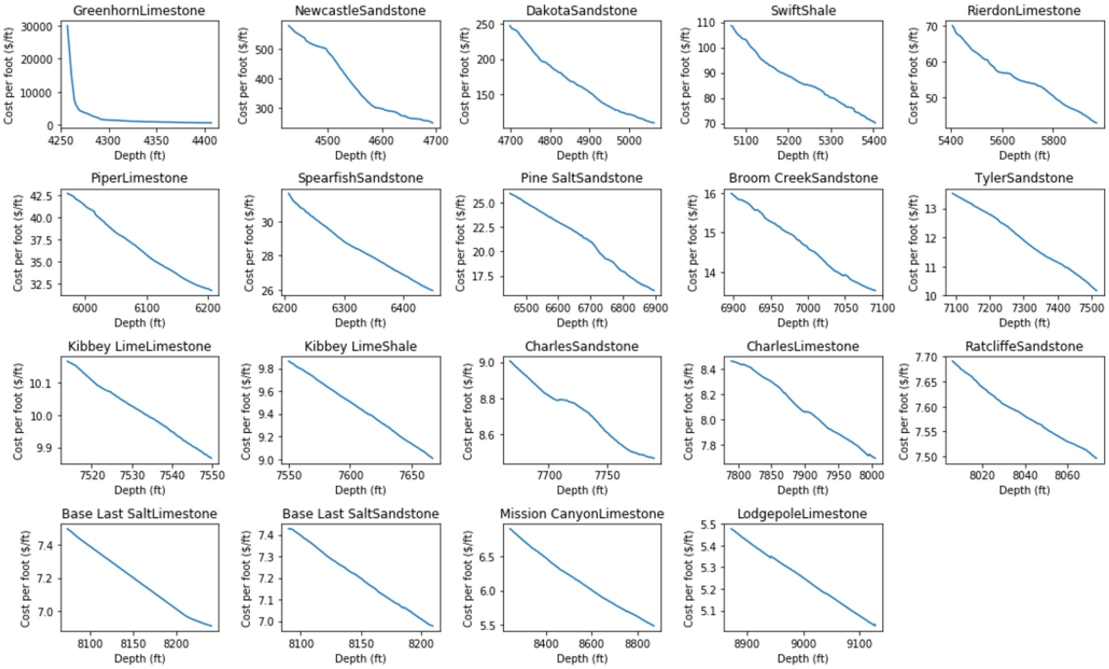

Figure 5.2: Cost-per-foot plots for each formation. A deviation from linear decline has neem commonly used to infer the presence of a dull bit.

Figures 5.1 and 5.2 were plotted using well-related costs outlined in Table 5.1 and a rig rate of $30000/day. Even though intuitively it makes sense to use cost-per-foot as a metric for drilling optimization, it does not work well from an optimization perspective. The main component of Equation 5.2 which can be controlled is the time taken (including on-bottom and off-bottom) due to drilling. Out of these times, the only the ROP can be modeled using controllable drilling parameters and subsequently optimized. The decrease of downtime or tripping time cannot be controlled by changing rig-based parameters but should be approached from a management perspective. For example, if 100 ft of the well remains to be drilled, and the bit is dull, rather than change the bit, the remaining depth can be drilled at a lower speed. Drilling at a lower speed will outweigh the time taken to change the bit. Such a decision can be made by the engineer on-site (an algorithm is not required).

Similarly, improving tripping times – which depend on equipment and crew (C. M. Hegde et al., 2018) – can be better dealt with efficient management practices. The cost-per-foot metric heavily relies on drilling time which is controlled by ROP and for the purposes of this thesis does not differ from improving the rate of drilling. Hence, the main metrics which are evaluated in this chapter are ROP, TOB, and MSE.

Table 5.1: Drilling details from daily report used to calculate CPF

Start (ft)	End (ft)	DrillingTime (hrs)	Non-Drilling Time (hrs)	Depth Drilled (ft)
2237	4212	7	17	1975
4212	6763	22	2	2551
6763	7693	22.5	1.5	930
7693	8661	22.5	1.5	968
8661	9130	10.5	13.5	469

Models used for running experiments to evaluate these metrics have been covered in Chapter 2. Optimization algorithms and the optimization space (or dimensions) used to find optimal parameters have been covered in Chapter 4. Figure 5.3 plots the accuracy of a random forest model used for ROP, TOB, and MSE.

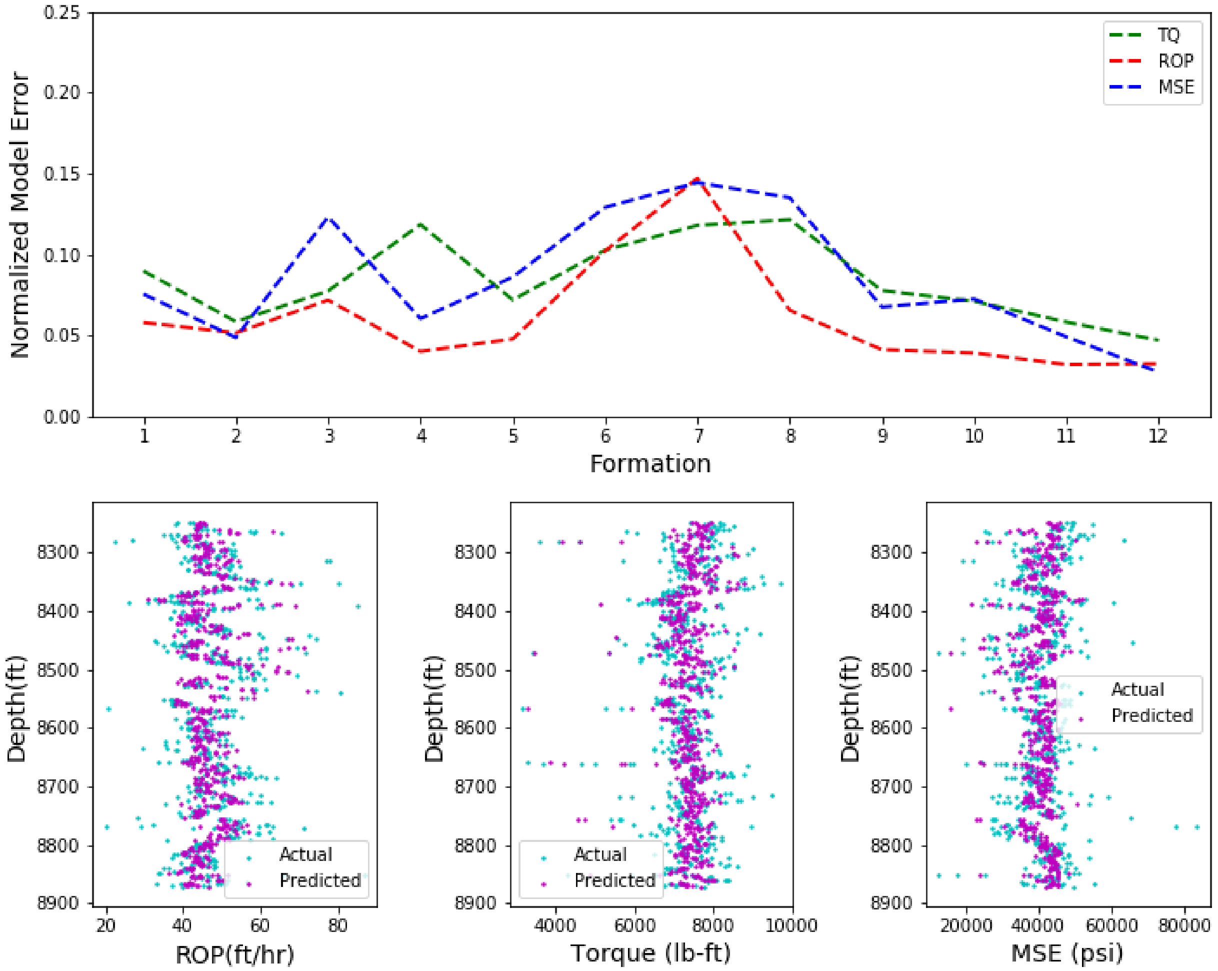

Figure 5.3: Evaluation of errors due to ROP, TOB and MSE model predictions. (top) Normalized errors of MSE, TOB and ROP model. Random forest algorithm was used to train a model on each formation using half the data for training. The trained model was evaluated for prediction accuracy on the test data. The errors are well within 15% for ROP, TOB, and MSE, showing the accuracy of these models as compared to the field evaluated measurements. (bottom) The models are used to predict ROP (left), TOB (middle) and MSE (right), and compared to the measurements seen in the field. As seen from these figures the models perform well with a low error during prediction. These models are sufficiently accurate to be used thereafter in this paper for optimization analysis and simulations.

5.2 Drilling Optimization Model

As described in the previous section, data-driven models are built for ROP, TOB, and MSE separately for each formation. The model can be used to predict ROP, TOB or MSE given the drilling control parameters. Since ROP and TOB models are built using the random forest algorithm they are coupled – a change in WOB will show a change in ROP as well as TOB. Correspondingly, since MSE is a function of ROP and TOB, changes in WOB can be translated into changes in MSE. Figure 5.4 shows the basic methodology behind data-driven optimization in drilling (similar to the process explained in Chapter 2). A portion of the well is drilled (without any modeling), and the data collected during drilling this interval are called the training set. This training data are used to build data-driven models for ROP, TOB, and MSE. An objective function is defined and optimized using an optimization algorithm with constraints set by the on-site engineer. It is important to note that only unmodified metrics (ROP, TOB, or MSE) have been analyzed in this chapter and not combinations since the model is already coupled. The ExxonMobil DAS combines different drilling metrics towards an integrated objective function which is then minimized. This is required since the DAS does not couple drilling metrics; ROP was estimated using response surfaces and TOB using an analytical model. The machine learning models used in this chapter are coupled, defining a single objective is sufficient. The optimized control parameters are used to drill ahead of the bit. After the completion of another joint/stand, newly acquired data are used to update the model. This process is repeated in a closed loop process until the entire formation is drilled.

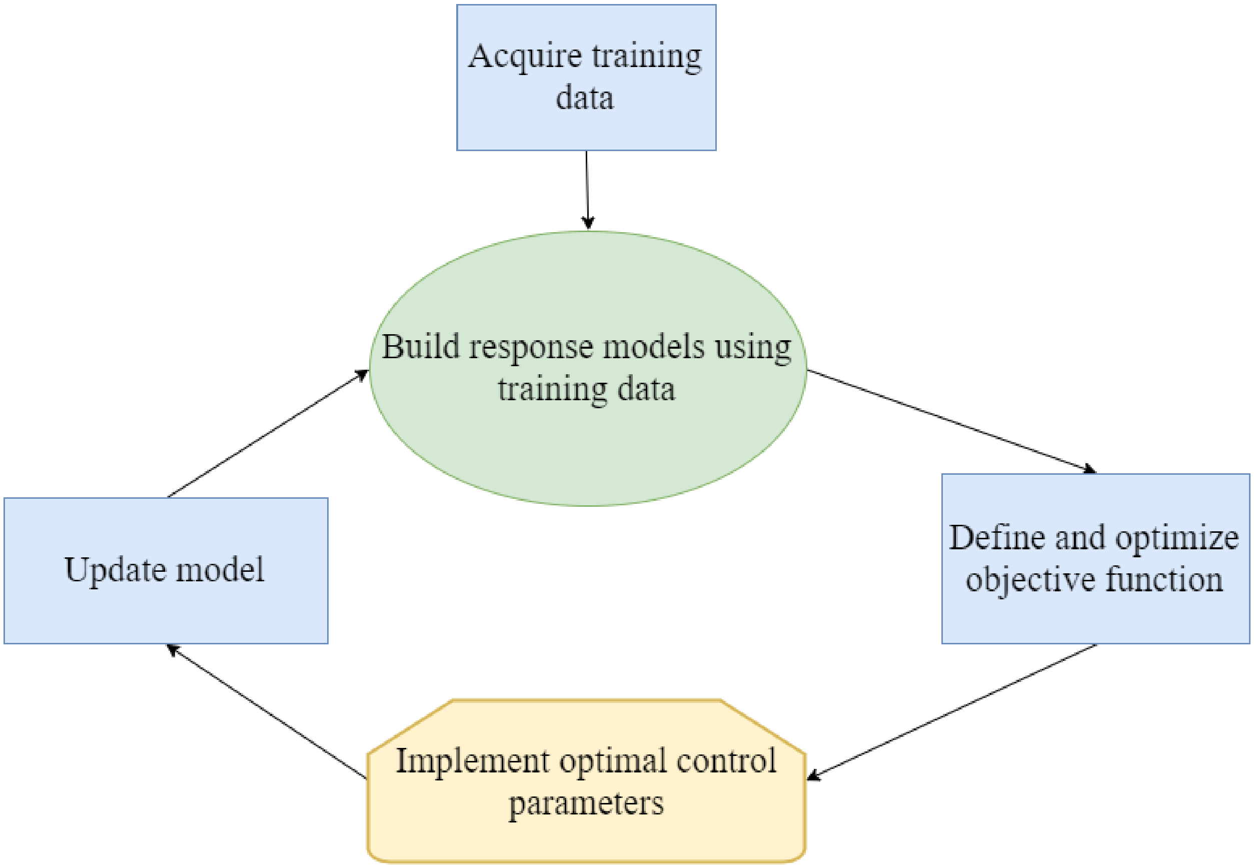
Acquire training data
Build response models using training data
Define and optimize objective function
Implement optimal control parameters
Update model

Figure 5.4: Flowchart describes the drilling optimization process. The first step is to acquire drilling data by drilling one or more stands into a formation. These data are not modeled and treated as training data. Models are then built on this training data. An objective function (MSE for example) is defined and optimized using a metaheuristic algorithm. The optimized drilling control parameters are implemented for drilling the next joint/stand. Data acquired from the next joint/stand can be used to update all the models, after which the cycle is repeated in a closed loop.

5.2.1 Experiments

Experiments (or simulations) are conducted to evaluate and compare different drilling metrics. A simulation would entail using a drilling model to simulate the actual drilling environment. The models can be used with new control parameters to simulate the

ROP, TOB, and MSE ahead of the bit (or in any other unknown environment). An example simulation has been expanded for clarity.

5.2.2 Simulation Example

5.2.2.1 Acquire Training Data

One stand of lodgepole limestone formation is drilled. While drilling this formation acquired surface parameters are used for training data-driven models. Driller is encouraged to vary drilling control parameters (WOB, RPM, flow-rate) while drilling this section so that sufficient ranges of drilling control parameters are covered (Sanderson et al., 2017). Implementation of a design of experiments sampling scheme such as the Taguchi method (Taguchi, 1986) can be an efficient way to carry out sampling training data (more details covered in the appendix).

5.2.2.2 Build Model

ROP is modeled using the random forests algorithm with WOB, RPM, flowrate, and UCS as input parameters. A similar model is constructed for TOB. The MSE is calculated using the ROP and TOB models.

5.2.2.3 Find Recommended Settings

MSE is chosen as an objective function; MSE is calculated for the training data and minimized using an optimization algorithm with control parameter constraints specified by the on-site engineer. Optimal control parameters which minimize the MSE are communicated to the driller to drill the next joint/stand.

5.2.2.4 Implement Control Parameters

The optimal control parameters (WOB, RPM, flow-rate) are used by the driller to drill the next joint/stand. Data are collected while drilling this stand, which is then added to the training set. In this thesis, to simulate such a setting, the optimal parameters are fed into the drilling model (consisting of data-driven models) to simulate the ROP, TOB, and MSE.

5.2.2.5 Update Model

Additional data improves the accuracy of data-driven models. Hence, it is recommended that the models are repeatedly retrained with newly added training data. The frequency of retraining or updating the model is left to the preference of the operator. This process is repeated in a closed-loop until the entire formation is drilled. Additional details are covered in the appendix.

5.3 OPTIMIZATION ALGORITHMS

As introduced in chapter 4, meta-heuristic algorithms work well for finding optimal parameters of data-driven models. Hence the particle swarm optimization (PSO) algorithm has been used to find optimal parameters of drilling metrics. While this algorithm works well for ROP and TOB minimization, it takes very long to converge for optimizing MSE due to the Pareto optimality condition (Censor, 1977). Since the minimization of MSE requires TOB to be minimized and ROP to be maximized, it contains a dual objective which depends on the same input parameters, converging to a solution can be difficult. The random search algorithm converges fast but fails to retrieve a good solution. Bayesian optimization using Gaussian processes yield a good solution which converges quickly. The theory, application, and results of using Gaussian processes for optimization have been briefly explained in this section.

Bayesian optimization can be used to find optimal parameters of a “black box” function by sampling from a Gaussian process (GP). The tractable posterior distribution induced by the GP leads to efficient use of the information gathered by previous experiments, enabling optimal choices about what parameters to try next (Snoek, Larochelle, & Adams, 2012).

5.3.1 Theory

As with the case of metaheuristic algorithms, this algorithm seeks to find the minimum (or maximum) of the objective function ($f(x)$) on some bounded set χ. Bayesian optimization is different from other optimization algorithms discussed in chapter 4 since it first constructs a model for $f(x)$ before exploiting this model to find the next evaluation point. The idea is to use all previous evaluations of the function $f(x)$ and not just the local gradients which provide a great trade-off in the exploration-exploitation space. This helps find the minimum (or maximum) in a few iterations or function calls as opposed to other algorithms such as DE or PSO. The reader is referred to the paper by Brochu (Brochu et al., 2010) for an in-depth analysis of the algorithm.

Figure 5.5 shows a typical run of Bayesian optimization on a toy 1D problem (Brochu et al., 2010). The optimization starts with two points. At each iteration, the acquisition function is maximized to determine the next sampling location. The objective is then sampled at the argmax of the acquisition function; the Gaussian process is updated, and the process is repeated. Bayesian optimization has two primary components: the posterior distribution over the objective (solid line in Figure 5.5) and the acquisition function (colored green in Figure 5.5).

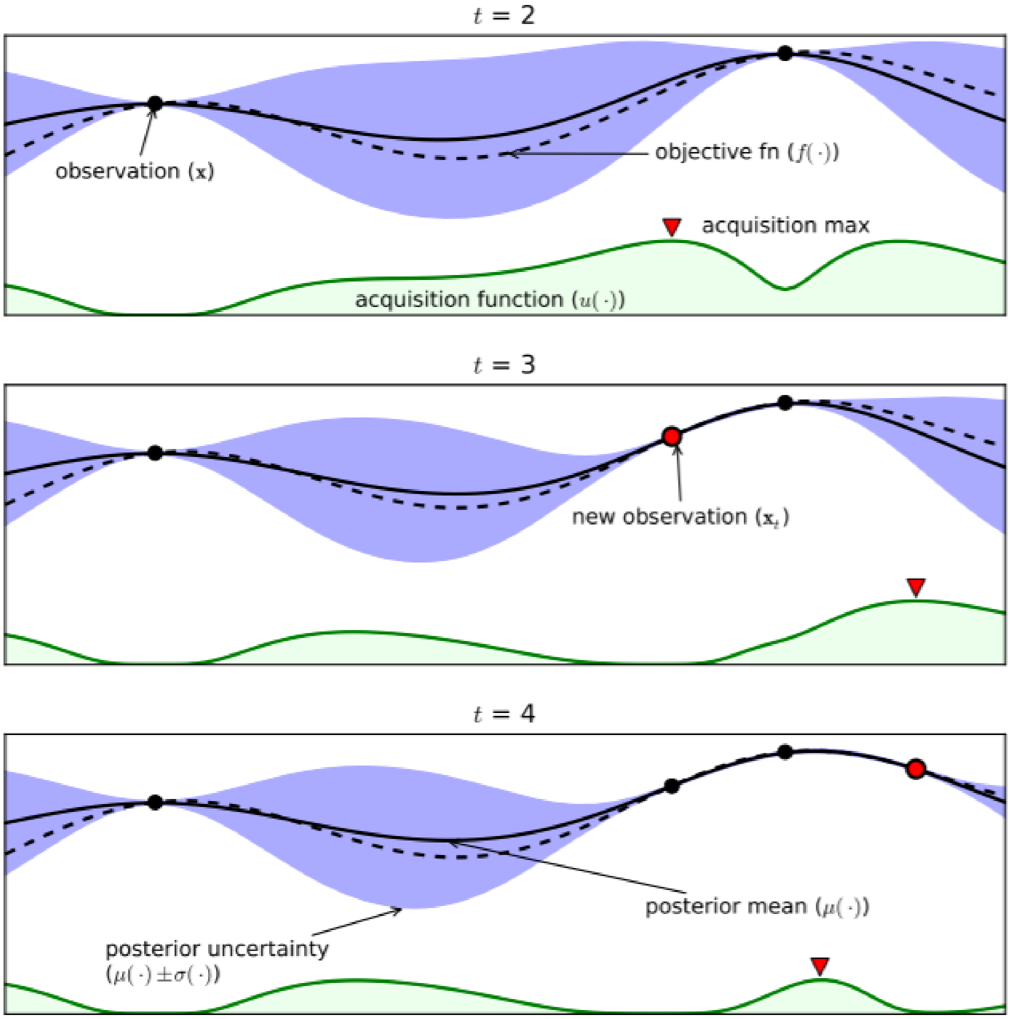

Figure 5.5: An example of using Bayesian optimization on a toy 1D design problem reproduced *(Brochu et al., 2010)*. The figures show a Gaussian process (GP) approximation of the objective function over four iterations of sampled values of the 1D objective function. The figure also shows the acquisition function in the lower shaded plots. The acquisition is high where the GP predicts a high objective (exploitation) and where the prediction uncertainty is high (exploration)—areas with both attributes are sampled first.

5.3.1.1 Posterior Distribution

Baye's rule (Equation 5.3) defines the prior, posterior and likelihood.

$\mathrm{P(M|D)} = \mathrm{P(E|D)} * \mathrm{P(D)}$, (Equation 5.3)

where, $\mathrm{P(D)}$ is the prior, $\mathrm{P(E|D)}$ is the likelihood, and $\mathrm{P(M|D)}$ is the posterior distribution. The prior of the model is some prior belief that is expected of the function being approximated. Many families of functions can be used for a prior, however, it has been shown that using a Gaussian Process (GP) as a prior is successful for Bayesian Optimization (Mockus, 1994). A GP is a stochastic process such that each collection of random variables has a multivariate normal distribution. A Gaussian distribution can be completely specified by its mean and covariance; similarly, a GP can be specified by its mean and covariance matrices. Hence, when evaluated at *x*, rather than returning a scalar value, the GP returns a mean and variance of a normal distribution which define that input *x*. It is commonly assumed for simplicity that the mean function is 0, however, the covariance function is predefined. The covariance function or kernel are application specific; commonly used kernels in machine learning are exponential distance kernel, Matern kernel (Minasny & McBratney, 2005), and the logistic kernel (Ng, 2000). The best results were reported with the use of the exponential distance kernel (Equation 5.4).

$\mathbf{k}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{exp}(-\frac{1}{2}||\boldsymbol{x_i} - \boldsymbol{x_j}||^2)$, (Equation 5.4)

This function approaches a value of 1 when points are close together and a value of 0 as points are far apart. A covariance matrix can be constructed for each pair of points sampled from the data as shown in Equation 5.5.

$$\mathrm{K} = \begin{pmatrix} \mathbf{k}(\boldsymbol{x_1}, \boldsymbol{x_1}) & \cdots & \mathbf{k}(\boldsymbol{x_1}, \boldsymbol{x_n}) \\ \vdots & \ddots & \vdots \\ \mathbf{k}(\boldsymbol{x_n}, \boldsymbol{x_1}) & \cdots & \mathbf{k}(\boldsymbol{x_n}, \boldsymbol{x_n}) \end{pmatrix}, \text{(Equation 5.5)}$$

For the purposes of optimization, previously evaluated points are used to make future predictions using the GP. Assuming that *n* points have already been sampled from the function, the value of the *n+1th* point can be evaluated using Equation 5.6.

$$\begin{pmatrix} f_{1:n} \\ f_{n+1} \end{pmatrix} \sim N\left(0, \begin{pmatrix} K & k \\ k^T & \mathbf{k}(\boldsymbol{x_n}, \boldsymbol{x_n}) \end{pmatrix}\right), \text{(Equation 5.6)}$$

where, f_{n+1} is the function evaluation at the unknown point, $f_{1:n}$ are the known function values, K is the covariance matrix, and k is the covariance of the unknown point with all known points (using Equation 5.4). This can be solved using the multivariate gaussian theorem (P. Murphy, 1991) giving the solutions laid out in Equation 5.7 and 5.8.

$$\mu(x_{n+1}) = k^T K^{-1} f_{1:n} \text{ , (Equation 5.7)}$$

$$\sigma^2(x_{n+1}) = k^T K^{-1} f_{1:n} \text{ , (Equation 5.8)}$$

where, μ is the mean and σ is the standard deviation of the newly predicted point.

5.3.1.2 Acquisition Functions

The role of the acquisition function is to guide the search for the maximum or minimum of the objective function. The sequential design for optimization (SDO) algorithm (Cox & John, 1992) is used as the acquisition function. A lower confidence bound (LCB) for minima and upper confidence bound (UCB) for maxima (Equations 5.9 and 5.10) are defined.

$\mathrm{LCB}(\mathrm{x}_{n+1}) \ = \mu(x_{n+1})$ - $\kappa * \sigma\ (x_{n+1})$, (Equation 5.9)

$\mathrm{UCB}(\mathrm{x}_{n+1}) \ = \mu(x_{n+1}) + \kappa * \sigma\ (x_{n+1})$, (Equation 5.10)

where, κ is a hyperparameter chosen using cross-validation.

5.3.2 Bayesian Optimization for ROP and MSE

ROP optimization is efficient using the PSO algorithm, however, can be time intensive. The use of Bayesian optimization yields a good solution with fewer iterations. A GP prior was used with an exponential distance covariance kernel. Ten function evaluations were initiated and with the use of the UCB acquisition function for optimization. Figure 5.6 plots the contour of ROP as a function of drilling control parameters – WOB and RPM – for all formations. The true maximum of ROP has been plotted as a star; the algorithm iterations are numbered. It is evident that a good solution close to the true optimum is returned by the algorithm in all cases, with a few iterations. The average number of iterations or function evaluations for the PSO algorithm was 1075. The contour of Lodgepole Limestone formation has been shown in Figure 5.7 for better clarity. These figures show that Bayesian optimization using an exponential distance kernel and UBC acquisition function can successfully be used for ROP optimization and the optimal parameters returned are close to the true global optimum.

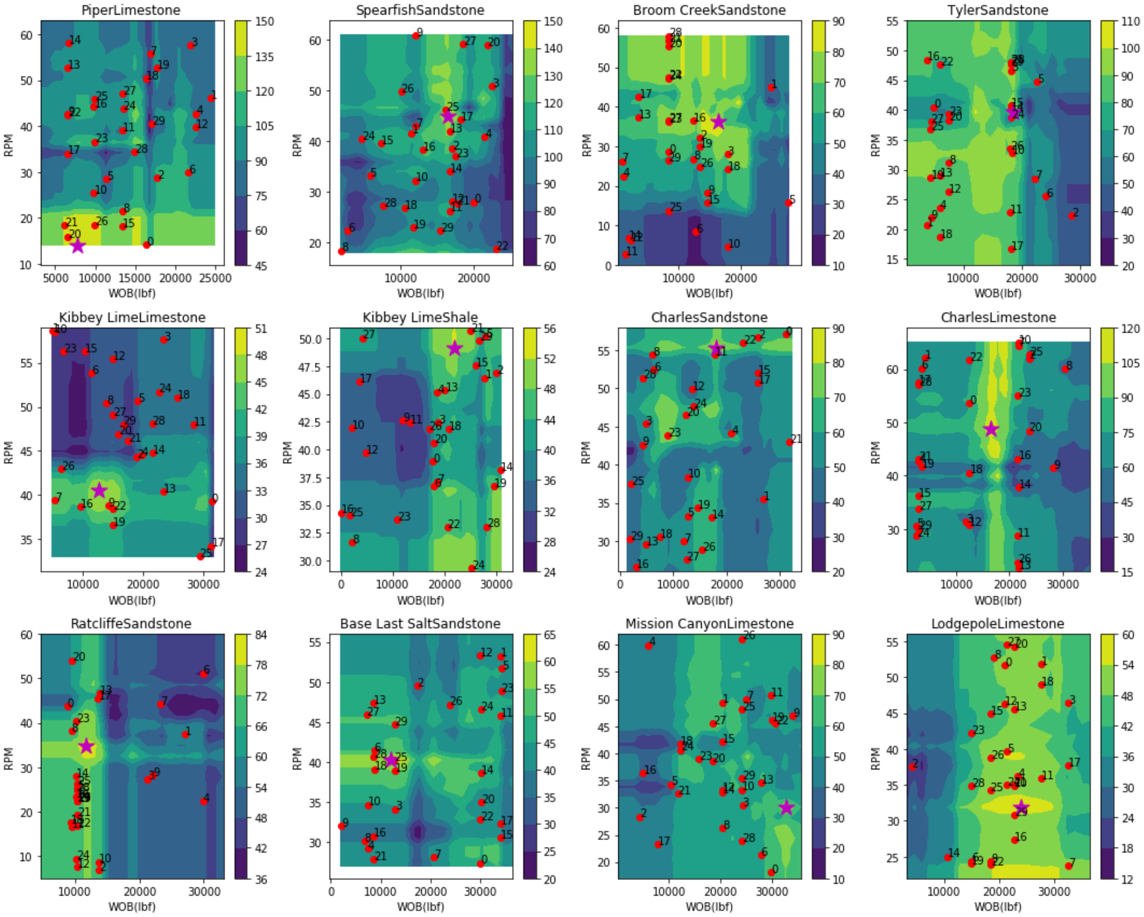

Figure 5.4: Bayesian optimization algorithm used to find optimal parameters for ROP for all formations; The iterations of the algorithm have been labelled on each plot; The minimum MSE has been plotted using a maroon star; Each plot is the contour of MSE predictions as retrieved by the model; The star represents the actual minimum as observed in the field.

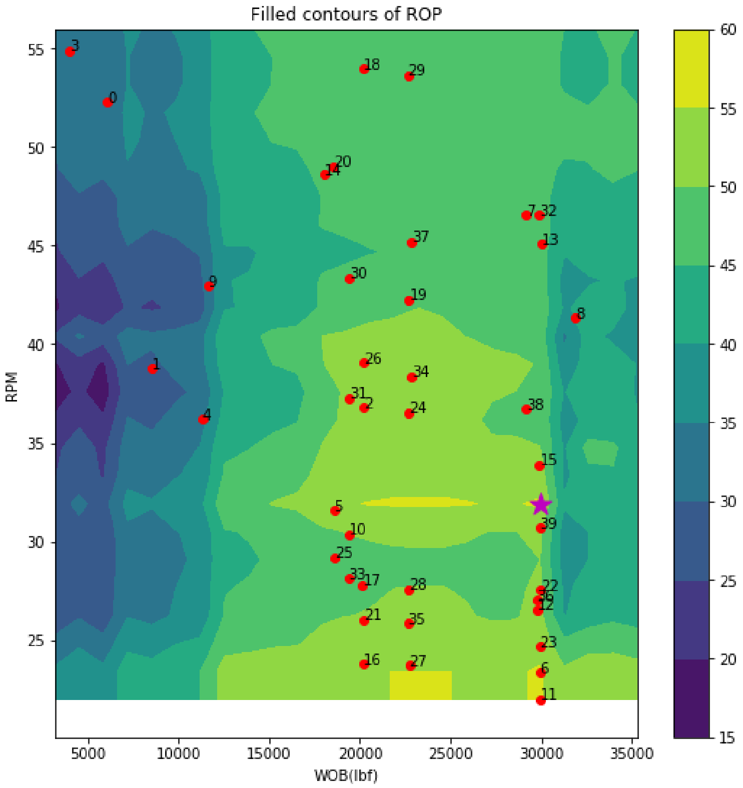

Figure 5.7: ROP contour plot using the model for Lodgepole Limestone; The true maximum of ROP has been plotted as a maroon star; Bayesian optimization is used to find the optimal WOB and RPM; Each iteration or function evaluation of the algorithm has been numbered.

The optimization of MSE is more complicated as compared to ROP since MSE is a function of ROP and TOB. The minimization of MSE requires maximization of ROP as well as the minimization of TOB, both of which depend on the same drilling control parameters. Figure 5.8 plots contours of all three metrics along with their individual argmin/argmax. As seen from this figure, it is evident that simply picking the optima individually for ROP or TOB will not produce the optimal MSE. Figure 5.9 plots the contour of MSE as a function of drilling control parameters – WOB and RPM – for all formations. The true minimum of MSE has been plotted as a star; the algorithm iterations are numbered. It is evident that a good solution close to the true optimum is returned by the algorithm in all cases, with a few iterations. The simplex and differential evolution algorithms did not converge for this problem. The average number of iterations or function evaluations for the PSO algorithm was over two thousand. The contour of Lodgepole Limestone formation has been shown in Figure 5.10 for better clarity. These figures show that Bayesian optimization using an exponential distance kernel and LCB acquisition function can successfully be used for MSE optimization.

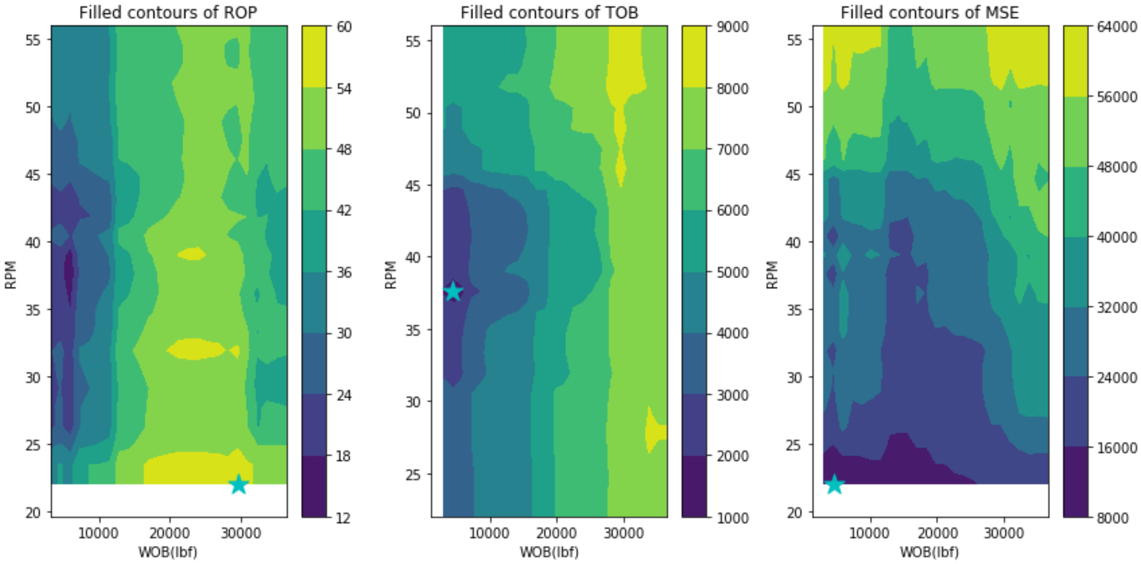

Figure 5.8: Contours for ROP, TOB, and MSE in Lodgepole Limestone as modeled; The true optimal values of each metric have been plotted as a cyan star; Since MSE is a function of ROP and TOB, the optimal control parameters depend on ROP and TOB which are in turn controlled by WOB and RPM.

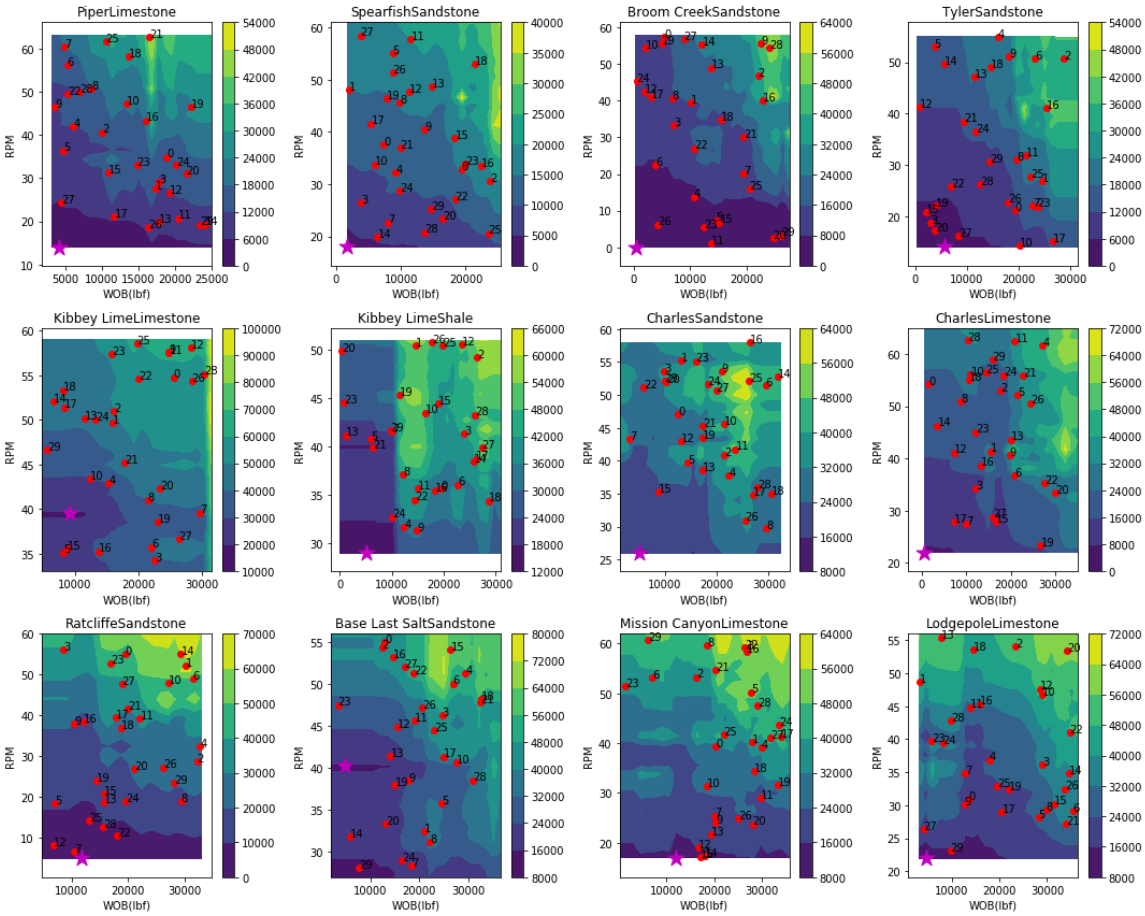

Figure 5.9: Bayesian optimization algorithm used to find optimal parameters for MSE for all formations; The iterations of the algorithm have been labelled on each plot; The minimum MSE has been plotted using a maroon star; Each plot is the contour of MSE predictions as retrieved by the model; The star represents the actual minimum as observed in the field.

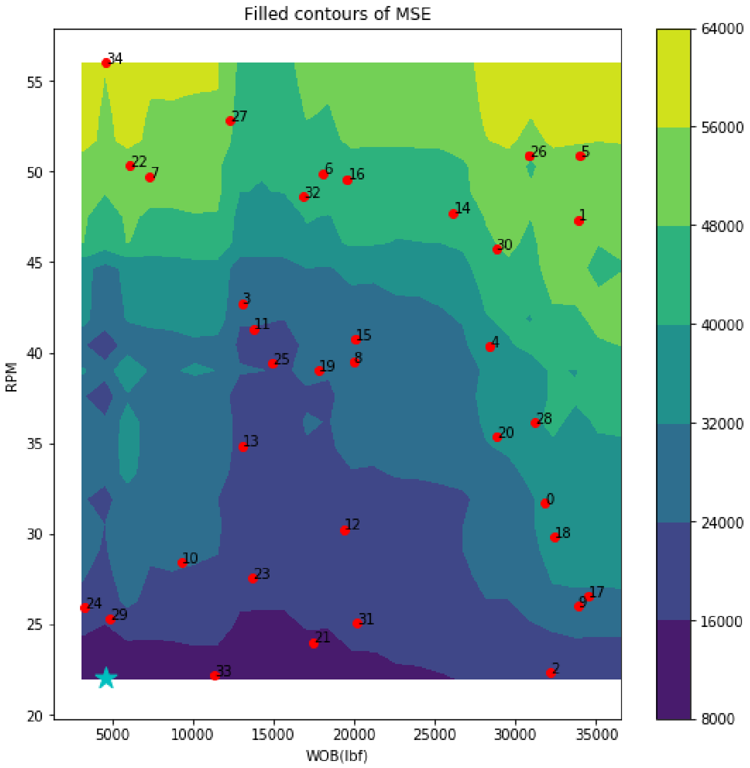

Figure 5.10: MSE contour plot using the model for Lodgepole Limestone; The true maximum of MSE has been plotted as a cyan star; Bayesian optimization is used to find the optimal WOB and RPM; Each iteration or function evaluation of the algorithm has been numbered.

5.4 Results and Discussions

5.4.1 ROP Optimization

ROP is used as the objective function, hence drilling control parameters will be changed to maximize ROP in each formation. In this case, ROP is modeled as a function of WOB, RPM, flow-rate, and UCS on the training data (50% of formation data in this case). Ideal parameters are computed using the PSO algorithm on the trained model for each formation. The change in ROP, MSE, and TOB because of changing these control parameters to maximize ROP has been observed and plotted in Figure 5.11.

The 95% confidence interval of the response has also been plotted. By definition, 95% of the time the range defined by this interval will contain the possible improvement in ROP. In each simulation (to generate the confidence intervals), the training and test set are changed by randomly choosing data points to act as the test and training set respectively. The test-train split – ratio training set length to test set length – is kept fixed at 0.5. At each iteration, 50% of the data points selected in a formation are chosen to be part of the training set, and the rest will be a part of the test set. 1000 bootstrapped simulations are evaluated to calculate the 95% confidence interval for each drilling response parameter. ROP change for each formation is plotted in Figure 5.12. Figures show that using ROP as the objective function has the potential to improve ROP by 28% on average for all formations. However, along with the increase in ROP, there is a corresponding increase in MSE and TOB which may not be desirable.

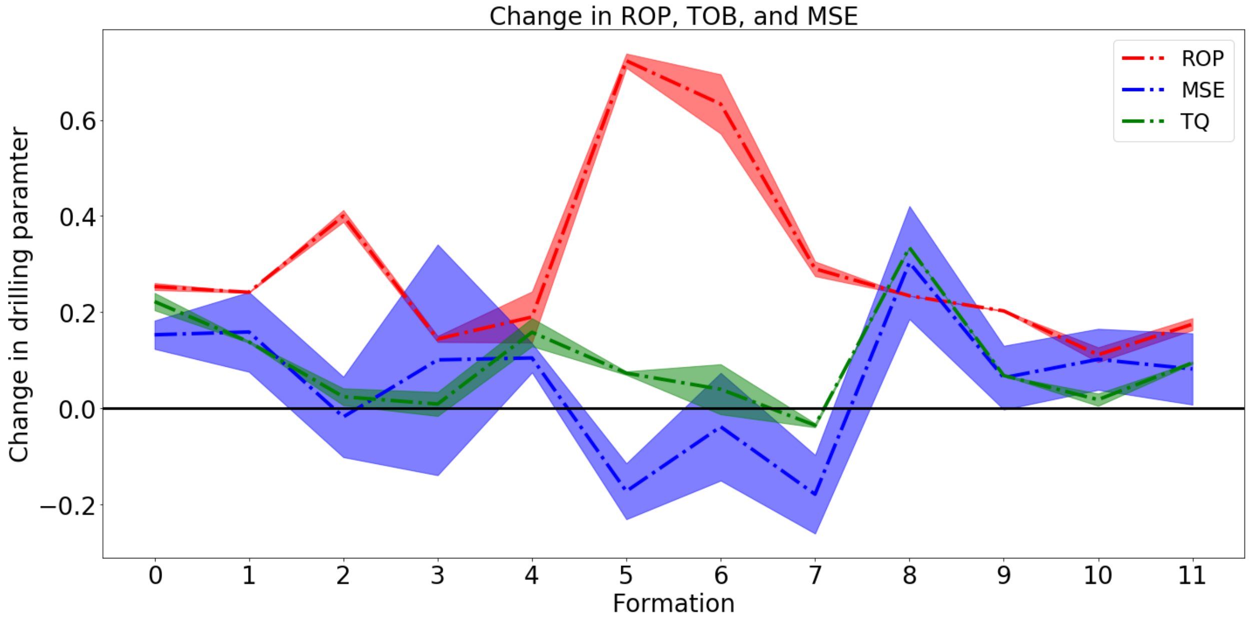

Figure 5.11: Effect on ROP, MSE and TOB on the test set when the formation is drilled with the optimal control parameters calculated by the PSO algorithm using ROP as an objective function. The changes in the drilling parameters are simulated by observing the changes on a machine learning model built for each drilling parameter. The figure shows that if ROP purely is optimized, an increase in ROP is accompanied with an increase in TOB and MSE which might be undesirable. The shaded regions around the dotted lines (for ROP, TOB, and MSE) represent the 95% confidence interval for each prediction.

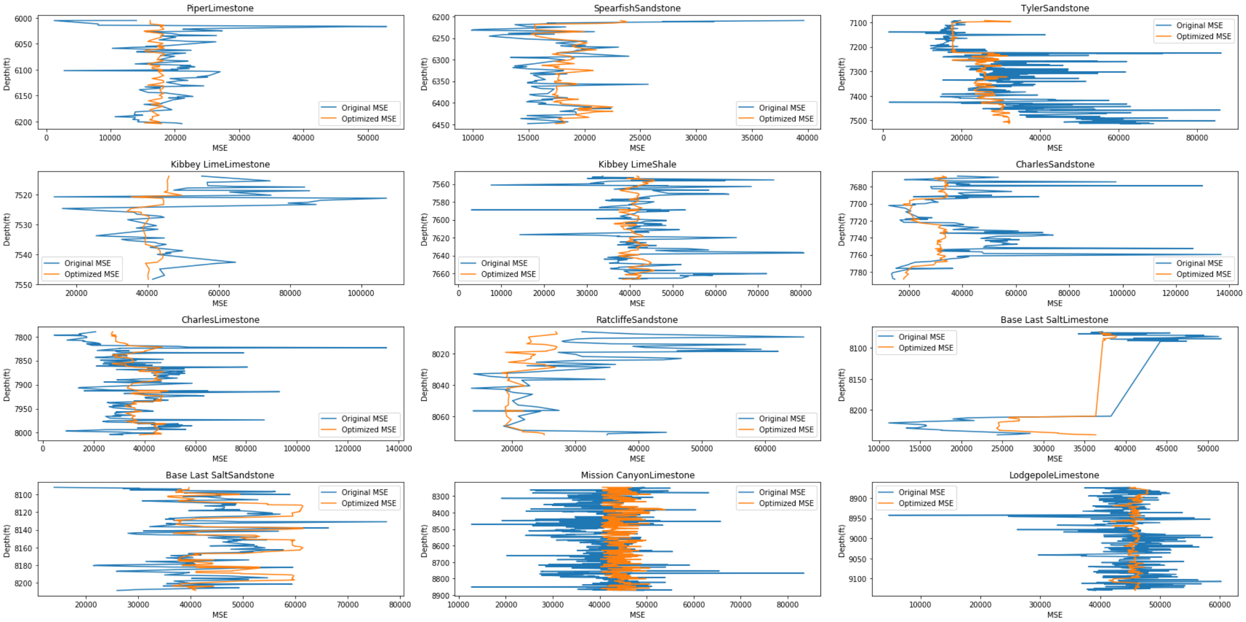

Figure 5.12: Effect of ROP optimization on MSE in each formation. The changes in MSE are simulated by observing the changes on a machine learning model built for ROP and TOB. The figure shows that if ROP purely is optimized, an increase in ROP is accompanied by an increase in MSE which might be undesirable.

5.4.2 TOB Optimization

The TOB response is modeled as a function of WOB, RPM, flow-rate, and UCS on the training data (50% of formation data in this case). These drilling control parameters can be changed to manage or reduce the TOB response and thereby reducing vibrations. Reducing the TOB response of the drill bit will reduce the MSE since MSE is directly proportional to TOB. Ideal parameters are computed using the PSO algorithm on the trained model for each formation. Effect of optimal control parameters and their 95 % confidence interval has been plotted in Figure 5.13. Changes in TOB in each formation have been plotted in Figure 5.14. Figures show that using TOB as the objective function has the potential to reduce TOB by 12% on average for all formations. However, this results in a smaller increase in ROP and an increase in MSE which is undesirable.

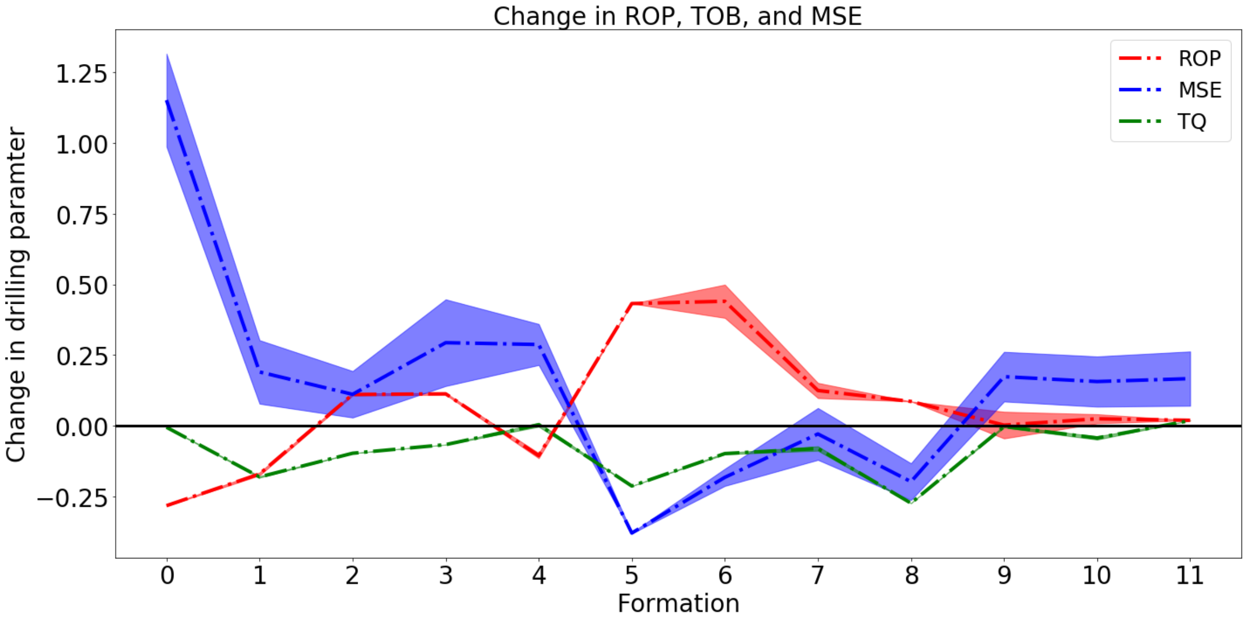

Figure 5.13: Effect on ROP, MSE, and TOB on the test set when the formation is drilled with the optimal control parameters as calculated by the PSO algorithm using TOB as an objective function. The changes in the drilling parameters are simulated by observing the changes on a machine learning model built for each drilling parameter. The figure shows that if TOB purely is optimized, a small decrease in TOB and MSE is accompanied by a reduced increase in ROP. The shaded regions around the dotted lines (for ROP, TOB, and MSE) represent the 95% confidence interval for each prediction.

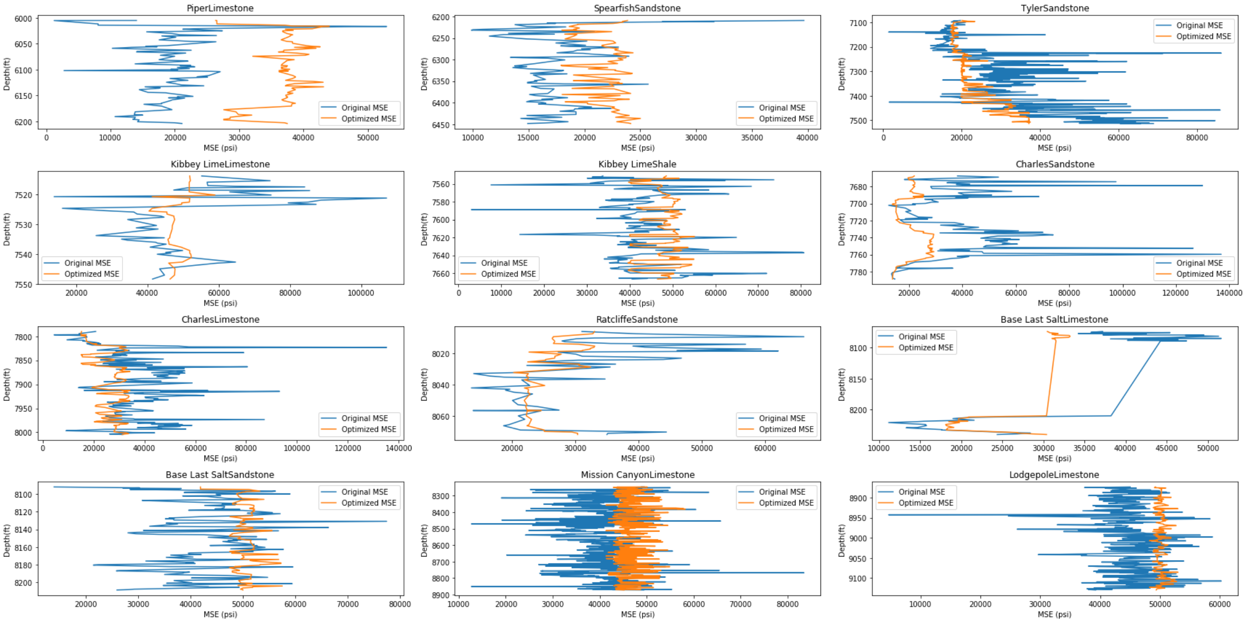

Figure 5.14: Effect of TOB optimization on MSE in each formation. The changes in MSE are simulated by observing the changes on a machine learning model built for ROP and TOB. The figure shows that if TOB purely is optimized, a smaller increase in ROP is accompanied with a slight increase in MSE as compared to purely optimizing ROP

5.4.3 MSE Optimization

MSE (Equation 5.1) is calculated using the ROP and TOB data-driven models with a train-test ratio of 50%. Ideal parameters are computed using the PSO algorithm. However, the convergence, in this case, takes substantially longer (~10x) since there are competing responses. Bayesian optimization described in section 5.3.2 is used to optimize MSE since it leads to faster convergence. ROP and TOB themselves are functions of WOB, RPM, flow-rate, and UCS. Parameters must be chosen such that ROP can be increased, with a slight increase or decrease in torque so that the overall MSE can be minimized. The changes in ROP, MSE, and TOB because of changing these control parameters has been observed and plotted in Figure 5.15 along with their 95% confidence interval. Changes in MSE in each formation have been plotted in Figure 5.15. A slightly smaller increase in

ROP is observed when compared to Figure 5.11, however, reduced TOB and MSE are observed which are desirable. A decrease in MSE shows that less energy is being utilized by the bit to destroy a given volume of rock which shows that it is more efficient. A reduction in TOB may also lead to a reduced vibrational response which can help prevent drilling dysfunction.

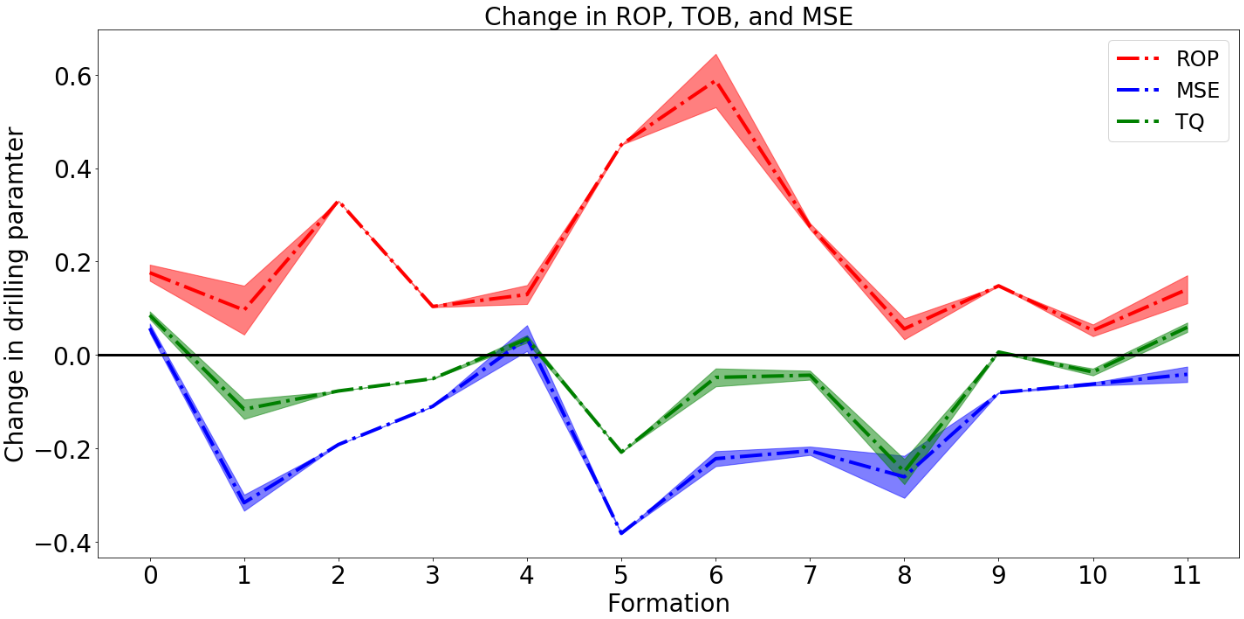

Figure 5.15: Effect on ROP, MSE, and TOB on the test set when the formation is drilled with the optimal control parameters as calculated by the PSO algorithm using MSE as an objective function. The changes in the drilling parameters are simulated by observing the changes on a machine learning model built for each drilling parameter. The figure shows that if MSE is minimized by controlling RPM, WOB and flow-rate to manipulate the MSE, an increase in ROP is accompanied by a decrease in TOB and decrease MSE which is highly desirable. The shaded regions around the dotted lines (for ROP, TOB, and MSE) represent the 95% confidence interval for each prediction.

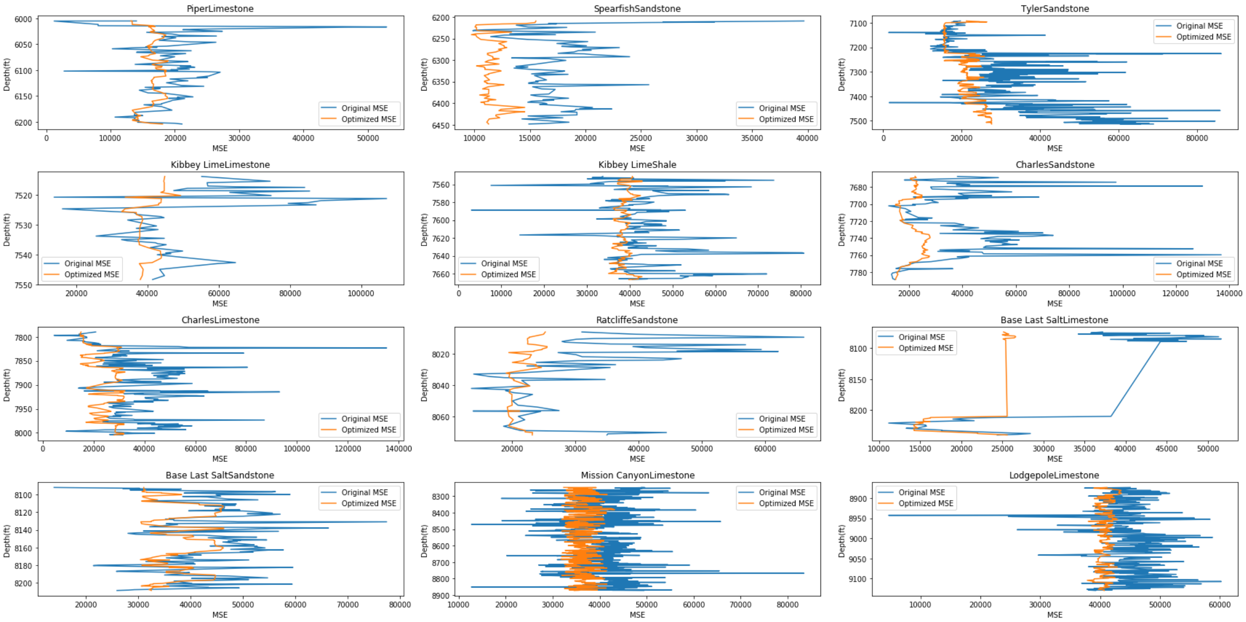

Figure 5.16: Effect of MSE optimization on the MSE response in each formation. The changes in MSE are simulated by observing the changes on a machine learning model built for ROP and TOB. The reduction in MSE in the simulated test set shows that optimal parameters can be calculated which will increase ROP and decrease MSE at the same time

5.4.4 Hypothesis Testing

Hypothesis testing is commonly used in the scientific analysis to test the validity of the hypothesis from a statistical perspective. Hypothesis testing consists of evaluating a null hypothesis and rejecting the null hypothesis if sufficient evidence of the alternative hypothesis exists. It is widely used in the fields of statistics, science, and engineering.

The confidence interval commonly chosen with a width of 95% represents the range which will contain the true value of the mean from the population. For example, if the confidence of interval for MSE in the lodgepole limestone formation is: 5000-15000 psi, this means that the population means of MSE corresponding to lodgepole limestone will lie between 5000 and 15000 psi 95% of the time. Confidence interval in a way represents the lower and upper limits of allowable statistical variation. Hence if the confidence interval of two different distributions intersect they cannot be claimed to be statistically different.

For the study in this dissertation, the objective of the hypothesis tests is to test whether optimization of drilling models results in a significant change in ROP, MSE, and TOB; a difference large enough that it could not have occurred by chance and is actually significant. An example is illustrated in the Figure 5.17 below. In Figure 5.18, ROP optimization has been studied in the Piper limestone formation. The distribution in red plots the ROP distribution before optimization: that observed while drilling the formation. The green distribution represents the ROP post optimization: new ROP once optimal parameters are used for drilling the same formation. The null hypothesis being tested is that both distributions are the same i.e. there is no difference between the distributions observed in red and green. This would imply that the formation is being drilled as efficiently as possible. The alternative hypothesis states that ROP increases post optimization. In other words, there exists a solution or set of input parameters which can result in an improved ROP while drilling this formation. Based on the p-value for a two-sample t-test (Casella & Berger, 2002) conducted on the data – which was 5.43 E-6 – the null hypothesis is rejected in favor of the alternative hypothesis for the analysis pertaining to Figure 5.17. The optimization algorithm’s solution helps improve the ROP. Such an analysis can also be useful in determining the efficiency of the optimization algorithm. This principle can be easily extended to MSE and TOB analysis, to check if the improvements shown by the simulator post optimization is significant compared to the data observed before.

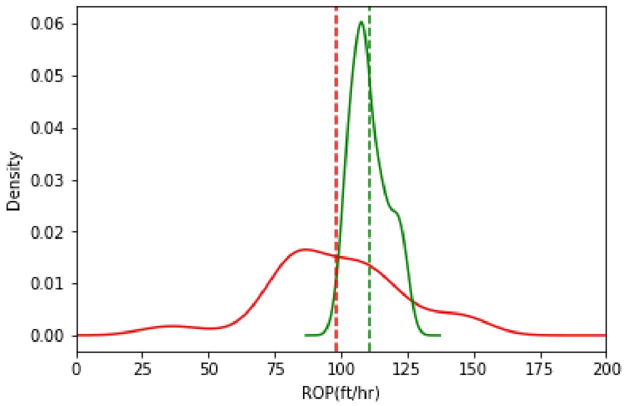

Figure 5.17: Figure showing hypothesis test on ROP simulation. Distribution in red represents the distribution of ROP values before ROP optimization. The distribution in green shows the ROP values after optimization. Hypothesis testing is used to determine the p-value based on the difference between the two distributions based on their mean (represented as dotted lines in the figure) and standard deviation. This case when analyzed using two-sample t-test results in a p-value of 5E-6 showing that they are different distributions.

This analysis assumes that the data points are independent, the distributions are normal or Gaussian, and ROP predictions made using the ROP model are accurate. The ROP predictions are accurate since they have been modeled using a random forest algorithm; they have been tested to show a low error, which indicates that the random forest algorithm predicts ROP accurately and should generalize well for a given formation. The data points are independent given their sampling procedure and filters employed. However, not all distributions of ROP are approximately normal. Distributions can deviate from normality as shown in Figure 5.18 in the Charles Sandstone formation. Violation of this normality assumption can lead to incorrect conclusions due to overinflated p-values and falsely narrow confidence intervals. An alternative is to utilize a non-parametric method such as the bootstrap to calculate confidence intervals for each distribution. If confidence intervals of the difference between the means of the two distributions do not include 0, they are said to be statistically different: the null hypothesis is rejected. In the case of Figure 5.18, the confidence interval of the difference of the means is: 21.63 - 29.29 ft/hr.

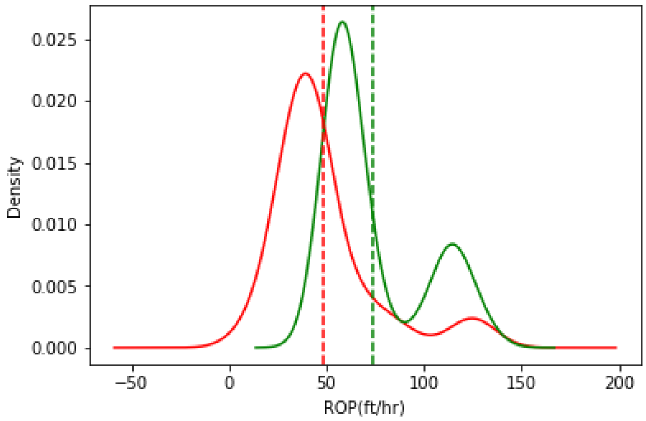

Figure 5.18: Figure showing hypothesis test on ROP simulation. Distribution in red represents the distribution of ROP values before ROP optimization and green after. The means are plotted in the form of dotted lines. In this case, the normal assumption breaks down for both distributions which affect the p-values and conclusions of hypothesis testing.

A hypothesis test was conducted to determine the significance of ROP improvement in each formation. The distributions of the original and improved ROP along with their means have been plotted in Figure 5.19. Both analytic confidence intervals and bootstrap confidence intervals have been shown in Table 5.2. Based on the results seen in Table 5.2, it can be concluded that ROP optimization significantly improves the ROP in all formations and is not likely to have been caused by chance. Figure 5.20 and Table 5.3 summarize similar statistics for TOB which show that all TOB improvements are significant. Figure 5.21 and Table 5.4 summarize similar statistics for MSE. MSE reduction using optimization for all but one formations are significant. The confidence interval for Kibbey Limestone contains 0, which means that a statistically significant difference in the means is not observed. All analysis have been tabulated and displayed in Tables 5.2,5.3, and 5.4. The difference in their values depicts the results due to the non-normality of the probability distributions.

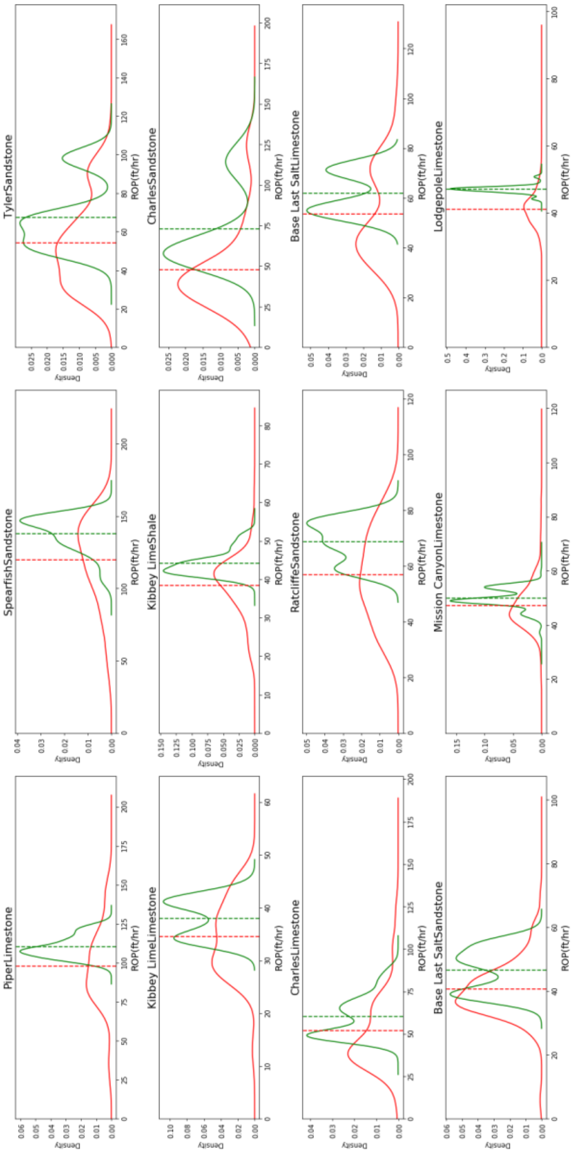

Figure 5.5: ROP optimization distribution for all formations. ROP plotted in red refers to measured ROP. The distribution plotted in green is the distribution of the optimized ROP. The means are plotted in dotted lines.

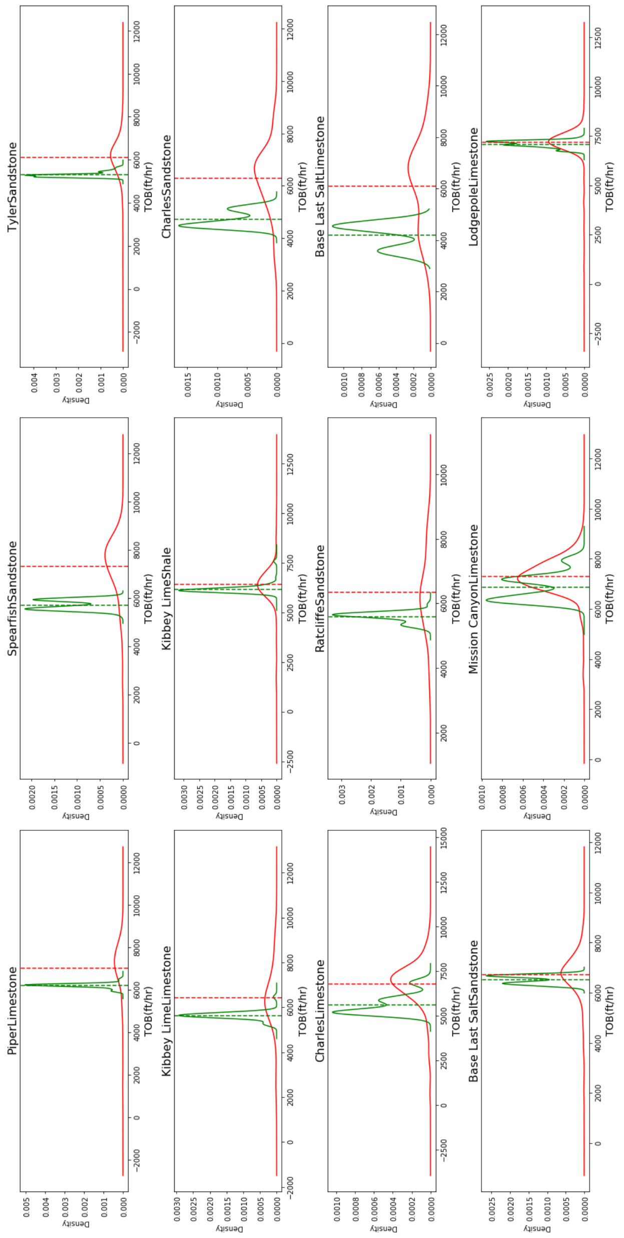

Figure 5.20: TOB optimization distribution for all formations. TOB plotted in red refers to measured TOB. The distribution plotted in green is the distribution of the optimized TOB. The means are plotted in dotted lines.

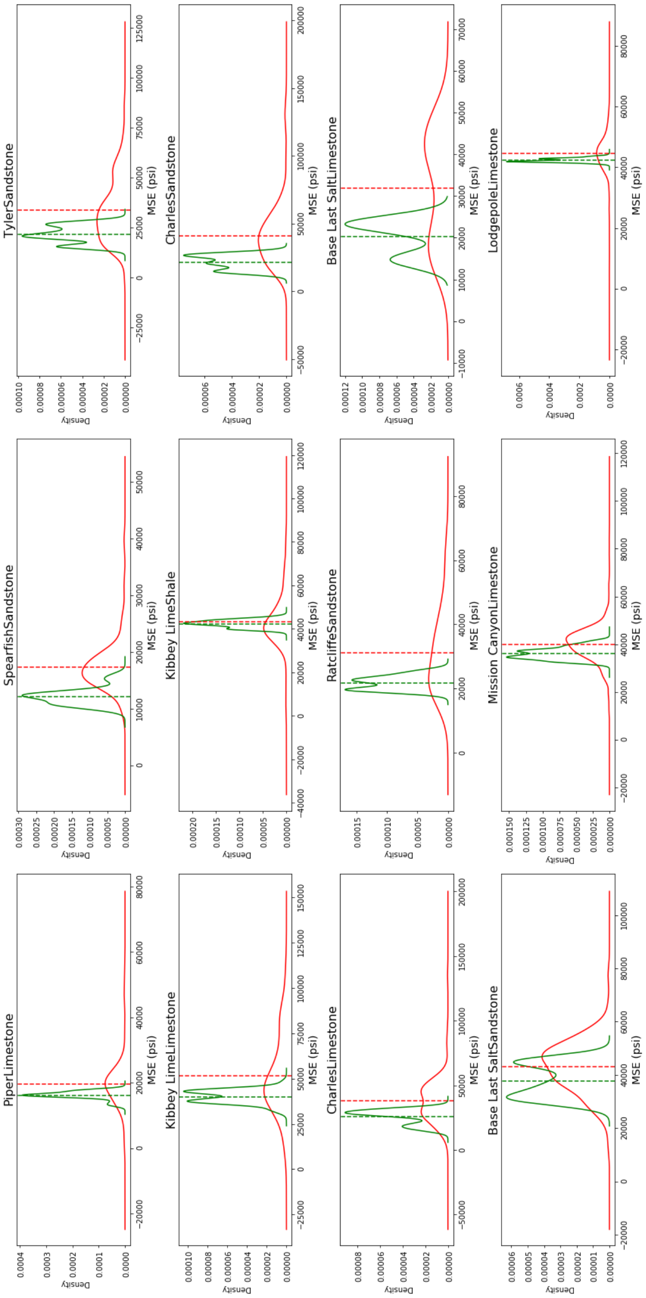

Figure 5.21: MSE optimization distribution for all formations. MSE plotted in red refers to measured MSE. The distribution plotted in green is the distribution of the optimized MSE. The means are plotted in dotted lines.

Table 5.2: Confidence Intervals for difference in means for ROP Optimization

Formation	Analytic 2.5 Percentile (ft/hr)	Analytic 97.5 Percentile (ft/hr)	Boostrap 2.5 Percentile (ft/hr)	Boostrap 97.5 Percentile (ft/hr)
Piper Limestone	7.88	17.00	8.00	17.03
Spearfish Sandstone	12.66	23.84	13.18	24.40
Tyler Sandstone	11.74	14.91	11.70	14.92
Kibbey Lime Limestone	2.06	5.04	2.40	5.60
Kibbey Lime Shale	4.29	7.05	4.37	7.09
Charles Sandstone	21.36	28.99	21.54	29.16
Charles Limestone	5.05	10.81	5.93	10.69
Ratcliffe Sandstone	8.69	14.91	8.68	14.88
Base Last Salt Limestone	4.58	12.12	3.95	11.60
Base Last Salt Sandstone	4.66	7.13	4.70	7.14
Mission Canyon Limestone	2.18	3.38	2.16	3.36
Lodgepole Limestone	5.29	5.49	5.32	5.53

Table 5.3: Confidence Intervals for difference in means for TOB Optimization

Formation5.3	Analytic 2.5 Percentile (lb-ft)	Analytic 97.5 Percentile (lb-ft)	Boostrap 2.5 Percentile (lb-ft)	Boostrap 97.5 Percentile (lb-ft)
Piper Limestone	-1092.71	-561.972	-1048.28	-497.676
Spearfish Sandstone	-1835.13	-1369.51	-1814.27	-1352.63
Tyler Sandstone	-903.065	-713.406	-898.202	-707.66
Kibbey Lime Limestone	-1204.53	-423.405	-1187.92	-425.699
Kibbey Lime Shale	-384.939	-95.6488	-370.342	-77.3619
Charles Sandstone	-1815.55	-1332.19	-1808.06	-1321.17
Charles Limestone	-1349.05	-965.215	-1334.8	-951.774
Ratcliffe Sandstone	-1077.36	-451.148	-1067.67	-452.202
Base Last Salt Limestone	-2265.38	-1525.56	-2263.11	-1531.31
Base Last Salt Sandstone	-344.398	-49.3379	-324.733	-29.2358
Mission Canyon Limestone	-495.743	-365.405	-493.632	-362.429
Lodgepole Limestone	-189.388	-32.6174	-174.009	-11.4547

Table 5.4: Confidence Intervals for difference in means for MSE Optimization

Formation	**Analytic 2.5 Percentile (psi)**	**Analytic 97.5 Percentile (psi)**	**Boostrap 2.5 Percentile (psi)**	**Boostrap 97.5 Percentile (psi)**
Piper Limestone	-4931.69	-2095.17	-5203.52	-2334.1
Spearfish Sandstone	-5868.18	-4527.06	-6020.56	-4655.14
Tyler Sandstone	-13339.7	-10670.5	-13419.9	-10727.9
Kibbey Lime Limestone	-16887.3	-6412.12	-17345.8	-6847.8
Kibbey Lime Shale	-2863.79	722.2739	-2929.93	715.076
Charles Sandstone	-23027	-15903.2	-23845.6	-16535.8
Charles Limestone	-14330.8	-10321.9	-14567.1	-10574.6
Ratcliffe Sandstone	-12378.3	-6249.13	-12729.8	-6585.07
Base Last Salt Limestone	-14605.4	-8943.01	-14638.7	-9008.38
Base Last Salt Sandstone	-6853.47	-4081.14	-6941.69	-4140.9
Mission Canyon Limestone	-4535.67	-3412.12	-4554.22	-3399.94
Lodgepole Limestone	-2869.99	-1593.38	-2837.36	-1551.74

5.5 Conclusions

Even though Drilling optimization is a key area of research, the effect of metrics for drilling optimization remains relatively unexplored. This chapter evaluated three metrics for drilling optimization by changing the objective function and evaluating its effect on other key performance indicators (KPI) used in drilling. Based on the objective function being optimized, the results vary.

Cost-per-foot is not a good objective function for algorithmic optimization since it translates to ROP optimization. Good engineering decisions and improved management practices can reduce the aspects of cost-per-foot not covered by ROP. The best objective function to use for a well will depend on several factors such as the operator, the costs of drilling, equipment, operating costs, history of drilling dysfunctions.

ROP optimization has been the workhorse of this industry for many decades. Several downsides of ROP optimization have been established, however, it is still the most popular metric used for drilling optimization. Based on the analysis in this chapter, when ROP is used as an objective function and maximized, it results in increased MSE and increased torque responses. This may lead to the non-optimal use of bit energy, excessive vibrations, and drilling dysfunction which can offset the time saved due to improvements in ROP. Using ROP as an objective function leads to an improvement of ROP by an average of 31%, whereas MSE increases by 4% and torque on average increased by 10%. If the vibration response of the bit is minimal then these settings would provide the maximum ROP increase and lead to considerable time savings. However, in most cases, drilling vibrations inhibit ROP and techniques to deal with vibrations are addressed in Chapter 6.

When TOB is used as an objective function, reduction in TOB resulted in a small ROP improvement and an increase in MSE (which is undesirable). However, in cases with high axial, lateral or torsional vibrations, this might be more desirable since reducing vibrations can help avoid drilling dysfunction. An average of 10% reduction in TOB was noticed with a 16% increase in ROP and 15 % increase in MSE. This setting may be optimal in cases where the bit has excessive vibrations.

MSE was initially introduced in drilling for bit selection but has soon become an important KPI in drilling. It represents the efficiency of the use of energy downhole. Using MSE as an objective function resulted in improved ROP, reduction in TOB, and a reduction in MSE. Given the nature of the MSE equation, just minimizing TOB will lead to an improved MSE since TOB is measured in lbs and ROP in ft/hr – the magnitude of units of

TOB is higher. This can be avoided by thresholding a minimum ROP or by normalizing the units during optimization. Bayesian optimization is used for faster convergence over the PSO algorithm. Optimizing using the MSE objective function lead to an average decrease of MSE by 15%, increase in ROP by 20% and reduction of torque by 7%. By far, using MSE as an objective function has the most balanced improvement for drilling – an increase in ROP, reduction in torque and MSE at the same time. While the increase in ROP is not as high as compared to purely optimizing ROP, there is an improvement in MSE and TOB which have potential to increase the longevity of the bit.

Chapter 6: Classification of drilling vibrations using machine learning

Previous chapters have covered the prediction, inference, modeling, and optimization of ROP. Chapter 5 discussed the definition of an objective function and its optimization for drilling. However, a key component of drilling optimization not addressed so far is drilling vibrations. This chapter introduces the use of classification algorithms to model and classify the intensity and effect of drilling vibrations – a major inhibitor.

6.1 INTRODUCTION

A significant part of ROP optimization is the mitigation of drilling vibrations since they are a major ROP limiter (Fred E Dupriest & Koederitz, 2005; Macpherson, Mason, & Kingman, 1993). Drilling vibrations can be largely divided into axial (vertical), lateral (whirl), and torsional vibrations (stick-slip). Over 40% of drilled depth per year is affected by drilling vibrations and ROP improvements are observed when they are mitigated (Abdul Rahman et al., 2012; Bailey et al., 2008; Janwadkar et al., 2006). Low levels of vibrations have also been identified to limit weight-on-bit (WOB), ROP, and borehole quality (D. Ertas, Bailey, Wang, & Pastusek, 2013).

Such adverse effects of vibrations have resulted in the development of static, basic and enhanced electrodynamic, and numerical modeling techniques (such as finite element or finite difference) to analyze the dynamic behavior of the drill string (Ghasemloonia, Rideout, & Butt, 2015). Vibration issues are commonly addressed via bit, BHA or drill string redesign which are often expensive. A cheaper alternative is controllable drilling parameter management. Axial vibrations can commonly be avoided with good bit design in the planning phase (D. Ertas et al., 2014) or the use of polycrystalline diamond compact (PDC) bits. Lateral vibration modeling and mitigation has been carried out extensively by ExxonMobil (Bailey et al., 2008), concluding that BHA design can help significantly mitigate lateral vibrations. Stick-slip mitigations using BHA and bit improvements have been proposed (Davis, Smyth, Bolivar, & Pastusek, 2012; Jaggi, Upadhaya, & Chowdhury, 2007; Janwadkar et al., 2006; Mahyari, Behzad, & Rashed, 2010; Pastusek, Brackin, & Lutes, 2005; Zhu, Tang, & Yang, 2014). Yet a cheaper alternative is to use combinations of drilling input parameters to "break out" of the harmonic vibrations or stick-slip such as increasing rotary speed (RPM) and reducing weight-on-bit (WOB) (Macpherson et al., 1993) or just reduction of WOB (Patil & Teodoriu, 2013). The use of real-time drilling parameter control for reduction of stick-slip with active monitoring has been successful (Dufeyte & Henneuse, 1991; Pavone & Desplans, 1994; Robnett, Heisig, McGinley, & Macpherson, 2002; Shuttleworth, Van Kerkoerle, Folmer, & Foekema, 1998). However, these methods depend more on the reservoir, BHA, geology, heterogeneity, etc. and as such cannot be generalized. The use of active control systems (Kyllingstad & Halsey, 1987) was introduced; it has been improved to model stick-slip vibrations using surface data using the transfer matrices approach (D. Ertas et al., 2014). This stick-slip modeling method (which makes some linearity assumptions) has been incorporated into a drilling optimization workflow by ExxonMobil called the drilling advisory system (Chang et al., 2014).

The use of vibration modeling and monitoring is a complex process often requiring many simplifying assumptions; they may not always generalize to different BHAs, reservoirs, geology, and formations. Additionally, physical processes such as bit wear, heterogeneity, cuttings loading, buckling of the drill string – which are continuously changing during drilling – can violate many simplifying assumptions. In such cases, modeling vibrations using a data-driven approach specific to a particular geology, BHA, and formation may be more lucrative. Furthermore, if the primary aim of modeling of vibrations is to identify optimal drilling input parameters for improving ROP and reducing vibrations: a data-driven approach can provide an excellent alternative modeling scheme.

This chapter introduces a simple and effective method to model the effect of axial, lateral and torsional drilling vibrations using machine learning. A data-driven approach is used given its simplicity, high accuracy, versatility, and ease of coupling with other ROP data-driven models (C. Hegde et al., 2017; Chiranth Hegde & Gray, 2017). The objective of this chapter is to show that machine learning classification algorithms can be used to accurately classify the effect of drilling vibrations with a high accuracy. This classifier can be coupled with a ROP optimization model (be it data-driven or not) to optimize ROP while keeping drilling vibrations to a minimum as shown in Chapter 7.

Since the objective is to control the vibrations, a classification approach is introduced wherein different classification algorithms – logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), Gaussian mixture models (GMMs), and random forests are used to classify the effect of vibrations in each formation. Common issues related to machine learning classification such as an imbalance of data sets and choice of metrics have been addressed. Stick-slip index (SSI) is used as a metric to define the effect of torsional vibrations on a drilling system. Axial and lateral vibrations are can be classified based on their readings – a separate metric is not required since the accelerometer readings directly translate into drilling dysfunction and tool failure indices (Y. I. Arevalo, Medina, & Naslausky, 2011). SSI has been effectively classified using classification algorithms with high accuracy; alternative metrics (F-1 score and area under the receiver operating characteristic curve) which better define success in classification are used to pick the best algorithm. The random forest algorithm performs extremely well (average F-1 score of 0.9) to successfully classify SSI into pre-defined binary classes. Axial and lateral vibrations are also classified by the random forests classifier with high F-1 scores. This algorithm can be easily implemented into a ROP optimization framework to provide constraints for selecting optimal control parameters (discussed further in Chapter 7).

6.2 Theory: Drilling Vibrations

6.2.1 Types of drilling vibration

Drilling vibrations can be classified into torsional vibrations (stick-slip), axial vibrations (bit-bounce phenomenon), and lateral vibrations (whirl) as shown in Figure 6.1.

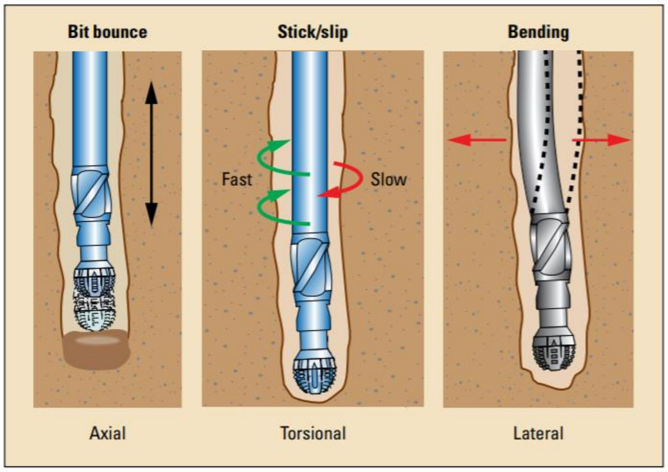

Figure 6.1: Visualization of different types of vibrations (*Drillstring Vibrations and Vibration Modeling*, 2010)

6.2.1.1 Torsional Vibrations

Downhole measurements have shown that not all the rotational energy used to rotate the drill string is transferred to the bit. The bit exhibits large fluctuations in RPM most of the time. This rotational motion is due to the cuttings build up at the bit and the torsional flexibility of the drill string which causes a non-linear relationship between torque and angular velocity at the bit (Jansen, 1991). The RPM during such events has been visualized in Figure 6.2. Torque at the bit as compared to the surface follow similar trends but can vastly differ in magnitude (Figure 6.3). It is difficult to predict downhole conditions by purely relying on surface drilling data. As visualized in the figures, energy builds up at the bit and is released. At times the buildup of energy is so severe that the bit momentarily comes to rest before it unwinds with excessive energy which may lead to the bit rotating in the opposite direction. This event is referred to as a stick-slip. A useful measure or metric used to represent the effect of stick-slip is the stick-slip index (SSI) as described in Equation 6.1.

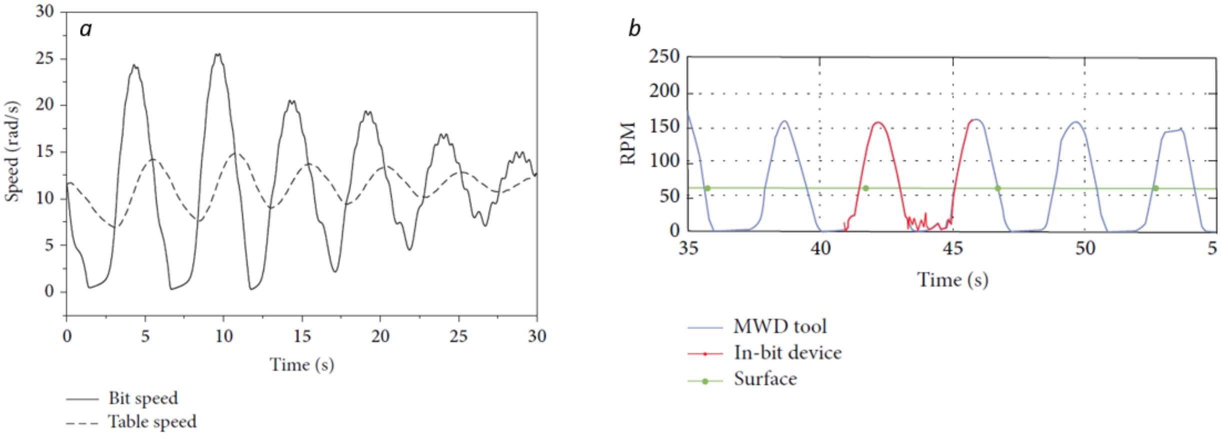

Figure 6.2: (a) Stick slip vibrations as seen at the bit versus that seen at the surface (or table speed)*(Christoforou & Yigit, 2001)* ; (b) RPM during stick-slip events measured using surface and downhole sensors(Ledgerwood III et al., 2013).

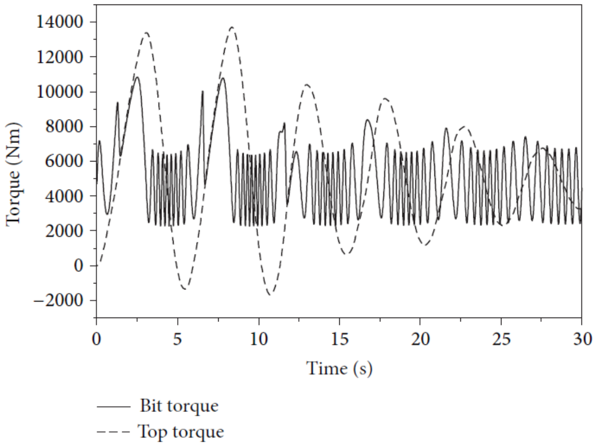

Figure 6.3: Bit torque vs surface torque during a stick-slip *(Christoforou & Yigit, 2001)*

6.2.1.2 Axial Vibration

This vibration mode is due to irregular movements of the drill string along its vertical axis causing bit-bounce which may result in tooth wear and bearing failure (Li & Guo, 2007). Additionally, due to coupling mechanisms, it may excite lateral vibrations of the string (Shyu, 1989). The bit-bounce pattern can be detected at the surface; axial vibrations are prolific with the use of roller-cone bits. It is more common during drilling of hard formations. They may cause accelerated bearing and tool wear, seal failure, broken tooth cutters, and reduction in the ROP; these are the most common catastrophic outcomes of axial vibration and bit bounce (Ashley, McNary, & Tomlinson, 2001). However, the use of polycrystalline diamond compact (PDC) bits reduces axial vibrations during drilling.

6.2.1.3 Lateral Vibration

Lateral vibrations are difficult to detect on the surface and can be considered one of the most lethal forms of drilling vibration (Marquez, Boussaada, Mounier, & Niculescu, 2015). Bending waves are not propagated up to the surface via the drill string as are torsional and longitudinal waves – due to the difference in the wave speed for different types of modes (Ghasemloonia et al., 2015). Hence, the modeling of the vibration modes, extracting the natural frequencies and analyzing the dynamic behavior of the BHA is important for failure prevention. The rotating BHA interacts with the borehole wall which generates shocks to form lateral vibrations. The collisions with the borehole wall may result in hole eccentricity. These shocks can severely damage the bit (Figure 6.4). They may also result in modification of the drilling direction with repeated shocks to the wellbore (Jansen, 1993). The lateral vibration behavior of the drill string is strongly influenced by BHA vibration and its prevention includes the effective design of BHA (Bailey et al., 2008).

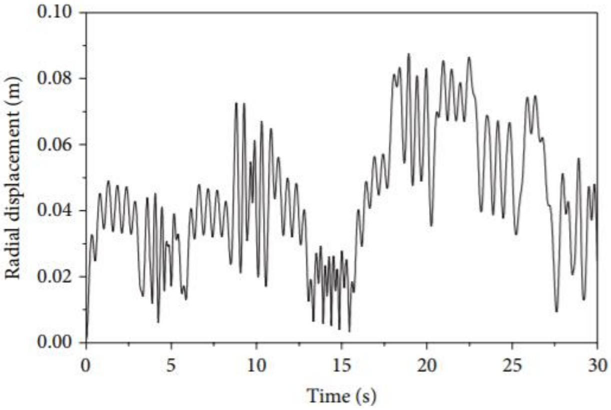

Figure 6.4: Vibrations during lateral motion of the drill string *(Christoforou & Yigit, 2001)*

6.2.2. Vibration Frequencies and Resonances

Natural frequencies are frequencies at which a structure can vibrate harmonically. If a structure is excited at one of these frequencies then large oscillations are observed, the largest of which occur at the fundamental frequency (Thomson, 1996) as shown in figure 6.5. A drill string may get excited due to load or displacement excitations at various points along the drill string. Vibration modeling (time or frequency domain) can be utilized to model critical drilling RPM which can result in resonant frequencies for the current BHA configuration. The critical RPM is the surface RPM at which the frequency of excitation will correspond to the natural frequency of the BHA.

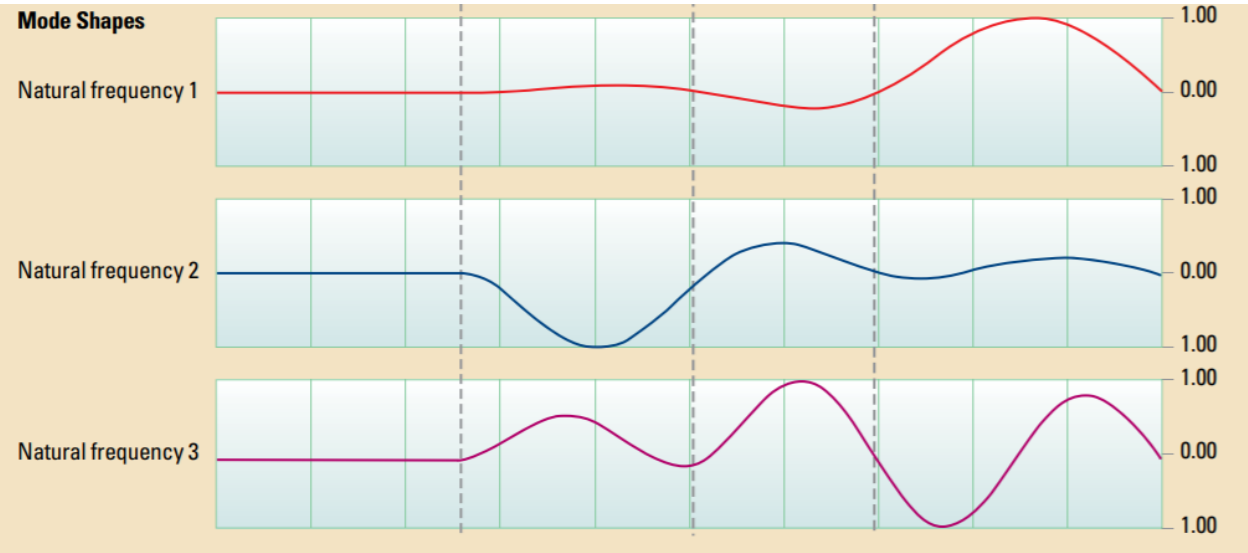

Figure 6.5: Lateral vibration mode shapes (Drillstring Vibrations and Vibration Modeling*, 2010)*

6.2.3 Measurement of Vibrations

Vibrations are measured downhole using an accelerometer. The three axes refer to a system comprising the vibration-acquisition board and three off board accelerometers (Y. Arevalo & Fernandes, 2011) as a part of the measurement while drilling (MWD) tool. The three accelerometers are mounted in a mutually orthogonal arrangement along the central axis of the tool. The X-axis sensor measures axial shocks, the 'Y' and 'Z' sensors measure lateral shocks in orthogonal directions. Data can be transmitted in real-time using downhole telemetry. A commonly used metric to measure the effect of torsional vibrations is the stick-slip index (SSI) which can be calculated using downhole measurement tools as shown in Equation 6.1. SSI is a convenient method to represent stick-slip since a value of 1 would represent full stick-slip, i.e. the bit is momentarily stationary. At higher values of SSI, the bit stop turning for a period of time or is said to be “stuck”(D. Ertas et al., 2014). Lateral and axial vibration intensities do not require a metric and the measurement from the downhole accelerometer can be directly used for classification.

$$SSI = \frac{Max(bit\ RPM) - Min(bit\ RPM)}{Average\ (bit\ RPM)}\ \text{, (Equation 6.1)}$$

Axial vibrations are not generally a problem while using PDC bits. However, an axial severity estimate (ASE) has been defined in literature (M. D. Ertas et al., 2015) which measures the axial severity estimate of the drill string (Equation 6.2).

$$ASE = \frac{Max(bit\ WOB) - Min(bit\ WOB)}{Average\ (bit\ WOB)}\text{, (Equation 6.2)}$$

However, the effect of axial and lateral vibrations on the drill string and tool failure has been better studied using simpler metrics: the acceleration of the drill string vibrations (or g’s as reported by the downhole accelerometers). Studies (Y. Arevalo & Fernandes, 2011) have shown that capping lateral and axial vibrations at less than 1.0 g’s results in a reduction of tool failure by 80%.

6.3 THEORY: MACHINE LEARNING CLASSIFICATION

A function can be estimated in terms of its input parameters using a model: $Y = f(X_1, X_2, .., X_n)$. Algorithms used to determine the unknown function are often grouped as classification algorithms (ones used to determine a qualitative result) and regression algorithms (used to determine a quantitative result). Qualitative variables take on values of different classes (vibration is low or high) or categories and quantitative variables take on numeric values (value of ROP or MSE). This section will introduce the use of classification techniques to classify drilling vibrations.

Classification algorithms estimate the class of the output based on input parameters. For example, drilling vibrations are to be classified as low or high using measured while drilling (MWD) drilling parameters such as rate of penetration (ROP), rotations per minute (RPM), weight on bit (WOB), flow-rate (Q) and torque (TQ) as inputs. A classification algorithm can be used to estimate the severity of vibrations based on MWD data. A classifier (classification algorithm) seeks a way to determine if the drilling vibrations will be low or high based on data. Figure 6.6 illustrates classification of torsional drilling vibrations into two classes.

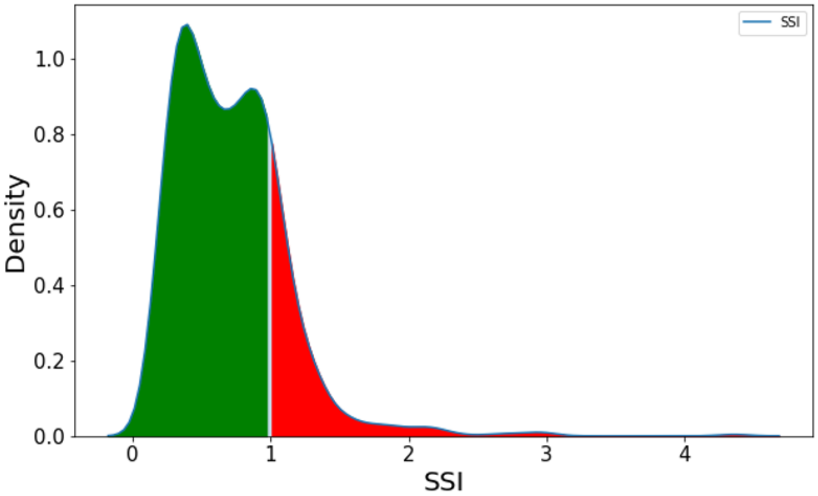

Figure 6.6: Density plot of the SSI of torsional drilling vibrations in the Tyler sandstone formation. The vibrations have been classified into two distinct classes colored green (low SSI <1) and red (high SSI > 1). A classifier is used to determine the class of vibrations (low or high) based on input MWD parameters.

Commonly used classification algorithms are logistic regression, linear discriminant analysis, support vector machines, Gaussian mixture models, random forests, and neural networks. Neural networks and deep neural networks perform well for high dimensional data such as image classification; they will not be discussed in this chapter (since drilling data is low dimensional); the reader is referred to other resources for the application of neural networks for classification (Krizhevsky et al., 2012; LeCun et al., 2015; Ng et al., 2015).

6.3.1 Logistic Regression

Logistic regression is a classification technique used to model the odds of an event. It models the probability that an output Y belongs to a particular class: probability that the vibrations are low or high. The mathematical notation can be written as: $P(Y = y|X_1, X_2, .., X_n)$ or $P(SSI = low|ROP, MSE, RPM, .., Q)$. The values of probability will range from 0 to 1. Hence, with a given set of input parameters the probability of a particular class can be computed (James et al., 2013).

In a linear regression setting this would be modeled as: $P(y) = w_1X_1 + w_2X_2 +, .., w_nX_n$ or concisely $P(y) = W^TX$,where the input parameters $X_1, X_2, .., X_n$ are represented as a matrix X and $w_1, w_2, .., w_n$ are represented as the vector W (Goodfellow et al., 2016). In the case of logistic regression, the expression W^TX is embedded within a sigmoid function (Equation 6.3) which forces the values of the outputs (*P(y)*) to lie between 0 and 1. This transforms the regression equation to yield the logistic regression equation (Equation 6.4).

$Sig(y) = \frac{1}{1+e^{-y}},$ (Equation 6.3)

$P(y) = Sig(W^TX) = \frac{1}{1+e^{-W^TX}} = \frac{e^{W^TX}}{1+e^{W^TX}} = \frac{e^{w_1X_1+w_2X_2+,..,w_nX_n}}{1+e^{w_1X_1+w_2X_2+,..,w_nX_n}}$ (Equation 6.4)

The weight matrix (W^T) can be determined using the principle of maximum likelihood (James et al., 2013) using training data. The maximum likelihood function estimates the values of W for which the training data (*X*) are most likely. The probability of a particular class occurring can then be calculated directly using Equation 6.4.

An extension or variant of logistic regression which can be useful is principal components analysis (PCA) logistic regression – logistic regression on the principal components of the data – similar to PCA regression (C. M. Hegde, Wallace, & Gray, 2015). The first principal component of a dataset is calculated by determining the direction along which the data vary the most. The second principal component orthogonal to the first component calculates the direction along which the remaining data varies the most. PCA is used as a dimension reduction tool. Additionally, it can also serve to congregate the most important features of the data into a low dimensional form. Equation 6.4 can be modified by replacing X_i with Z_i where Z_i represents the i$^{\text{th}}$ transformed component for PCA logistic regression.

6.3.2 Discriminant Analysis

Linear discriminant analysis (LDA) is a classification method which works by flipping Bayes' rule to estimate $P(SSI = low|ROP, MSE, RPM, .., Q)$. It tends to work well when data are close to normally distributed. LDA is more stable than logistic regression when classes are well distributed, training sets are small, and when the output contains more than two classes (James et al., 2013). Probability of a given class can be estimated using LDA using Equation 6.6.

$P(Y = k|\, X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$, (Equation 6.6) where, π_k is the prior probability of class 'k'; this is the probability that a given sample belongs to class 'k'. And $f_k(x)$ is the class conditional density of X in the class K='k'. In general, π_k can be estimated using the fraction of training data that belong to class 'k'. However, the calculation of $f_k(x)$ is more challenging; it is assumed to follow a Gaussian distribution (Equation 6.7) to make calculations tractable.

$$P(Y = k|\, X = x) = \frac{\pi_k N(\mu_k,\sigma_k^2)}{\sum_{l=1}^{K} \pi_l N(\mu_l,\sigma_l^2)}, \text{ (Equation 6.7)}$$

where, $N(\mu_k, \sigma_k^2)$ is a normal distrbition with a mean of μ_k and variance σ_k^2. The decision boundary for LDA is linear. In cases of nonlinear behavior of the data it may be better to use quadratic discriminant analysis (QDA) which is an extension of LDA, except it assumes that each class has its own covariance matrix (James et al., 2013) – resulting in nonlinear decision boundaries.

6.3.3 Support Vector Machines (SVM)

Logistic regression can breakdown in cases where there exists a linearly separable hyperplane (maximum likelihood fails and some parameters tend to infinity (Efron & Hastie, 2016)). While uncommon, it can happen when the number of predictors is large. SVMs pursue the ability to separate classes with a hyperplane using basis expansions and transformations.

An optimal hyper plane as shown in Figure 6.7 shows a case where two classes can be separated using a hyperplane as a decision boundary. In Figure 6.7 (left) three possible hyperplanes are shown which can be used to classify the data. Out of all possible separating hyperplanes, an optimal separating hyperplane is preferred (Figure 6.7 (right)) since it ensures that the distance between the two classes is maximum (which helps prevent overfitting on test data). If the data are not perfectly separable, a soft margin classifier can be used instead; the soft margin classifier allows some data points to violate the decision boundary (often called cost). This process of finding a hyperplane can be written in the form of an optimization problem as shown in Equation 6.8.

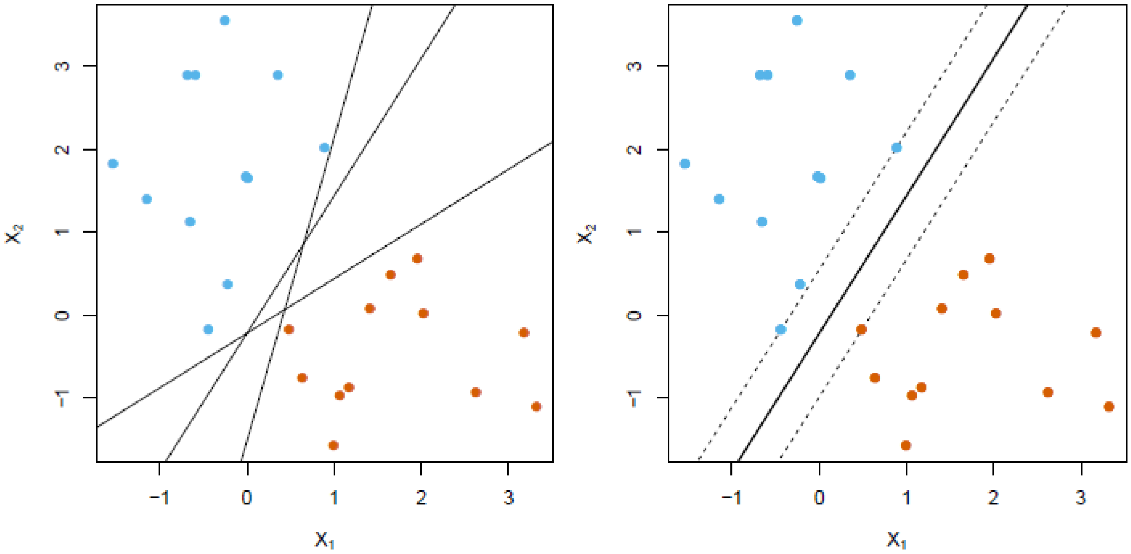

Figure 6.7: Two classes of data are separated using a decision boundary. In this case the decision boundary is a hyperplane of one dimension. (Left) three potential hyperplanes are shown which can be used to classify the data; each classifier can be used to separate the data; (Right) The optimal separating hyper plane which create a maximum margin between the two classes (Efron & Hastie, 2016)

$$minimize_{\beta_0\beta_1} \sum_{i=1}^{n}[1 - y_i(\beta_0 + x_i\beta)] + \lambda|\beta|_2^2 \qquad \text{(Equation 6.8)}$$

where, β are the weights, λ is a regularization parameter, and $[1 - y_i(\beta_0 + x_i\beta)]$ is the cost for being on the wrong side of the margin. β can be rewritten as an inner product of the input features or in other words a kernel or basis function on the input features can be used before feeding it to Equation 6.8; this expands the application of these hyperplanes to non-linear data. The hyperplane of a support vector machine can be non-linear depending on the kernel used to perform the dot product; popular kernels include linear, polynomial, sigmoid and radial kernels.

6.3.4 Gaussian Mixture Models (GMMs)

A Gaussian mixture model (GMM) attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset. GMMs are an unsupervised classification technique often used for kernel density estimation. However, the can be modified to be used as a supervised learning algorithm for the classification of SSI by fitting a GMM for each class. The GMM model can be fit to a set of data using the expectation maximization theorem.

Intuitively the algorithm is similar to K-means clustering: it uses an expectation–maximization approach which qualitatively does the following:

1. Choose starting guesses for the location and shape
2. Repeat until converged:
    1. *Expectation-step*: for each point, find weights encoding the probability of membership in each cluster (or class)
    2. *Maximization-step*: for each cluster, update its location, normalization, and shape based on *all* data points, making use of the weights

The algorithm fits the data with a smooth Gaussian model (not a sphere like K-means). Just as in the *k*-means expectation–maximization approach (James et al., 2013), this algorithm (being greedy) can sometimes miss the globally optimal solution, and thus in practice multiple random initializations are used (VanderPlas, 2016). The unsupervised learning algorithm can be modified to fit a classification role: test data are assigned to the most likely gaussian cluster.

For the data analyzed in this paper, the data belong to two classes – low and high vibrations – and this GMM model attempts to find the underlying distribution of each class. Once the underlying probability distribution of each class has been estimated, new data can be automatically classified to one distribution over the other based on the probability that they are drawn from either distribution. GMMs have a special property; since they estimate the underlying probability distribution of the data as a mixture of gaussians, they are generative models. This means that new data – which have not been observed yet – can be generated using a GMM model (which is useful for simulation).

Assume that a total of 'K' classes is present in the population of vibrations (in this case 2) and a total of 'N' training data samples. Each class 'K' can be estimated using a gaussian distribution with a mean (μ_k) and variance (σ_k^2). The probability that a datum from this population (of vibrations) belongs to a given class (low or high) can be categorized using a mixture coefficient or the prior (π_k) with a constraint – $\Sigma\, \pi_k = 1$ – so that the probability distribution is normalized. The probability that a datum belongs to a given distribution can be calculated using Equation 6.8.

$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k^2)$, (Equation 6.8)

where, Σ is the covariance matrix of the two classes. Typically, for classification problems data are assumed to come from '*K*' gaussian distributions (number of gaussians = 2 for estimation). This can be improved by assuming that each class comes from a mixture of gaussians, expanding the role of this classifier to better estimate highly nonlinear dsitributions. Hence each class is fitted with an 'M' component mixture of gaussian where 'M' is chosen using the Bayesian information criterion (BIC). During classification, the normalized likelihood ratio can be used to determine class assignment. This method can outperform a vanilla GMM model for nonlinear data.

6.3.5 Random Forest Classifiers

The random forest algorithm is an ensemble based machine learning algorithm that performs well for low dimensional drilling data (Chiranth Hegde & Gray, 2017). It is made up of averaging many decision trees. A decision tree is a nonlinear classification algorithm but suffers from high error, large variance and generally overfits the data as discussed in Chapter 2. This is solved by averaging many decision trees – built on different bootstrapped datasets – to effectively reduce the variance and improve the predictive power of the algorithm. Additionally, at each step in the tree building process, a subset of input features is arbitrarily chosen; number of features chosen is modeled as a hyperparameter often determined using cross-validation. This randomization of feature selection decorrelates the input trees which helps improve its accuracy. Classification can be performed by selecting the class using the majority voting scheme. Additionally, a probability can be calculated by measuring the number of trees which advocate for a particular class. This algorithm can be used to measure the importance of each feature vector in classification – a feature ranking – which is extremely useful for model inference. Feature importance is calculated by measuring the entropy loss due to a split of a given input variable in a decision tree (averaged over all trees).

6.3.6 Ensemble Models

Ensemble models are meta-algorithms – they combine other algorithms – to form one machine learning model. Classification ensemble models use a voting or weighted voting scheme for ensembling models. For example, three machine learning algorithms –

logistic regression, support vector machines and gaussian mixture models – can be ensembled into a single model. The class predicted by the ensemble model will be the majority vote among the three machine learning classes used within the ensemble. A modified version of this algorithm will incorporate weighted voting, where each individual classifier is assigned weights to their votes. For example, if it is believed that the distribution is more non-linear, GMMs and SVMs can be given twice the weight of a logistic regression within the ensemble. Optimal weights can be calculated using cross-validation or using the stacking algorithm for the metric of interest.

6.4 Classification Metrics

In the case of regression algorithms accuracy can be calculated by looking at the $l_1$or l_2 norm. However, classification algorithms have a variety of metrics which need to be chosen depending on the problem. These metrics are based on a confusion matrix which is defined in terms of type-I and type-II errors. A confusion matrix has been shown in Table 6.1. When a classification model is used to predict the class of a test datum, there are a total of four possibilities (assuming the two classes are true and false) : the classifier predicts true and it's actually true – a true positive; the classifier predicts true and it's actually false – a false positive; the classifier predicts false and it's actually false – a true negative; the classifier predicts false and it's actually true – a false negative.

Table 6.1: Table describing a confusion matrix for classification models

Confusion Matrix	**True Condition**	
Predicted Condition	True Positive (TP)	False Positive (FP) or Type I error
	False Negative (FN) or Type II error	True Negative (TN)

6.4.1 Classification Accuracy

Classification accuracy is the simplest metric which can used for classification models. It represents the percentage of test data that are classified correctly, i.e. either a true positive or a true negative as shown in Equation 6.9. Albeit simple, this metric may not always capture all the nuances of classification problems.

$$Classification\ Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \text{(Equation 6.9)}$$

In the case of a class imbalance – where one class occurs much more often than the other class – this metric would fail. Let's assume that the number of vibration measurements which are below the threshold occur 99% of the time, and the number of vibrations measurements which are high occur 1 %. If a classifier has an accuracy of 99.2% it does not imply that the classifier is any good since a baseline classifier which classifies all data as low would result in an accuracy of 99%. Another instance where falls short is when the price or cost of misclassifications vary; the price of a classification error for a false positive as compared to a false negative may be vastly different. In this case, a false positive refers to a situation where vibrations are actually high but are predicted low. If the classifier results in too many false positives, it can lead to drilling dysfunction. However, a false negative implies that low vibration was classified as high; this does not pose the same danger as a false negative, however, represents a missed opportunity. Since accuracy does not differentiate between the two types of misclassifications, it can be an ineffective way to evaluate classification models. If classes are equally balanced and error type is unimportant, this metric provides an easy way to evaluate different classification algorithms.

6.4.2 F-1 Score

The F-1 score provides an alternative to evaluate classification algorithms by addressing the deficiencies of classification accuracy. The f-1 score is a harmonic sum of two metrics called precision and recall. Precision is a measure which calculates the proportion of vibrations that were classified true and are actually true (Equation 6.10). Recall on the other hand calculates the proportion of vibrations that are actually true when predicted true by the algorithm (Equation 6.11). Precision is about being precise, implying that all data which are true have been correctly identified. Recall measures the effectiveness of capturing all the true (or low vibration) cases correctly; a classifier which labels all data as true (or low) will have perfect recall. If the focus of the problem is to minimize false negatives, recall should be as close to 1 as possible without precision being too low. If false positives are to be minimized then precision is to be maximized. In most cases it is preferable to combine precision and recall into a single metric which can be used to evaluate classifiers. This can be done using a harmonic mean called the F-1 score (Equation 6.12) which reaches 1 at its best and 0 at its worst. If either false positives are to be weighted higher than the false negatives F-beta score can be used instead (Baeza-Yates & Ribeiro-Neto, 1999).

$$Precision = \frac{TP}{TP+FP} \quad \text{(Equation 6.10)}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{(Equation 6.11)}$$

$$F1\ score = \frac{2*Recall*Precision}{Precision+Recall} \quad \text{(Equation 6.12)}$$

6.4.3 Receiver Operator Characteristic (ROC) curve

Recall is often called the sensitivity of a classifier. Specificity measures the proportion of negatives that were predicted correctly (Equation 6.13). Specificity is the exact opposite of recall.

$$Specificity = \frac{TN}{TN + FP} \qquad \text{(Equation 6.13)}$$

ROC analysis (Fawcett, 2006) plots the true positive rate (or recall) on the Y-axis and the false positive rate (or 1 - specificity) is plotted on the X-axis. The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the ROC curve (AUC). An ideal ROC curve will hug the top left corner, so the larger area under the curve implies a better classifier.

6.5 Class Imbalance

In many classification problems, classes are rarely evenly split; one class is generally more populous than the other(s). This imbalance can be severe in some cases such as outlier detection (99:1). Classification with highly imbalanced classes leads to an underestimation of conditional probabilities of the minority class as shown in Figure 6.8. A simple strategy is to attempt to rebalance the classes to improve the performance of the classifier. Some strategies for rebalancing classes are:

- Undersampling: all the observations from the minority class are retained. The majority classes are sampled without replacement; the number of samples corresponds to the size of the minority class.

- Upsampling: a new fixed size – one between the sizes of the minority and majority classes are chosen. The minority class is sampled with replacement and the majority class is sampled without replacement.
- Negative downsampling: different samples sizes are used in this procedure. All observations from the minority class are retained. A different number of observations from the majority class are sampled without replacement.
- Weighting: the cost function can be reweighted to account for the class imbalance. In the case of logistic regression, a weighted cross entropy can be used. The weight can be determined using cross-validation.

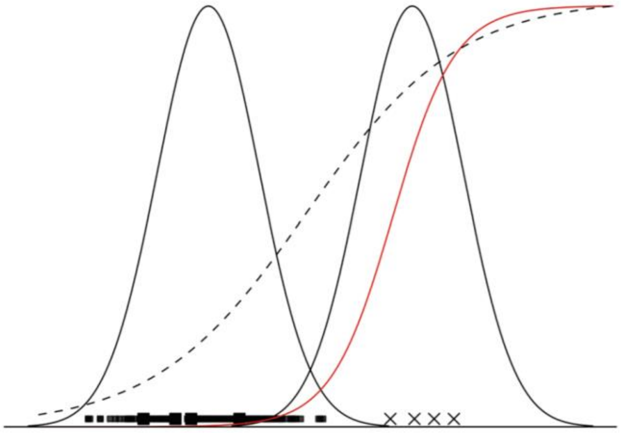

Figure 6.8: The effect of fitting a classifier to imbalanced data sets. The minority class (right distribution) has been represented by crosses and the majority class (left distribution) using blocks. The red curve refers to a fit when classes are imbalanced and the dotted curve when the classes are balanced. Vertical height represents the probability that an observation belongs to the minority class (B. C. Wallace & Dahabreh, 2014).

6.6 METHODOLOGY

In this section, the effect of drilling vibrations on dysfunction, specifically, SSI are classified into two classes: low and high. This method can easily be extended to multiple classes – low, medium or high. The dataset described in Chapter 2 has been utilized for model evaluation and validation. A scatter plot of torque versus WOB (colored by SSI) is shown in Figure 6.9. Since the vibration data, in this case, are imbalanced, a procedure to balance the classes (described in section 6.5) can be utilized to improve the accuracy of the classifier. Classification of lateral and axial vibrations are carried out in the same manner, except the classification is performed directly on the accelerometer reading – a metric such as SSI is not required. Figures 6.10 and 6.11 show a scatter plot of torque versus WOB (colored by axial vibration and lateral vibration respectively).

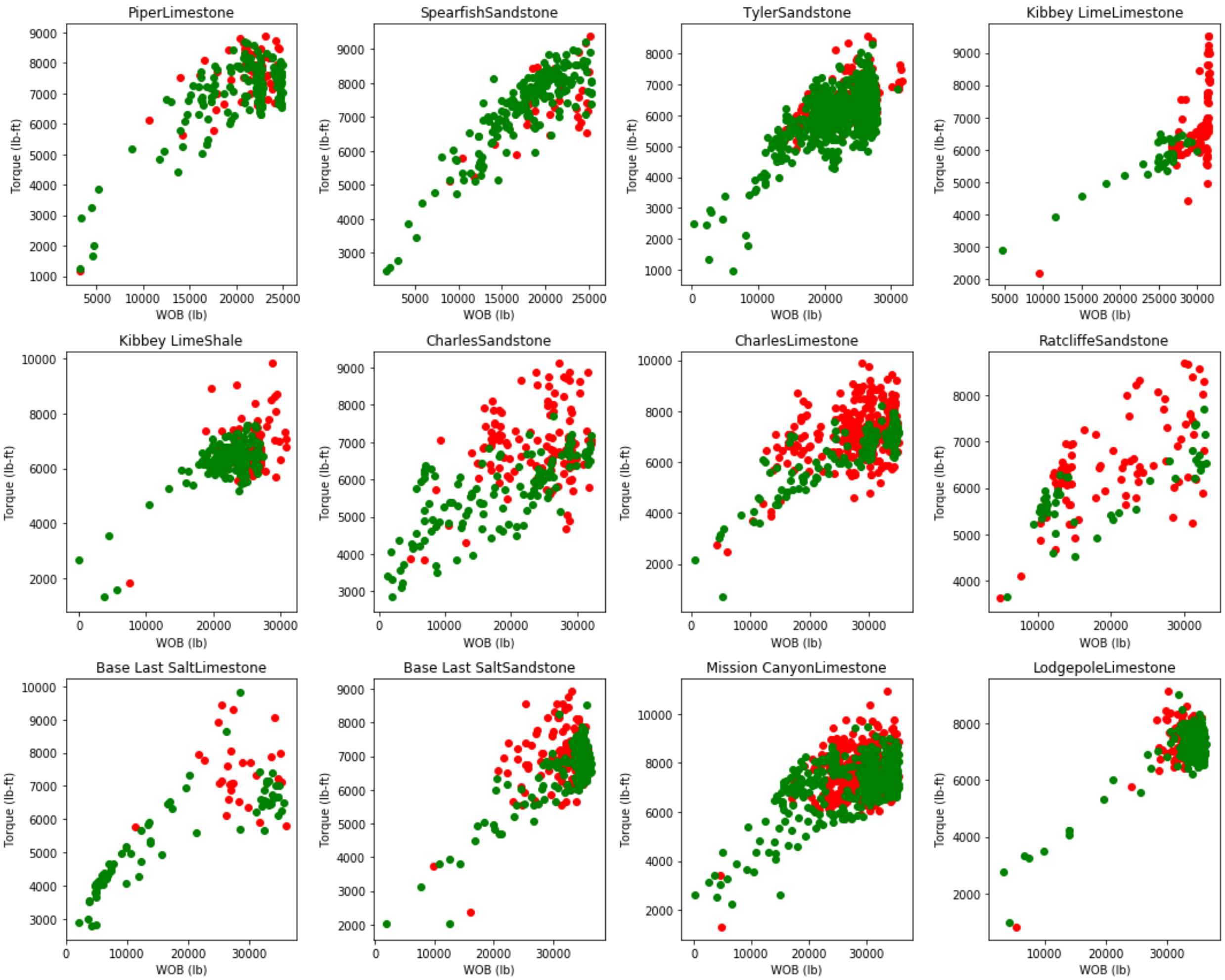

Figure 6.96: Scatter plot of torque vs WOB for 12 formations in the dataset; Data are color-coded based on the intensity of SSI (green<1, red>1); Some data are easy to classify such as data in Kibbey Limestone where a threshold on WOB will suffice. However, in most cases, a non-linear classification algorithm is necessary to classify the data. In a few cases, the two classes are intermixed which increases the difficulty of classification.

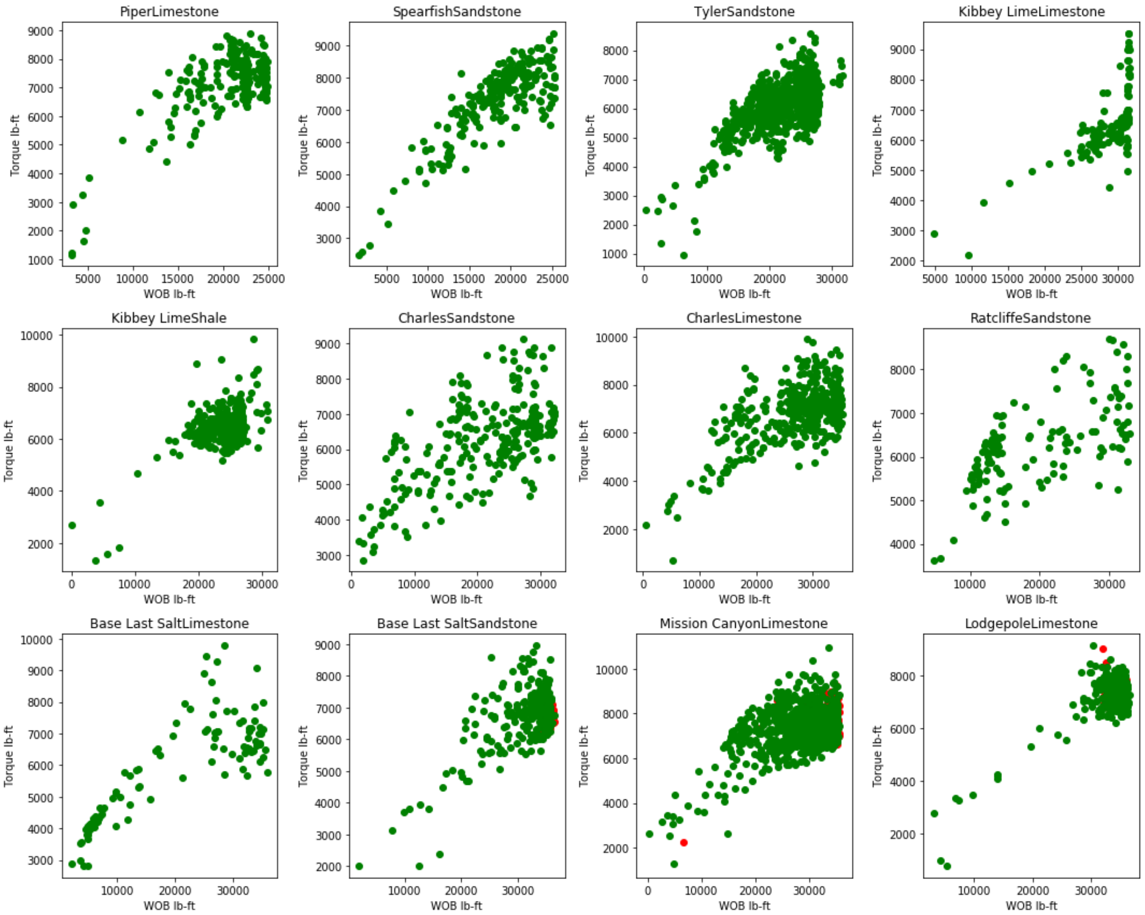

Figure 6.10: Scatter plot of torque vs WOB for 12 formations in the dataset; Data are color-coded based on the intensity of axial vibration (green<0.75, red>0.75); Most data in this dataset are in the safe threshold for axial vibrations. In some cases like Lodgepole limestone, a nonlinear classifier may be necessary for separating the two classes.

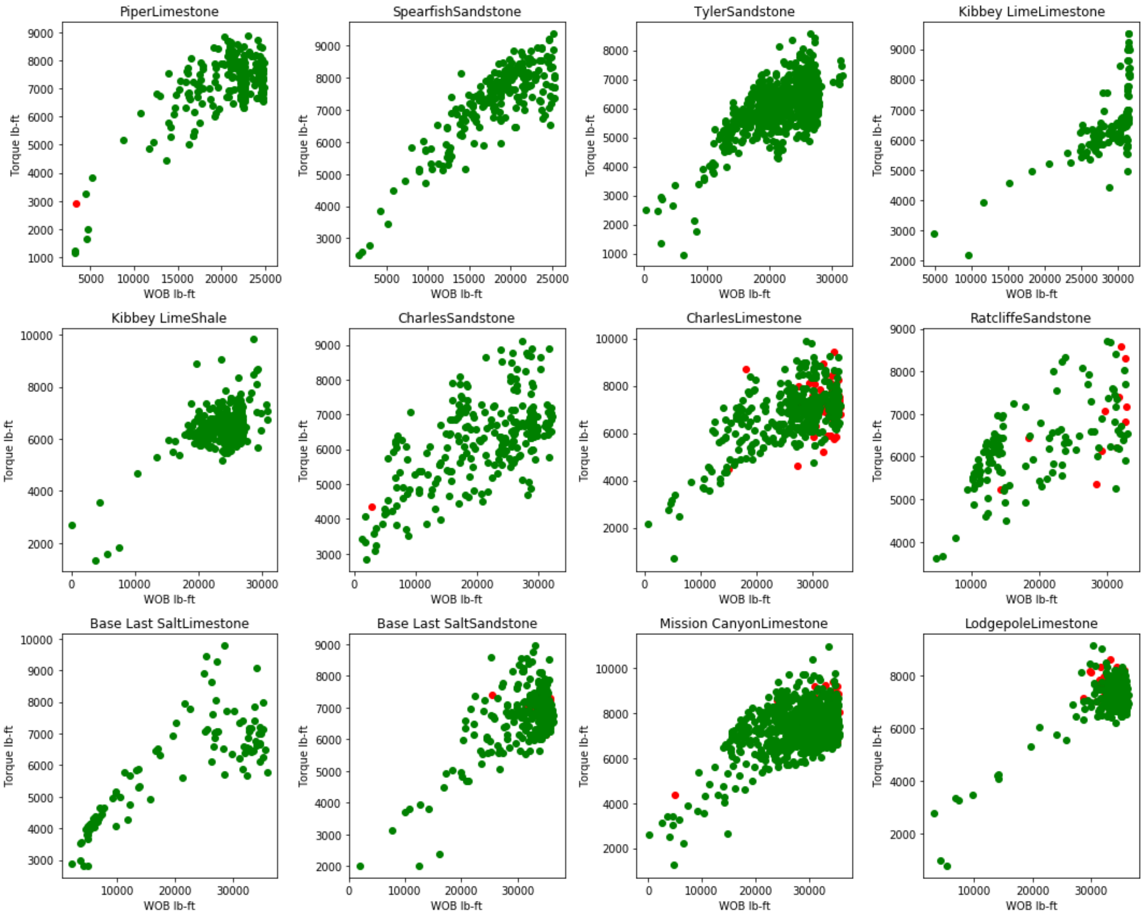

Figure 6.11: Scatter plot of torque vs WOB for 12 formations in the dataset; Data are color-coded based on the intensity of lateral vibration (green<1, red>1); Most data in this dataset are in the safe threshold for axial vibrations. In some cases, like Lodgepole limestone, a nonlinear classifier may be necessary for separating the two classes.

6.6.1 Model

A classification algorithm is used to classify the SSI of torsional vibrations (or accelerometer readings of axial and lateral vibrations). A binary case has been evaluated in this paper where a pre-determined threshold is used to separate the data. The threshold used in this paper is 1; if the SSI is greater than 1 the data are classified as high (red in Figure 6.9) and if it is less than 1, the data are classified as low (green in Figure 6.9). The classification model is built using several classification algorithms which are evaluated for their efficacy. The classification models are built using drilling operations parameters as input data – WOB, flowrate, RPM, ROP, and torque-on-bit (TOB). The actual partition between the two classes can be simple (limiting WOB in the case of Kibbey Lime limestone separates the two classes) or it can be complicated (Tyler sandstone would require a non-linear decision boundary). Estimates for ROP and TOB – using a deterministic (Soares et al., 2016) or machine learning model (Chiranth Hegde et al., 2015) – can be used in place of actual measurements for classification. Once a model is built, the model can be used to predict the class of the SSI for a new set of input parameters for the test data. The use of a classification model inside a ROP optimization model can lead to a safe increase of ROP without excessive vibrations as shown in Figure 6.9.

The exact same process that is carried out for classification of SSI is used to classify axial and lateral vibrations. All three vibration classification methods can be used together to ensure that drilling dysfunction due to vibrations is avoided during drilling.

6.6.2 Experiments

Simulation-based experiments are conducted to evaluate and compare different classification algorithms. Models are built individually on each formation. The vibration data collected in a given formation are divided into training and test sets; classification algorithms are fine-tuned using cross-validation. The test metric is calculated by evaluation of each algorithm on the held-out test set. The data were not severely imbalanced in this dataset. However, this is imbalance is dependent on the threshold used for class assignment, using a higher or lower threshold could lead to a major class imbalance in which case techniques discussed in Section 6.5 can be employed. Correcting class imbalance helped improve classification metrics such as F-1 score and AUC for data which were imbalanced towards higher SSI (fewer data points for low SSI).

Fine tuning models involve feature selection (which input features are to be retained for classification) and hyperparameter selection. For logistic regression, a simple hypothesis test on the coefficient of an input feature can be used for feature selection similar to linear regression. Only input features with a p-value lower than the critical p-value (0.05) are retained in the model. There are no hyperparameters for logistic regression. PCA logistic regression follows the same protocol as logistic regression, except the input features, are rescaled (based on their principal components). The priors for LDA are chosen based on training set ratios and dimensions are set based on a number of classes (in this case 2). QDA incorporates a full covariance matrix for both classes. GMMs are built assuming the data is drawn from a Gaussian of two mixtures since it is binary classification. The random forest algorithm utilized 500 bootstrapped samples, a leaf size of 5, and a random feature subset of 3. The ensemble model combined the results of logistic regression, LDA, QDA, random forests, and GMMs using weighted voting (weights determined using cross-validation).

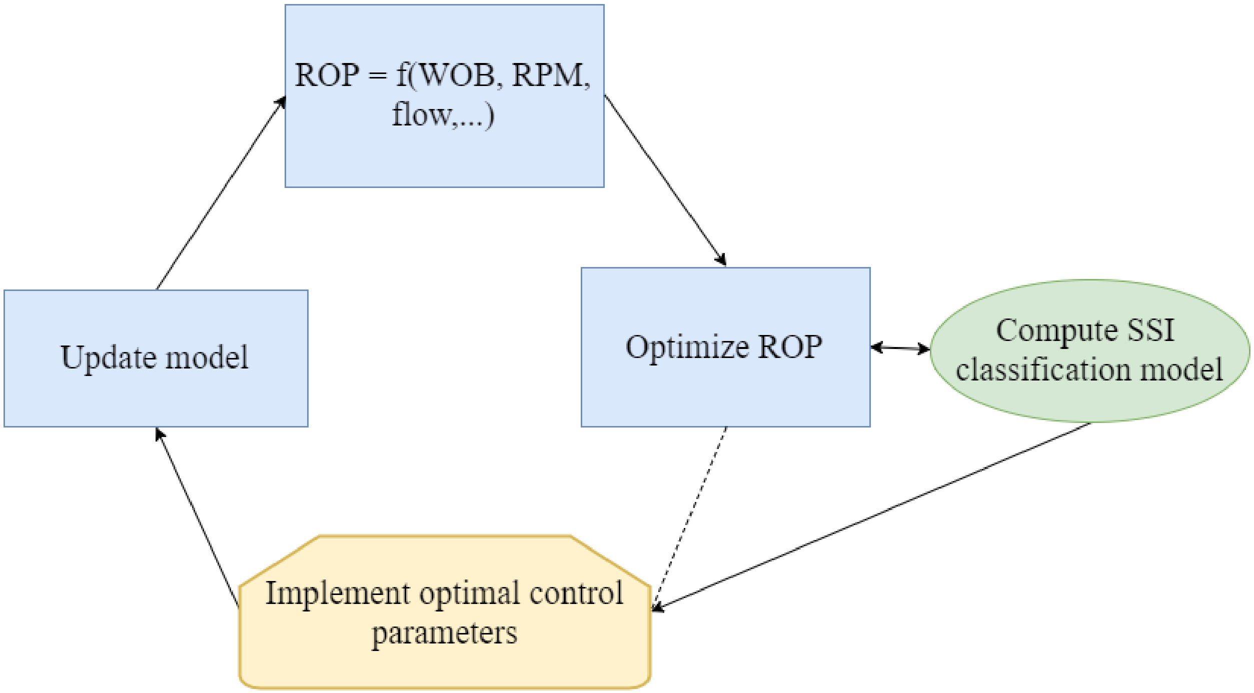
ROP = f(WOB, RPM, flow,...)
Optimize ROP
Compute SSI classification model
Update model
Implement optimal control parameters

Figure 6.12: Flowchart describes the ROP drilling optimization process. The first step is to acquire drilling data by drilling one or more stands into a formation. ROP models are then built on training data. ROP is optimized by manipulating input parameters *(Chiranth Hegde & Gray, 2017)*. These 'optimal' parameters are tested with the Vibration model to ascertain that SSI is below a threshold. The "accepted" optimized drilling control parameters are implemented for drilling the next joint/stand thereby optimizing ROP.

6.6.3 Practical Application and Scalability

One major use of this vibration classification algorithm is within a drilling optimization model as explained in Figure 6.12. This model allows for flexibility and the vibration model can be used as an "add-on" feature to any ROP or MSE optimization model. The optimal parameters predicted using the drilling optimization model is fed into the classification model for approval. The classification model checks if the vibrations are acceptable (by classifying and affirming that they belong to the "low" class) ensuring that the recommended parameters do not result in excessive SSI which may cause drilling dysfunction. From an optimization perspective, this classification model places additional constraints on the optimization space. Previously, optimization spaces were constrained using operational limits or practical limits; this model imposes an additional constraint which includes the effect of drilling vibrations. This feature is incorporated into the drilling optimization workflow to develop an end-to-end drilling optimization model in Chapter 7.

6.7 Results and Discussions: Torsional Vibrations

Models were trained and finetuned using cross-validation. Evaluation metrics have been calculated and plotted using a boxplot as shown in Figure 6.13. The AUC score for different classifiers indicates that random forests outperformed other algorithms. However, most algorithms evaluated in this paper except for GMMs performed reasonably well. Logistic regression with the addition of nonlinear features (such as WOB^2 or $WOB^{0.5}$) yielded lower AUC scores as compared to standard logistic regression. Classifiers from two formations have extremely low F-1 scores; these are classifiers belong to the Ratcliffe sandstone and Mission Canyon limestone formations. Models for these formations can be improved using feature engineering or class balancing and are further analyzed in section 6.7.1. Rather than choosing an overall best model, selecting the best model for each formation often yields better results (further discussed in section 6.7.2). This will also guard against deficient models (for specific formations). Table 6.2 shows the best model for each formation by classification metric. Fine tuning models involves input feature selection (which input features are to be retained for classification) and hyper parameter selection. For logistic regression, a simple hypothesis test on the coefficient of an input feature can be used for feature selection similar to linear regression. Only input features with a p-value lower than the critical p-value (0.05) are retained in the model. There are no hyper parameters for logistic regression. PCA logistic regression follows the same protocol as logistic regression, except the input features are rescaled (based on their principal components). The priors for LDA are chosen based on training set ratios and dimensions are set based on number of classes (in this case 2). QDA incorporates a full covariance matrix for both classes. GMMs are built assuming the data is drawn from a gaussian of two mixtures since it is binary classification. The random forest algorithm utilized 1000 bootstrapped samples (1000 trees), a minimum leaf sample size of 5, and a random feature subset of 3. The ensemble model combined the results of logistic regression, LDA, QDA, random forests, and GMMs using weighted voting (weights determined using cross validation).

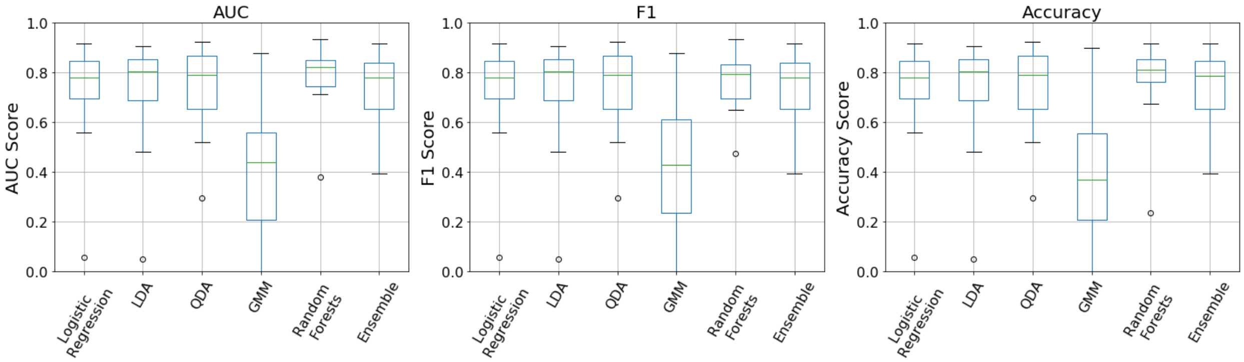

Figure 6.13: Model evaluation for SSI classification on the test dataset

Table 6.2: Tabulation of the best model by metric for each formation for SSI classification

Formation	Best AUC	AUC Score	Best F1	F1 Score	Best Classification	Accuracy
Piper Limestone	QDA	0.69	LDA	0.83	LDA	0.73
Spearfish Sandstone	Logistic	0.74	Random	0.93	Random Forests	0.86
Tyler Sandstone	QDA	0.78	LDA	0.90	LDA	0.81
Kibbey Lime Limestone	LDA	0.97	LDA	0.83	LDA	0.91
Kibbey Lime Shale	Ensemble	0.78	LDA	0.77	Ensemble	0.73
Charles Sandstone	PCA Logistic	0.85	Random	0.77	Random Forests	0.79
Charles Limestone	QDA	0.89	QDA	0.80	QDA	0.81
Ratcliffe Sandstone	QDA	0.82	PCA	0.62	QDA	0.72
Base Last Salt Limestone	Random	0.92	QDA	0.86	QDA	0.81
Base Last Salt Sandstone	QDA	0.85	QDA	0.88	QDA	0.83
Mission Canyon Limestone	Random	0.80	Random	0.68	Random Forests	0.74
Lodgepole Limestone	Ensemble	0.78	QDA	0.89	QDA	0.83

6.7.1 Analysis of Poor Classifiers

Figure 6.13 summarizes the performance of different classifiers. For the purposes of the application described in this dissertation, it is more important to use AUC and F-1 score as evaluation metrics as opposed to classification accuracy. All algorithms evaluated for the classification of SSI consistently performed poorly on two formations: Ratcliffe sandstone and Mission Canyon limestone. The data from these formations were not remarkably different as compared to other formations. Ratcliffe is a thin formation; however, Mission Canyon limestone is the formation with the most number of data points; hence the performance of the classifiers cannot be attributed to lack of data.

On further inspection of the logistic regression model, the deficiencies of the model are apparent. The input features used to create these models had low p-values: they are not statistically significant input features for the model (Table 6.3). Hypothesis testing is used to determine whether an input feature is relevant. A p-value – as determined in hypothesis testing – is the probability of obtaining as extreme a value given the null hypothesis is true. A critical p-value – a value above which the null hypothesis is rejected is taken to be 0.05.

If a p-value is above 0.05 it is not considered to be relevant to the output since the null hypothesis is not rejected (or fails to be rejected). In the case of Ratcliffe and Mission Canyon formations, the p-values for all input features are above critical p-value of 0.05; hence the SSI in these formations are unrelated to the input data. On the contrary, a successful model's p-values have been shown in Table 6.3 for Base Last Salt limestone. The obvious step would be to collect relevant data or modify the features so that they are relevant to SSI. The addition of nonlinear features in no way improves the model. However, when the features are modified using principal component analysis they appear to be significant. For the Ratcliffe sandstone formation, the first two (principal) components have significant p-values of 0.016 and 0.024 for a logistic regression model. Similarly, in the Mission Canyon limestone formation, the first two components have significant p-values of 0.009 and 0.03 for a logistic regression model. In this case, with the use of feature engineering, it was possible to improve the F-1 score and AUC of the classifier. While this analysis is possible for logistic regression due to its linearity, it is not possible to carry out a similar analysis for other classifiers. An equivalent analysis is to compute the proportion of variance explained by each feature for more complex classifiers.

Table 6.3: Hypothesis tests for logistic regression models for SSI classification

Input Feature	Ratcliffe Sandstone (p-value)	Mission Canyon Limestone (p-value)	Base Last Salt Limestone (p-value)
ROP	0.79	0.637	0.486
WOB	0.356	0.447	0.001
RPM	0.265	0.688	0.046
Torque	0.244	0.913	0.035
Flow rate	0.161	0.069	0.001

6.7.2 Balancing Classes

Class imbalances can lead to poor classification models. For the case analyzed in this dissertation, the class imbalance depends on the threshold used to split SSI into different classes. A class imbalance ratio: number of data points in the low SSI versus high SSI classes have been tabulated in Table 6.4. The upsampling algorithm was used to balance out the two classes. Accuracy, AUC, and F-1 score for formations with a class imbalance ratio of less than 1 drastically increased after using upsampling to balance classes. Results of the best models have been plotted in Figure 6.14.

Table 6.4: Class imbalance ratio for difference thresholds of SSI for data over all formations

	SSI					
Formation	**0.5**	**0.75**	**1**	**1.5**	**2**	**3**
Piper Limestone	0.30	0.93	2.64	17.15	46.20	117.00
Spearfish Sandstone	0.78	2.95	6.46	13.13	17.83	36.67
Tyler Sandstone	0.57	1.36	4.65	30.25	78.45	874.00
Kibbey Lime Limestone	0.11	0.22	0.44	1.43	1.80	3.15
Kibbey Lime Shale	0.07	0.20	1.33	11.86	14.65	26.69
Charles Sandstone	0.17	0.39	0.83	2.97	4.15	6.78
Charles Limestone	0.18	0.41	0.74	1.68	2.10	3.52
Ratcliffe Sandstone	0.11	0.16	0.51	1.55	1.81	3.50
Base Last Salt Limestone	0.40	1.19	2.55	3.68	4.15	7.58
Base Last Salt Sandstone	0.65	1.53	2.20	6.23	7.49	14.73
Mission Canyon Limestone	0.18	0.36	0.63	6.85	7.60	16.29
Lodgepole Limestone	0.36	1.19	3.31	12.07	20.21	64.55

Table 6.5: Tabulation of the best model by metric for each formation after correcting for class imbalance

Formation	AUC Model	AUC	F1 Score Model	F-1	Accuracy Model	Accuracy (%)
Piper Limestone	Logistic	0.83	Logistic Regression	0.83	Logistic	0.83
Spearfish Sandstone	Random Forests	0.92	Random Forests	0.92	Random Forests	0.93
Tyler Sandstone	LDA	0.91	LDA	0.91	Random Forests	0.91
Kibbey Lime Limestone	Logistic	0.91	Logistic Regression	0.91	Random Forests	0.96
Kibbey Lime Shale	Logistic	0.81	Logistic Regression	0.81	Random Forests	0.83
Charles Sandstone	Random Forests	0.82	Random Forests	0.82	QDA	0.82
Charles Limestone	Random Forests	0.87	Random Forests	0.87	Random Forests	0.89
Ratcliffe Sandstone	Random Forests	0.93	Random Forests	0.93	QDA	0.88
Base Last Salt Limestone	LDA	0.90	LDA	0.90	LDA	0.86
Base Last Salt Sandstone	Random Forests	0.92	Random Forests	0.92	Random Forests	0.88
Mission Canyon Limestone	Random Forests	0.85	Random Forests	0.85	Random Forests	0.85
Lodgepole Limestone	QDA	0.89	QDA	0.89	QDA	0.89

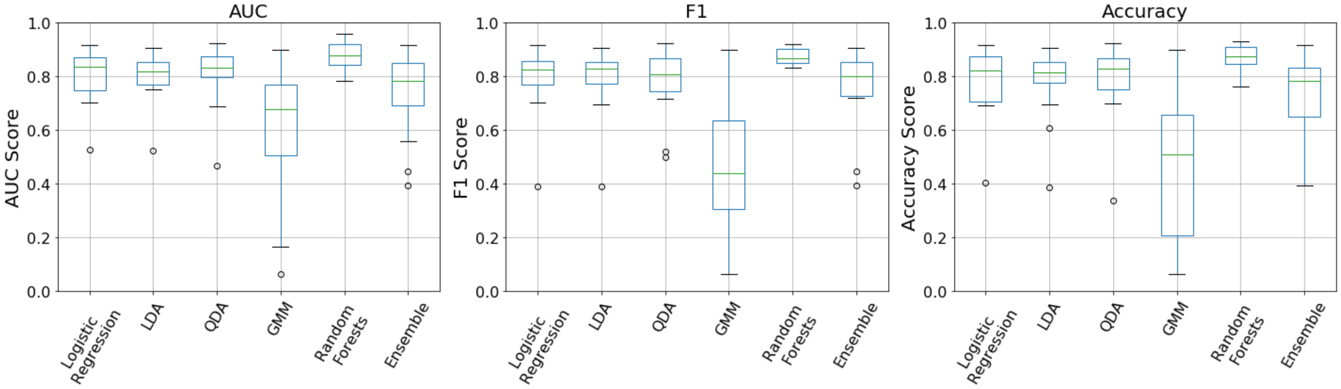

Figure 6.14: Model evaluation metrics on test dataset for SSI classification after class imbalance correction.

As seen from Figure 6.14 and Table 6.5 using upsampling results in an improvement of the F-1 score and AUC; the increase is generally higher for formations which have a class imbalance ratio less than 1. Upsampling also helps increase classifier accuracy for Ratcliffe and Mission Canyon formations which previously had very low F-1 scores. The increase in metric is related to the false positive count. Based on the analysis of confusion matrices for the balanced classes, the increase in F-1 score is related (in this case) to an increase or decrease in the precision of each classifier. When the class imbalance ratio is less than 1, the number of data points in the "low" class are rarer. Classifiers trained on these data are prone to high false positive error rates since false positives are the test data which are "low" but incorrectly classified as "high". After classes are balanced, the equality of the dataset prevents misclassifications to some extent which results in an increased precision. In particular, class balancing helps classification for this dataset since the classifiers developed were prone to false positives. Conversely, this does not occur in cases where the class imbalance ratio is high. On the other hand, small increases in some algorithms may be noticed in some cases with high-class imbalance. This is due to the increase in the number of true positives which affects the precision.

6.7.3 Bag of Models Approach

As opposed to defining a single best classification model for all formations, it is better to use the best classifier for a given formation. Using F-1 score as an evaluation metric, the model with the highest F-1 score for a given formation – based on cross-validation error – can be used to classify SSI for that formation. Adopting this method will result in a formation dependent classifier as shown in Table 6.2 and Table 6.5; this is the bag-of-models approach, where the best classifier among many is used for a specific formation based on some evaluation criteria. This method follows the no free lunch theorem in statistics.

An analysis of this approach has been shown in Figure 6.15; test error of four approaches have been plotted. The first approach is to use logistic regression for all formations, the second is to use linear discriminant analysis for all formations, the third is to use random forests for all formations, and the last is to use the bag-of-models approach.

While this approach does not improve the evaluation metric by a large amount for AUC, it makes a difference to the F-1 score for certain formations.

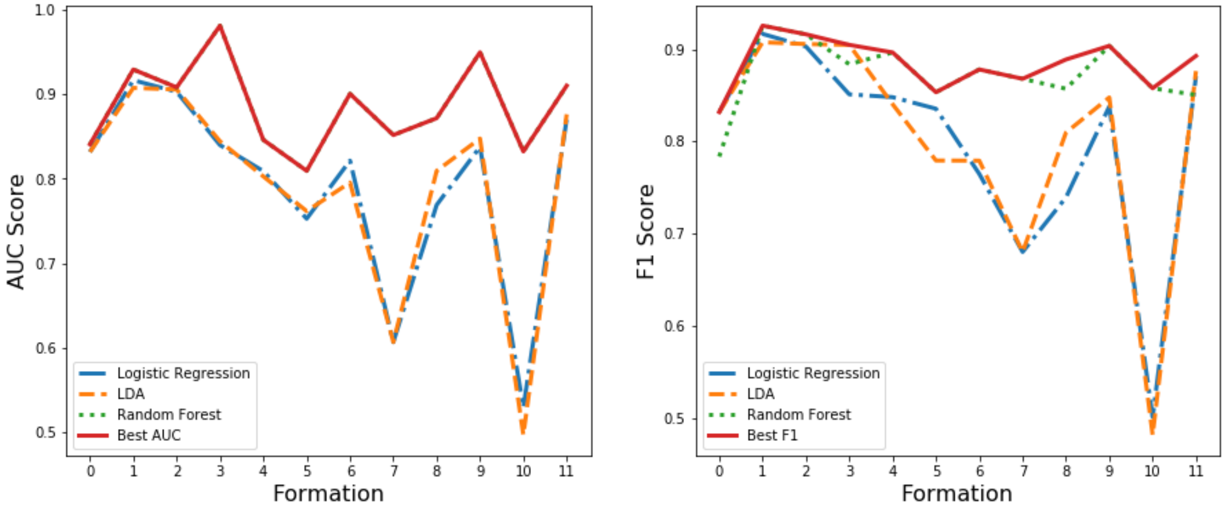

Figure 6.15: Analysis of performance of four approaches for classification of vibrations; (Left) Analysis of four classification approach on AUC score. There is almost no difference between the random forests and bag-of-models approach; (Right) Analysis of four classification approaches on F-1 score showing an improvement when using bag-of-models approach.

6.7.4 Inference

The classification models developed can also be used to create input variable importance charts. These charts are useful in gaining inferential intuition about SSI during drilling. For logistic regression, a pseudo R^2 (Equation 6.14) value can be used to assign variable importance to input features.

$$pseudo\ R^2 = 1 - \frac{\log(L_c)}{\log(L_{null})}, \text{(Equation 6.14)}$$

where, L_c denotes the (maximum) likelihood value from the current fitted model, and L_{null} denotes the corresponding value for the null model – the model with only an intercept. This method does not provide the direction of impact – if the influence towards the positive or negative class. The sign of the input feature's coefficient in the logistic regression can be used to determine the direction of influence. A positive sign indicates influence towards the positive class and vice-versa. The random forest classifier has a similar feature importance calculator which uses information gained to assign importance to features (Figure 6.16). This concept can be extended to other classification algorithms such as SVMs (Guyon & Elisseeff, 2003).

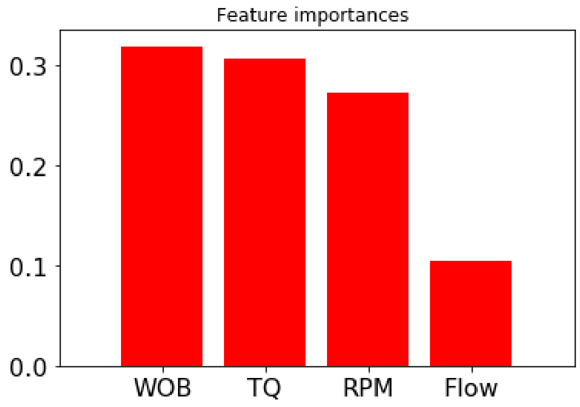

Figure 6.16: Feature importance of a random forest SSI classifier for Lodgepole limestone

6.8 Results and Discussions: Axial Vibrations

Classification of axial vibrations is performed with a threshold of 0.75 since axial vibrations were not a problem in this dataset, all accelerometer values recorded were below the threshold of 1 (Figure 6.17). However, for the sake of completeness, the threshold is lowered to 0.75 to demonstrate classification of axial vibrations. Models were trained and finetuned using cross-validation. Evaluation metrics have been calculated and plotted using a boxplot as shown in Figure 6.18. A correction for class imbalance (similar to analysis in section 6.7.2) has been applied to the classifier shown in Figure 6.18.

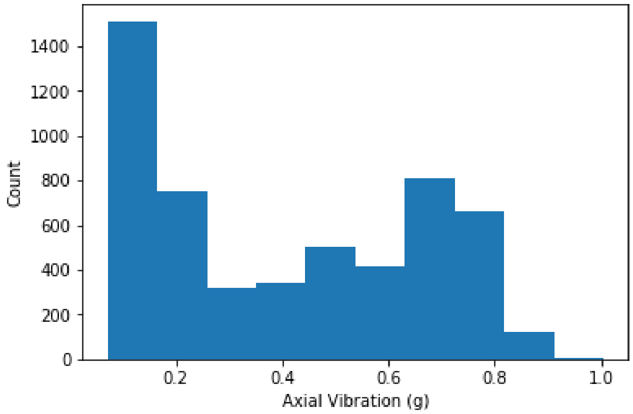

Figure 6.17: Plot of axial vibrations in the dataset; All measurements of axial vibrations were less than 1, hence to demonstrate classification a threshold of 0.75 is utilized.

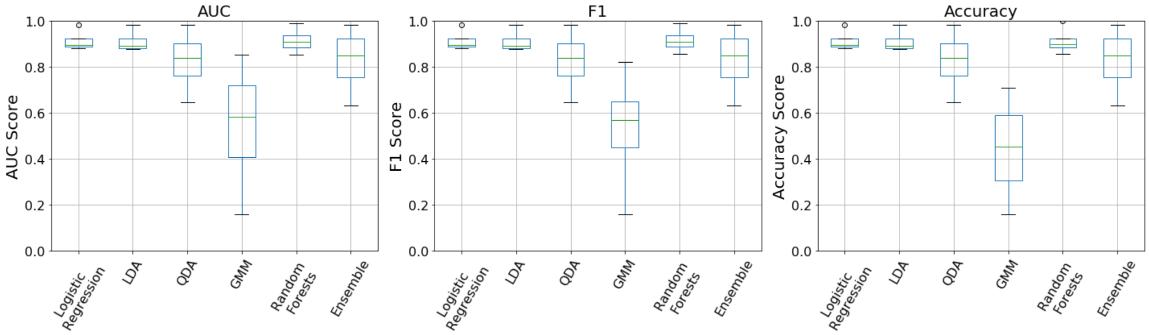

Figure 6.18: Axial vibration model evaluation metrics on test dataset after class imbalance correction.

The AUC score for different classifiers indicates that logistic regression, LDA, and random forests performed really well. This performance holds in cases of other metrics such as F-1 score and classification accuracy as well. Table 6.6 shows the best model for each formation by metric. Only formations where vibrations above the threshold (of 0.75) was observed have been modeled with a classifier. Only four formations in this dataset had axial vibrations above the threshold. Inspection of Figure 6.18 and Table 6.6 reveal that classifiers are extremely efficient at classifying axial vibrations. The bag-of-models approach can be used to select the best model for each formation. Model hyper parameters were chosen using cross validation. Only input features with a p-value lower than the critical p-value (0.05) are retained in the model. The priors for LDA are chosen based on training set ratios and dimensions are set based on number of classes (in this case 2). QDA incorporates a full covariance matrix for both classes. GMMs are built assuming the data is drawn from a gaussian of two mixtures since it is binary classification. The random forest algorithm utilized 1000 bootstrapped samples (1000 trees), a minimum leaf sample size of 4, and a random feature subset of 2. The ensemble model combined the results of logistic regression, LDA, QDA, random forests, and GMMs using weighted voting (weights determined using cross validation).

Table 6.6: Tabulation of the best axial vibration classification model by metric for each formation after correcting for class imbalance. Only formations with axial vibrations over the threshold have been classified.

	Best Model	**AUC**	**Best Model**	**F-1 Score**	**Best Model**	**CA**
0	Random	0.98	Random Forests	0.985714	Random Forests	0.99
1	Logistic	0.89	Random Forests	0.892857	Logistic Regression	0.89
2	Ensemble	0.91	Ensemble	0.906863	LDA	0.91
3	Logistic	0.88	Logistic Regression	0.881443	Logistic Regression	0.88

6.9 Results and Discussions: Lateral Vibrations

Classification of lateral vibrations is performed with a threshold of 1 similar to an analysis performed using SSI classification. Models were trained and finetuned using cross-validation. Evaluation metrics have been calculated and plotted using a boxplot as shown in Figure 6.19. A correction for class imbalance (similar to the analysis performed in section 6.7.2) has been applied to the classifier shown in Figure 6.19.

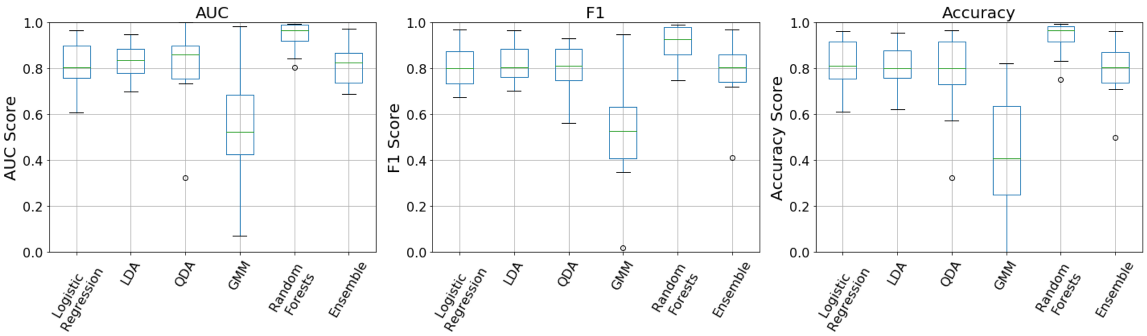

Figure 6.19: Lateral vibration model evaluation metrics on test dataset after class imbalance correction.

The AUC score for different classifiers indicates that the random forests classifier outperformed all others. This performance holds in cases of other metrics such as F-1 score and classification accuracy as well. Table 6.7 shows the best model for each formation by metric. Only formations where vibrations above the threshold (1) were observed has been modeled with a classifier. Only ten formations in this dataset had lateral vibrations above the threshold. Inspection of Figure 6.19 and Table 6.7 reveal that classifiers are extremely efficient at classifying lateral vibrations. The bag-of-models approach can be used to select the best model for each formation. Model hyper parameters were chosen using cross validation. The random forest algorithm utilized 1000 bootstrapped samples (1000 trees), a minimum leaf sample size of 5, and a random feature subset of 2. The hyper parameters used to build other models remained consistent with the axial vibration model.

Table 6.7: Tabulation of the best lateral vibration classification model by metric for each formation after correcting for class imbalance. Only formations with lateral vibrations over the threshold have been classified.

Formation	**Best Model**	**AUC**	**Best Model**	**F-1 Score**	**Best Model**	**CA**
Piper Limestone	Random Forests	0.81	LDA	0.78	Random Forests	0.82
Spearfish Sandstone	Logistic Regression	0.94	Random Forests	0.95	Logistic Regression	0.94
Tyler Sandstone	QDA	0.77	Random Forests	0.78	Random Forests	0.76
Kibbey Lime Limestone	Random Forests	0.97	Random Forests	0.90	Random Forests	0.97
Kibbey Lime Shale	Random Forests	0.99	Random Forests	0.99	Random Forests	0.99
Charles Sandstone	Random Forests	0.97	Random Forests	0.93	Random Forests	0.97
Charles Limestone	Random Forests	0.98	Random Forests	0.99	Random Forests	1.00
Base Last Salt Limestone	Random Forests	0.94	Random Forests	0.95	Random Forests	0.87
Base Last Salt Sandstone	QDA	1.00	QDA	1.00	QDA	1.00
Mission CanyonLimestone	Random Forests	0.99	Random Forests	0.99	Random Forests	0.99

6.10 CONCLUSIONS

Vibrations are the main inhibitor of ROP improvement in drilling, hence, a method to predict and control vibrations is crucial to maintaining efficient drilling operations. This paper introduced a novel method to classify a vibration-based metrics – stick-slip index (SSI), axial and lateral vibrations – using drilling operational parameters. These classification techniques use machine learning algorithms to classify the severity of vibration metrics. This model can be used in conjunction with a ROP optimization model to control drilling vibrations while improving ROP.

Different classification algorithms were evaluated for the classification of SSI. Based on the problem setup, classification accuracy is not the best metric to evaluate classifiers. AUC and F-1 scores were introduced to select the best classification algorithms. Overall the random forest algorithm outperformed other evaluated algorithms for classification of SSI when evaluated with the AUC and F-1 score. All classifiers had low F-1 scores for two formations: Ratcliffe and Mission Canyon. Classes were balanced by upsampling the minority class to help improve the performance of the classifiers. Upsampling also resulted in drastic improvements of random forests classifier for formations with a class imbalance ratio less than 1. It was concluded that a bag-of-models (one where the best classifier for that formation is used) approach always led to the best AUC and F-1 score. The random forest algorithm was used to create input feature importance ranking: adding valuable inferential capability to the model.

Classifiers for axial and lateral vibrations performed well resulting in high values of AUC score, F1-score, and classification accuracy. Classifiers were only built for formations in which vibrations exceeded the threshold. These classifiers can be easily incorporated into a ROP optimization workflow in addition to the SSI classifier.

Chapter 7: End-to-end Drilling Optimization

This chapter links all previous chapters to introduce a method to fully optimize drilling. Both machine learning and empirical models are analyzed and a workflow for end-to-end drilling optimization has been presented.

7.1 Introduction

Research in drilling optimization has always focused on modeling and optimizing ROP. Recognizing the limitation of industry standard drilling optimization tools, ExxonMobil developed a physics based drilling optimization model (Chang et al., 2014) called the drilling advisory system (DAS). The DAS is the first model of its kind that optimizes ROP while limiting drilling vibrations. Vibrations are more commonly modeled during the planning stage and are seldom considered during drilling operations. If the well does incur excessive vibrations, drilling is paused and resumed when vibrations subside. Other field-based rules such as the change of RPM and WOB to "break-out" of vibrations are commonly practiced. The key engineering contribution of the DAS is the incorporation of the effect of vibrations (D. Ertas, Bailey, Wang, Pastusek, et al., 2013; Gregory S. Payette et al., 2015) into the ROP optimization model. Much of its success in field related ROP improvements can be attributed to controlling drilling vibrations. Vibrations are modeled and controlled using a physics-based vibrations model; the model arrives at an analytical solution for torsional severity estimate using the transfer matrices method (D. Ertas et al., 2014). The DAS is a mixture of physics-based models and response surface modeling; the different elements of the DAS are combined using a modified objective function. The ROP model, MSE calculator, and vibrations model are independent and do not interact with each other – making this an uncoupled model. Since different drilling metrics interact with each other the DAS combines them in a modified objective function

(Gregory S. Payette et al., 2015). Rather than just optimize the ROP, the modified objective function seeks to maximize ROP while not increasing the MSE and controlling the intensity of drilling vibrations. This method may potentially ignore more complicated dependencies since it does not couple drilling metrics. Additionally, it is assumed that the modified objective can adequately represent downhole interaction.

This chapter links all earlier chapters to introduce two novel end-to-end drilling optimization workflows (or models). The first model – TVOPT – is similar in function to the DAS, where traditional ROP models can be used to optimize ROP while controlling for drilling vibrations and MSE. The second model – CMOPT – couples torque, ROP, and drilling vibrations using machine learning to optimize drilling – a novel technology which further builds on technologies such as ExxonMobil's DAS and TVOPT.

7.2 Drilling Optimization Workflow

7.2.1 TVOPT

The TVOPT model optimizes ROP while controlling vibrations and MSE. The ROP is modeled using traditional models or equation-based methods as discussed in Chapter 2. Drilling vibrations are modeled using classification algorithms used in Chapter 5. If drilling vibrations are instead modeled using ExxonMobil's physics-based algorithm, this model would essentially converge to the DAS. However, capturing vibrational intensity using classification algorithms has been shown to be extremely effective and has the capability to improve with additional training data (Chapter 6). A maximum MSE threshold can be set to ensure that adequate use of energy for drilling; this also controls the torque (Equation 5.1) since the MSE is directly proportional to torque. Setting a limit on surface torque is common practice in drilling operations, hence the field-based threshold on surface torque can be converted to an MSE threshold. The ROP models can be optimized using gradient ascent as shown in Chapter 4 (Section 4.4.2). A constraint is placed on the sample space used for gradient ascent a vibration classification models discussed in Chapter 6. This constraint can be used to limit vibrations in the form of axial, lateral or torsional vibrations. The process has been mapped using the flowchart shown in Figure 6.1. The ROP model presented in this workflow can be swapped out for any other ROP predictor. The method used to calculate the effect of vibrations in the DAS (D. Ertas et al., 2014) requires rigorous calculation and several manufacturer related parameters. The use of a classification algorithm for vibrations classification can be readily implemented into a WITSML stream of data without any delay. It can be implemented on any rig, by a company man, operator, or consultant. It can also be applied post-drilling to analyze deficiencies to improve future drilling wells. Additionally, since it's based on machine learning, with additional data it has the ability to improve in accuracy. Sadly, a comparison of the vibrational model used in the DAS to the classification algorithm is not possible for the dataset used in this dissertation since it is missing certain field related constants.

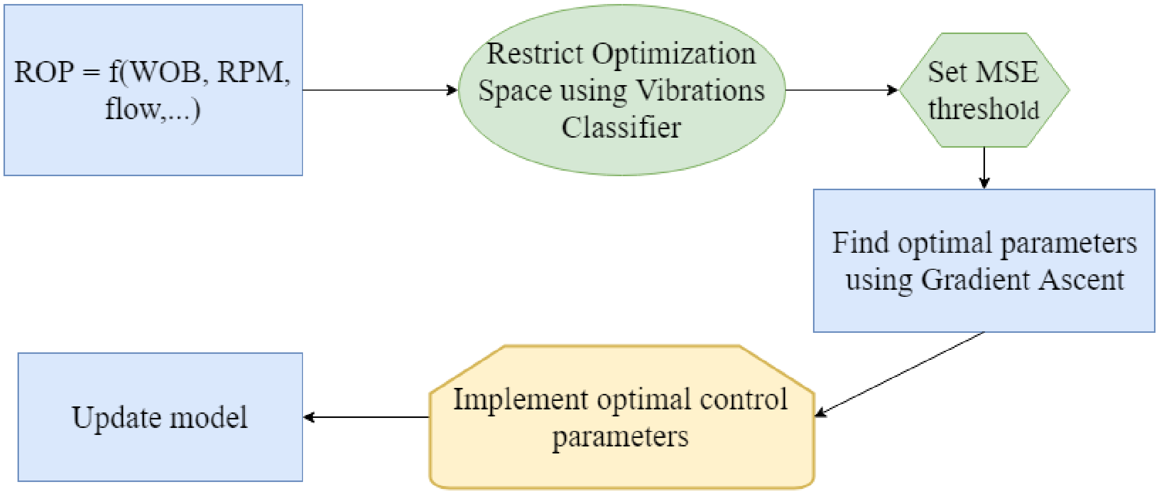
ROP = f(WOB, RPM, flow,...)
Restrict Optimization Space using Vibrations Classifier
Set MSE threshold
Find optimal parameters using Gradient Ascent
Implement optimal control parameters
Update model

Figure 7.1: Flowchart describing the TVOPT process; ROP is modeled as a function of drilling control parameters using the best traditional model; The model optimization space is restricted by using a classification algorithm to limit axial, lateral and torsional vibrations to a pre-defined threshold; Further restrict the optimization space using an MSE threshold; Determine optimal parameters using gradient ascent; The output control parameters can be implemented ahead of the bit.

7.2.2 CMOPT

The CMOPT model optimizes ROP, MSE or a user defined metric using a coupled drilling model while controlling vibrations. The ROP and TOB are jointly modeled using data-driven models which induce coupling between them (as discussed in Chapter 2). Drilling vibrations are modeled using classification algorithms (Chapter 6) using the coupled ROP and TOB models. Limits for the optimization space are set using field-related constraints and the vibrations classification model. The ability to model ROP, TOB, and drilling vibrations together using the same inputs features – thereby coupling them – is its key differentiating factor. A change in WOB can be used to simulate changes in ROP, TOB and drilling vibrations – which is not possible using ExxonMobil's DAS or TVOPT. Hence, this model does not require a complicated objective function which was necessary for DAS and TVOPT; the use of a modified objective function made it possible to incorporate the effects of vibrations and MSE on ROP optimization for the aforementioned models.

If ROP is optimized, PSO can be used to find optimal parameters implemented ahead of the bit; Bayesian optimization can be used for optimizing MSE. A constraint is placed on the optimization space using a vibration classification model. This constraint can be used to limit vibrations in the form of axial, lateral or torsion. The workflow of the CMOPT model has been visualized using a flowchart (Figure 7.2). The machine learning algorithms used for training each drilling metric can be replaced with other algorithms, however, to couple the model, its integral that the entire system be based on machine learning. This model can be easily implemented on any rig, by a company man, operator, or consultant using an input data stream (cloud-based or otherwise). It can also be applied post-drilling to analyze deficiencies to improve future drilling wells. Additionally, since it's based on machine learning, the model accuracy can improve as more data are collected.

The model can be updated at a prespecified frequency. Assuming this frequency was set at one stand; upon drilling 90 ft, all models are retrained. The newly retrained models are optimized, and a new set of drilling control parameters are calculated. This new set of drilling control parameters can be implemented ahead of the bit. Since all models are data-driven, continuous retraining allows these models to update and improve accuracy over time since model accuracy increases as more data are collected (Chiranth Hegde, Daigle, Millwater, & Gray, 2017). The update period can also be defined based on a metric or drilling model. For example, models can be updated if the measured vibrations are too high or if MSE increases abruptly.

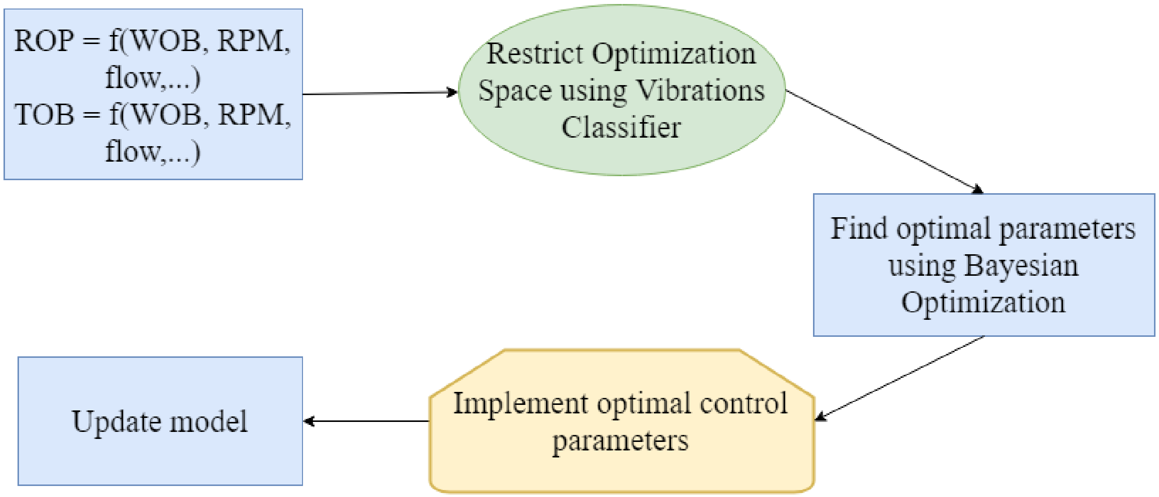
ROP = f(WOB, RPM, flow,...)
TOB = f(WOB, RPM, flow,...)
Restrict Optimization Space using Vibrations Classifier
Find optimal parameters using Bayesian Optimization
Implement optimal control parameters
Update model

Figure 7.2: Flowchart describing the CMTOP modeling process; ROP and TOB are modeled together as a function of drilling control parameters using the best data-driven algorithm; The model optimization space is restricted by using a classification algorithm to limit axial, lateral and torsional vibrations to a pre-defined threshold (can alternatively be used as a constraint in the implementation of the optimization algorithm); Determine optimal parameters using PSO or Bayesian optimization; The output control parameters can be implemented ahead of the bit; The model is retrained periodically.

7.3 Results and Discussions: TVOPT

The end-to-end drilling optimization model was implemented according to the template shown in Figure 7.1. Each formation was split into two parts – a training and test set using a 50/50 split. The initial 50% of the data collected in the formation is used as training data to train drilling models for ROP and drilling vibrations (SSI only). A threshold for MSE can be selected either using field-based intuition. The Hareland ROP model was utilized for ROP prediction since this model yielded the best accuracy for the dataset being analyzed (Chapter 2). The drilling vibrations are modeled using the classification model; regions of the sample space can be marked based on their vibration classification: low or high. The optimization model is constrained to the allowable (or green) regions of the sample space (Figure 7.5). This will ensure that the optimal control parameters as returned by the optimization algorithm will not induce vibrations (which can offset the positive effect of ROP improvement). The gradient ascent algorithm was used to find optimal parameters in the constrained region. An alternative approach to the optimization problem is to set the classification model as an equality constraint. The qualifying optimal control parameters are implemented ahead of the bit – on the test set (remaining 50% of data). The modified response curves can be compared to evaluate the change in ROP. This process has been carried out for all formations; results are summarized in the form of a box plot (Figure 7.3). Results show that an increase of 10ft/hr on average has been observed while drilling the well with the use of this model. While a deterministic model does not provide confidence intervals for a given formation, the 95% confidence interval around the mean for all data is shown in Figure 7.3. It is calculated using the data points observed for ROP prediction for all formations around the mean using a bootstrapping approach described in Chapter 5. A line plot has been used (Figure 7.4) to gain additional insight into the ROP change in each formation. Formation 3 (Tyler sandstone) shows a decrease in ROP despite attempting to optimize ROP. This can be explained by the ROP model whose predictions are an underestimation for Tyler sandstone. The optimal parameters improve the ROP with respect to the ROP model. Since the ROP model was itself an underestimation, the improved ROP is also lower than the ROP measured (Figure 7.5). Figure 7.6 shows the optimization space based on the SSI classification model (green for allowable vibrations and red otherwise) for each formation analyzed in this chapter.

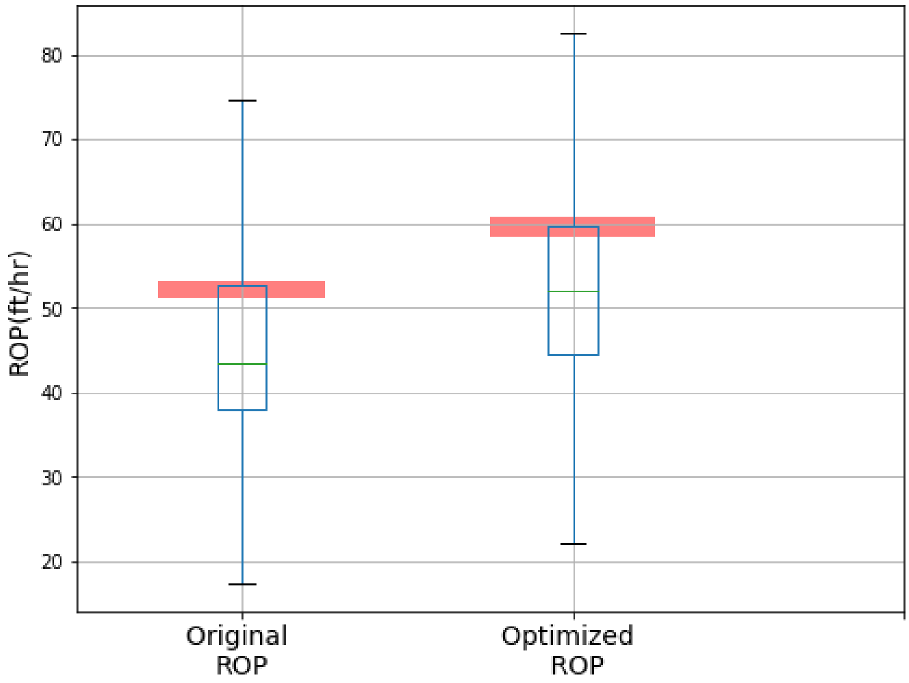

Figure 7.3: Results of ROP optimization using the TVOPT model. Regions marked in red on the plot cover the 95% confidence interval for the mean of the corresponding boxplot.

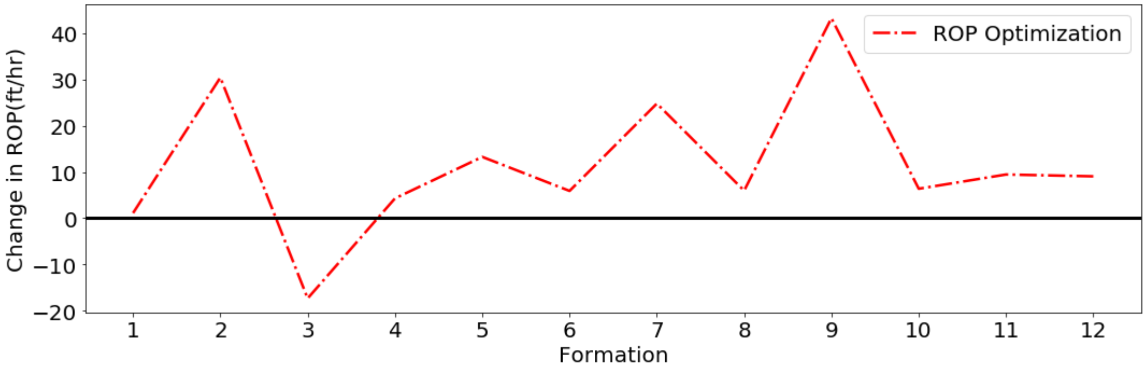

Figure 7.4: Results of ROP optimization shown as a line plot for the TVOPT model

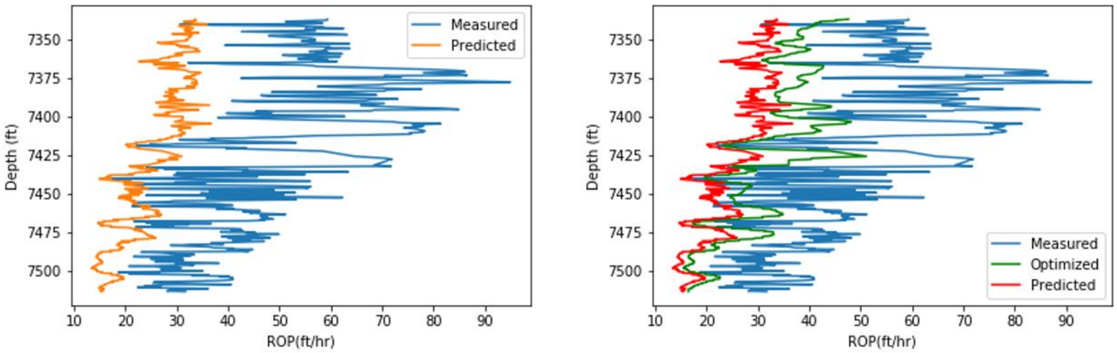

Figure 7.5: (left) ROP prediction in Tyler Sandstone – an underestimation; (Right) Optimized, predicted and measured ROP in the test dataset for Tyler Sandstone; The optimized ROP is higher than the predicted ROP, however, since the predicted ROP is an underestimation of the measured ROP, the optimized ROP is lower than the measured ROP.

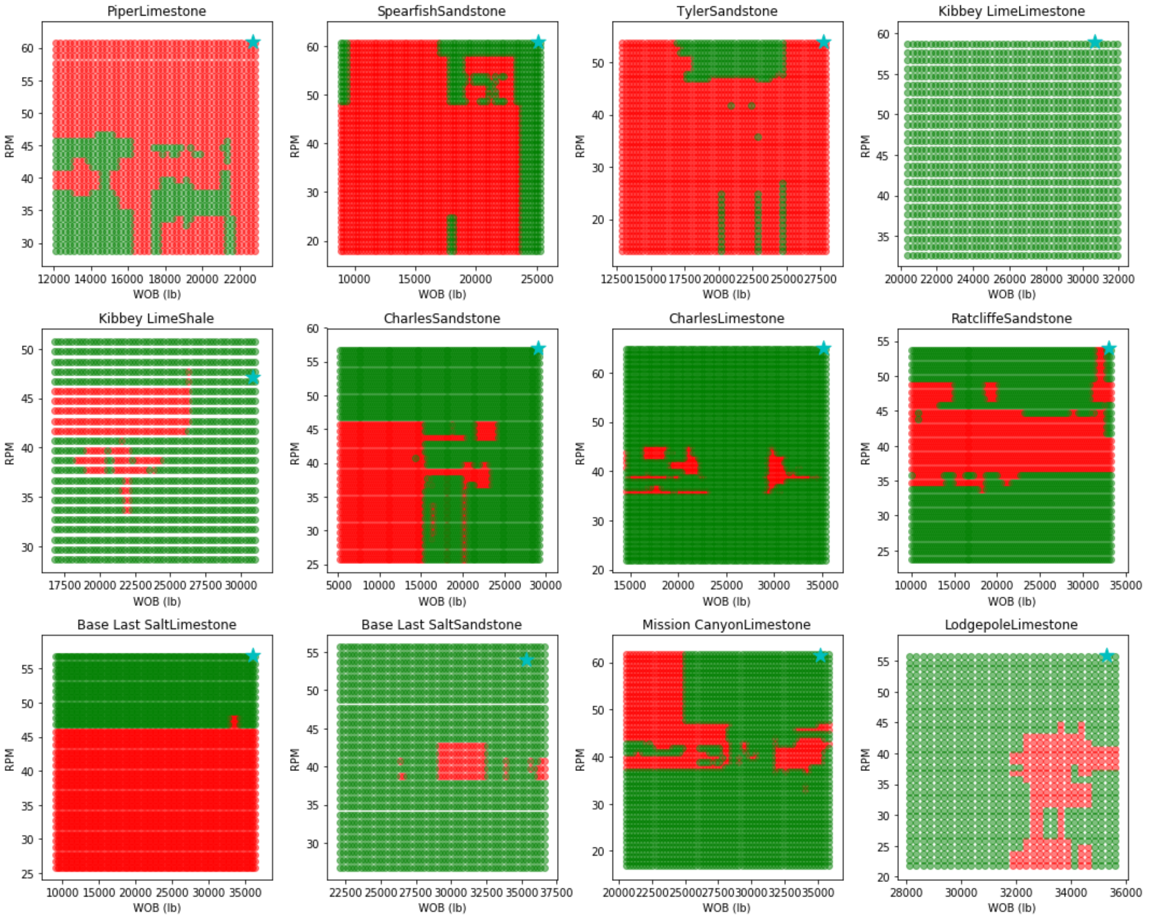

Figure 7.6: Optimization space colored using the SSI as predicted by the SSI classification model. Regions colored in red indicate high SSI (>1) and green otherwise. The optimal parameters of traditional ROP models are typically found in the upper right corner of a scatter plot, given the nature of the empirical ROP equations. In certain formations, these regions can induce high torsional vibrations during drilling; Modeling the SSI beforehand can prevent this situation as illustrated. The cyan star represents the optimal parameters as determined without the use of an SSI model

7.4 TVOPT Case Study: Charles Limestone

A case study is illustrated to show the application of the proposed drilling optimization model. It is assumed that the formation is drilled one stand at a time. The 90 ft of data was used as the training set. The rest of the formation was used as the test set for model evaluation. ROP models are built using the Motahhari ROP model using WOB, RPM, and UCS as input parameters. The cross-validation error for this model is less than 15%; hence, the models are declared to be efficient. The test set accuracy and ROP improvement due to optimization have been plotted in Figure 7.7. The SSI classification model was built using the random forests classifier algorithm since it resulted in the lowest validation set error; the test set accuracy was 84% (with an F-1 score of 0.79 and area under curve (AUC) score of 0.879). Models were optimized using the gradient ascent algorithm; the optimization space has been shown in Figure 7.8.

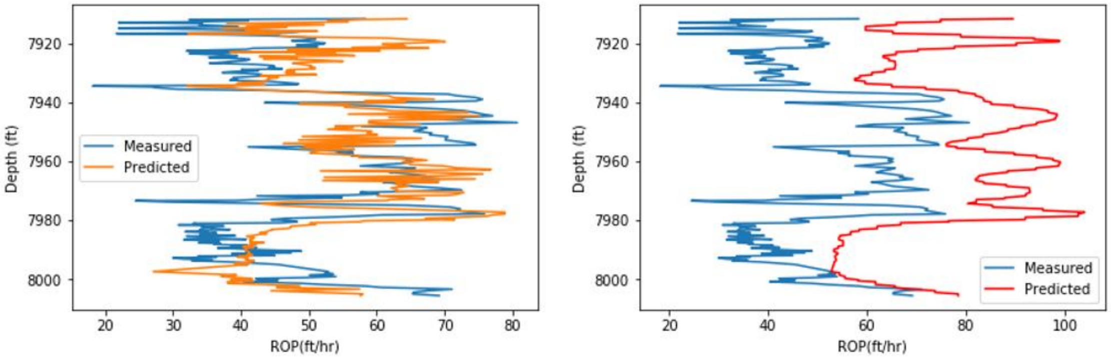

Figure 7.7: (Left) ROP prediction in the test set using the Motahhari ROP model for this formation; (Right) ROP improvement using optimal control parameters in the test set;

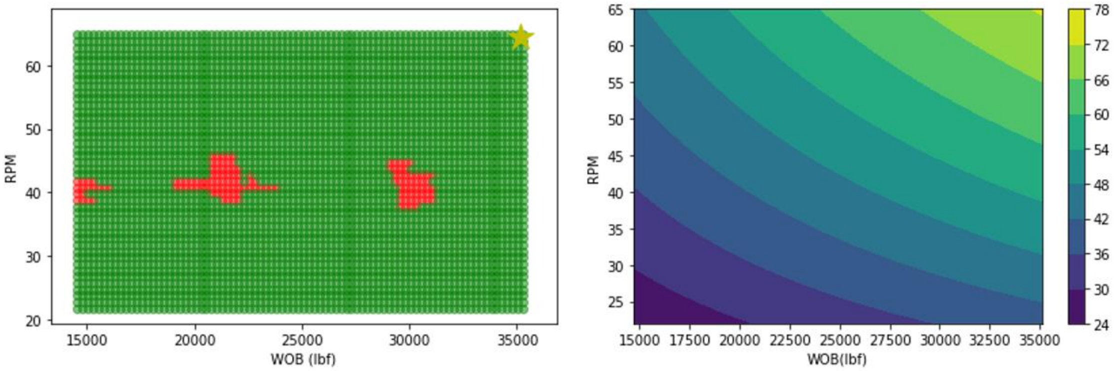

Figure 7.8: (Left) Optimization space as modeled by SSI classifier; (Right) ROP contours as modeled by Motahhari's ROP model.

7.5 Results and Discussions: CMOPT

The end-to-end drilling optimization model was implemented according to the template shown in Figure 7.2. Each formation was split into two parts – a training and test set using a 50/50 split. The initial 50% of the data collected in the formation is used as training data to train drilling models for ROP, TOB, SSI, and MSE. A metric is chosen for optimization – ROP or MSE. These models are used to optimize the objective function (ROP or MSE) using the PSO algorithm. The algorithm is implemented with inequality constraints using the SSI classification model; the output of the algorithm – the optimal control parameters – must lie in the acceptable region defined by the SSI model. These optimal control parameters are implemented ahead of the bit – on the test set (remaining 50% of data). The modified response curves can be compared to evaluate the change in the metric (ROP or MSE). This process has been carried out by using ROP and MSE as objective functions. Figure 7.9 summarizes the results for all formations analyzed using a boxplot.

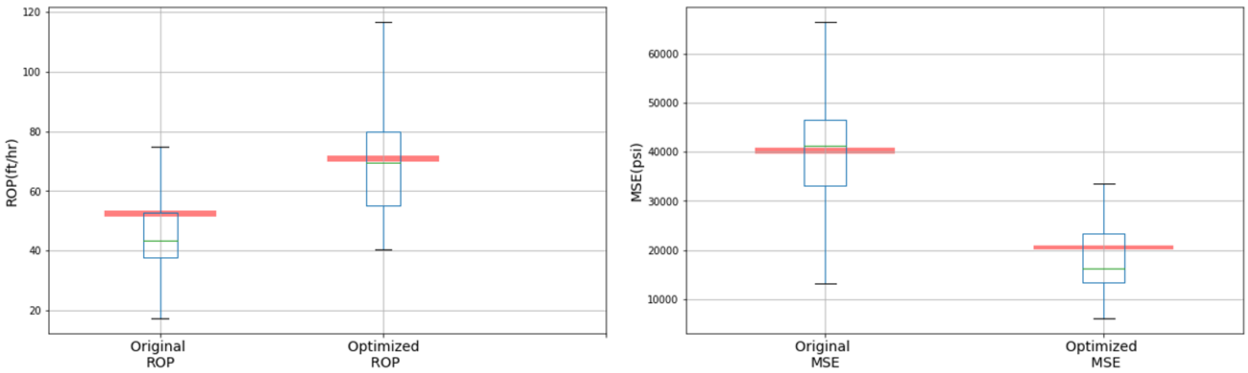

Figure 7.9: Results of drilling optimization using the end-to-end optimization model using ROP and MSE as metrics. Regions marked in red on the plot cover the 95% confidence interval for the mean of the corresponding boxplot. (Left)Average ROP increase of 19 ft/hr using the optimization model; (Right) Average decrease in MSE of 19643 psi using the optimization model

The average of each boxplot has been highlighted in Figure 7.9 using a shaded region colored red. The shaded region covers the 95% confidence interval of the mean of the dataset. The confidence interval was obtained using bootstrapping as opposed to using analytical methods since the data are nonparametric showing deviations from the normal distribution. ROP optimization shows an average increase of ROP by 19 ft/hr – much higher increase compared to TVOPT. The confidence interval for this improvement is between 63 and 64.75 ft/hr. MSE optimization shows massive improvements in MSE – a reduction of average MSE by 19643 psi. The confidence interval for the average optimized MSE lies between 19257 and 20044 psi. The larger decrease in MSE can be attributed to a simultaneous decrease in TOB and increase in ROP of the bit. It is recommended that a minimum and maximum TOB are specified while optimizing for MSE since the units of TOB are 100 times larger than the units of ROP. During optimization, a larger reduction of TOB by is possible as compared to ROP which can result in simply decreasing WOB to improve MSE. This would yield a poor result since the objective is to improve ROP or maintain ROP while reducing the MSE. An alternative is to normalize the data with a mean of 0 and standard deviation of 1 before performing optimization; not only does this method result in faster converge, but also tends to yield a better solution.

A line plot has been used to show the improvement of ROP and MSE for each formation in Figure 7.10. The line plot consists of a 95% confidence interval which shows the ROP or MSE improvement for each formation. A formation does not show improvements – beyond statistical doubt – in MSE or ROP if the if the confidence interval intersects with the X-axis. Figure 7.10 shows no changes or improvements in ROP for all but one formation – Base Salt Limestone. The improvement in ROP has been measured with respect to the actual data measured in the field. For this formation, the ROP model underestimated the actual ROP. The optimization algorithm is used on the ROP model and results in an improvement when compared to the predicted ROP but not the measured ROP (Figure 7.11).

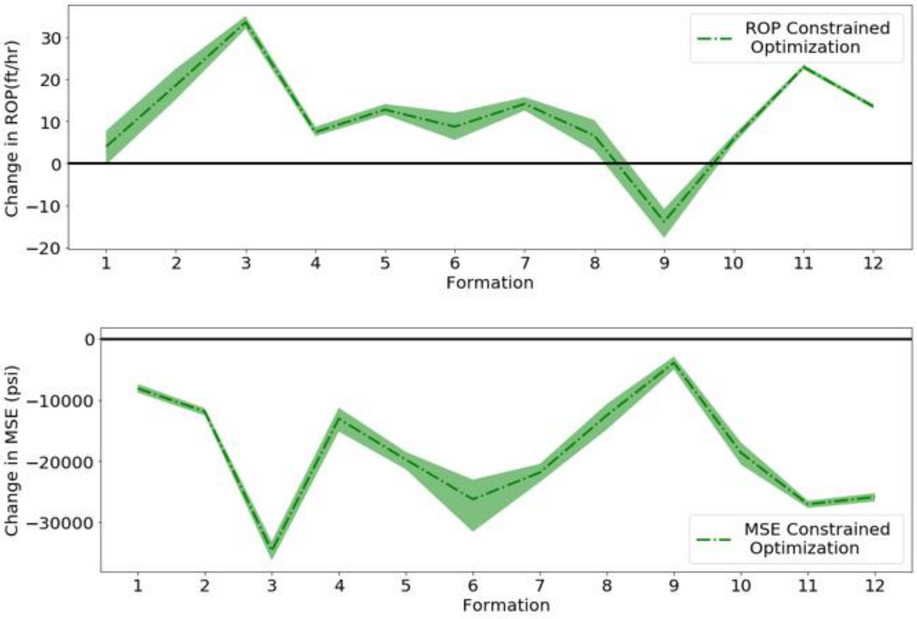

Figure 7.10: Results of drilling optimization by formation using the end-to-end optimization model using ROP and MSE as metrics. Regions marked shades around the thick line cover the 95% confidence interval for the average improvement for each formation. A formation's ROP or MSE does not improve if this confidence interval intersects with the X-axis of the corresponding boxplot. (Top) Average ROP increase; (Bottom) Average decrease in MSE

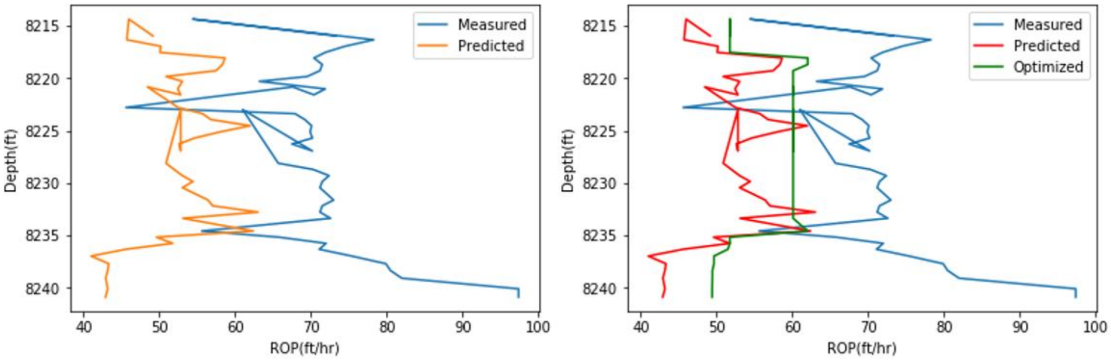

Figure 7.11: (Left) ROP Model prediction for Base Last Limestone; (Right) ROP improvement for Base Last Limestone; The prediction – which underestimates the data measured in the test set – is improved, however, the improved ROP is also an underestimate of the measured ROP

7.5.1 Coupled Effect

It is possible to investigate the change of other important drilling metrics due to optimization of one metric (as discussed in Chapter 5). This section investigates the change in MSE due to ROP optimization and the change in ROP due to MSE optimization.

If ROP is optimized, it would be ideal if MSE remains constant or decreases. Increases in MSE due to ROP optimization generally occur due to excessive vibrations (or formation change). Using the coupled model (CMOPT) it is possible to note the changes in MSE when optimal parameters to optimize ROP are implemented. The parameters for ROP optimization are calculated with the aim of purely optimizing ROP. Figure 7.12 shows the effect of ROP optimization on ROP and MSE; shaded regions around the line plots indicate 95% confidence intervals. If the confidence intervals intersection with the X-axis, then no statistical change is observed. An average improvement of 30 % is observed for ROP and decrease of MSE by 30%. Charles sandstone does not see an improvement in MSE and an MSE increase is observed in the case of Base Last Salt limestone.

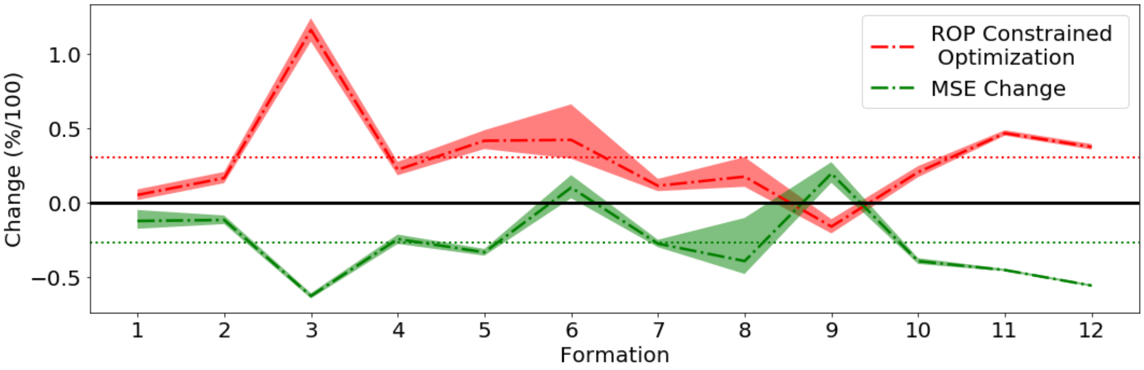

Figure 7.12: Effect of ROP optimization on drilling metrics. An increase of ROP by 30% and a decrease in MSE by 30% is observed. The shaded portions of the plot indicate 95% confidence intervals. The average change across all formations has been plotted as a dotted line by color.

MSE optimization is a bit more complicated since ROP is part of the objective function being optimized. The parameters for MSE optimization are calculated with the aim of purely optimizing MSE – which is possible by increasing ROP or reducing TOB. Hence there can be a situation – which is highly undesirable – where MSE is minimized by decreasing TOB using a reduction in WOB, and correspondingly reducing ROP. If the change in WOB results is a large decrease in TOB as compared to ROP, then this will reduce MSE, making it a valid set of optimal parameters as per its definition. It is possible to evade this solution by enforcing a lower limit on TOB. Figure 7.13 shows the effect of MSE optimization on ROP and MSE; shaded regions around the line plots indicate 95% confidence intervals. An average improvement of 21.2 % is observed for ROP and MSE decrease of 49.1 % has been observed in the case of MSE optimization. No ROP improvement is observed in Charles (sandstone and limestone) formations and a decrease in Base Last Salt limestone.

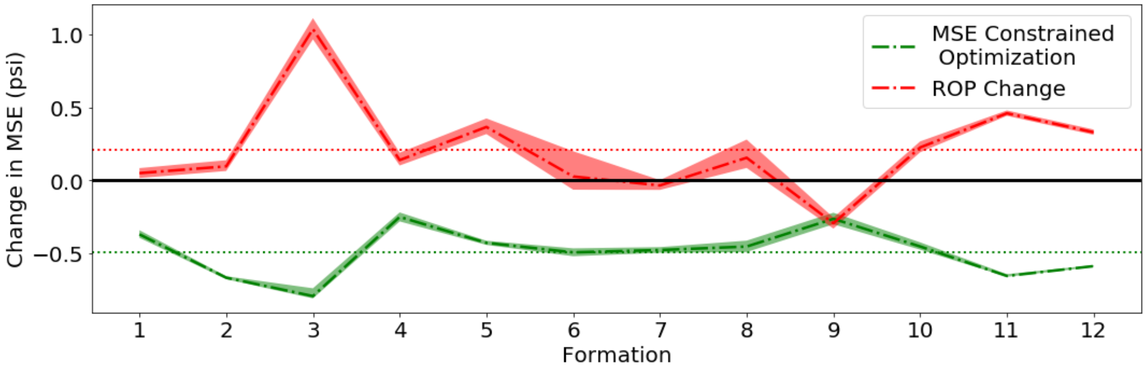

Figure 7.13: Effect of MSE optimization on drilling metrics. An increase of ROP by 21% and a decrease in MSE by 49% is observed. The shaded portions of the plot indicate 95% confidence intervals. The average change across all formations has been plotted as a dotted line by color.

7.6 CMOPT Case Study: Charles Limestone

A case study is illustrated to show the application of the proposed drilling optimization model. It is assumed that the formation is drilled one stand at a time. The 90 ft of data was used as the training set. The rest of the formation was used as the test set to evaluate the model. ROP and TOB models were built using the random forest algorithm using WOB, RPM, flow-rate, and UCS as input parameters. The cross-validation error for both models were less than 10%; hence, the models are declared to be efficient. The test set accuracy of both models has been shown in Figure 7.14. The SSI classification model was built using the random forests classifier algorithm since it resulted in the lowest validation set error; the test set accuracy was 87% (with an F-1 score of 0.78 and area under curve (AUC) score of 0.87).

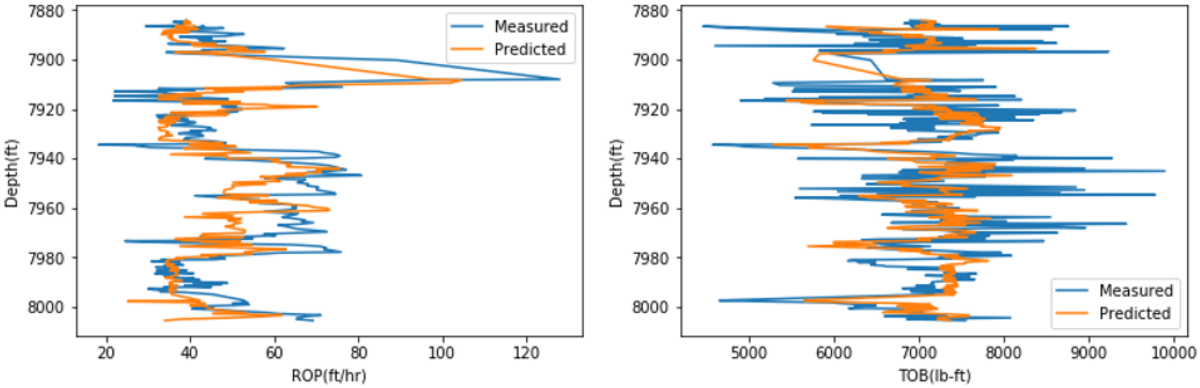

Figure 7.14: Test set predictions for ROP and TOB model built using the random forests algorithm on Charles Limestone. The model was trained on data collected while drilling the initial 90 ft of the formation.

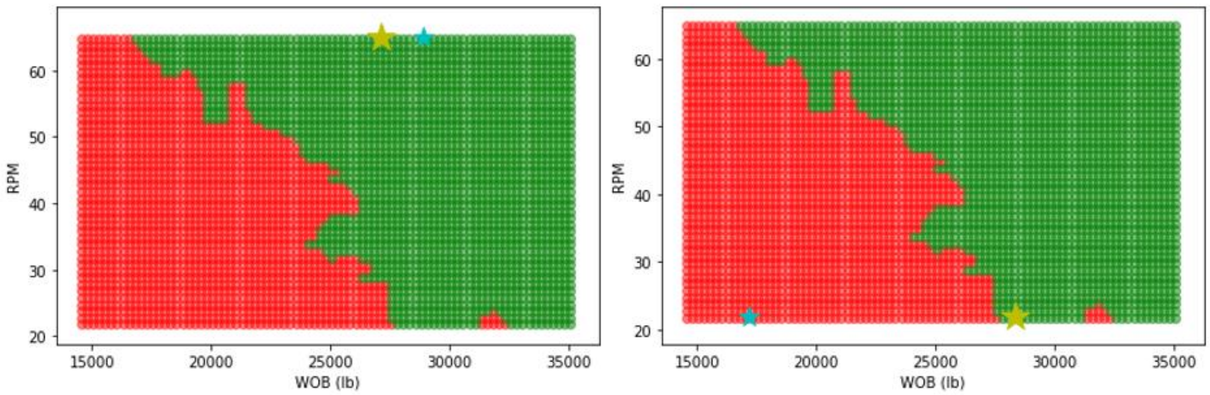

Figure 7.15: 2D representation of Optimization space for Charles Limestone. The area shaded in green includes drilling control parameters which will not result in excessive SSI – the constrained optimization space. Optimal parameters obtained upon constraining the optimization space – with the use of the SSI model – are plotted in yellow. The unconstrained optimization – which can result in excessive SSI – is plotted with a blue star. (Left) ROP optimization (Right) MSE optimization.

Drilling optimization results of two KPIs or metrics have been evaluated – ROP and MSE. The models were optimized using the PSO algorithm with constraints. The optimization space is obtained by taking the end ranges of WOB, RPM, and flow-rate in the training dataset. For example, since RPM varied between 20 and 60 in the training set, the optimization space is bound at these limits for RPM. The SSI model is used to evaluate any additional constraints which can occur due to excessive torsional vibrations. The entire optimization space is classified using the SSI model into low and high vibration zones. The optimization algorithm is constrained to the subspace which does not result in excessive torsional drilling vibrations – the area marked in green in Figure 7.15. The Bayesian optimization algorithm is used to find the optimal control parameters to optimize ROP or MSE ahead of the bit. Figure 7.13 shows a two-dimensional representation of the optimization space along with the optimal control parameters. The optimal parameters returned by the algorithm in the case of ROP and MSE optimization lie in the green region

(Figure 7.13). The optimal parameters for ROP and MSE optimization are different since the objective functions optimized are different. The optimized KPIs: increase in ROP and decrease in MSE are shown in Figure 7.14. In this case, no further model update was required for improving results. However, while drilling a longer formation, the ROP, TOB and SSI model can be continuously recalibrated by training them after drilling each stand.

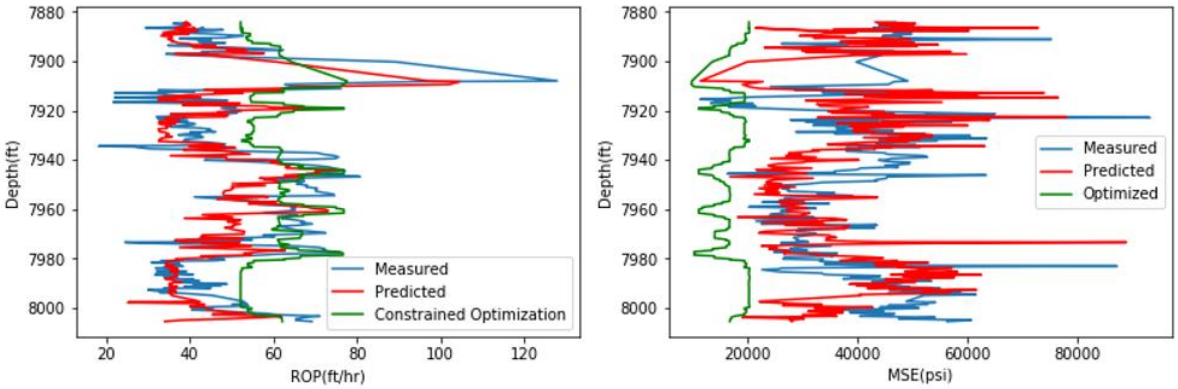

Figure 7.16: Optimized ROP (left) and MSE (right) using PSO algorithm for optimization with vibration-based constraints set using the SSI model

The case presented here did not incur lateral vibrations during drilling. The drilled well analyzed in this dissertation did not show high lateral vibrations in general. One formation –Mission Canyon limestone – exhibited signs of lateral vibrations (when the threshold is lowered to 0.75). The optimization windows for ROP and MSE optimization for this formation are shown in Figures 7.17 and 7.18. In this case, the CMOPT model would have two equality constraints – one for SSI and the second for lateral vibration – constraining the optimization space to account for both sources of vibrations. An additional constraint can be similarly added to the optimization algorithm in the CMOPT model to account for axial vibrations if required.

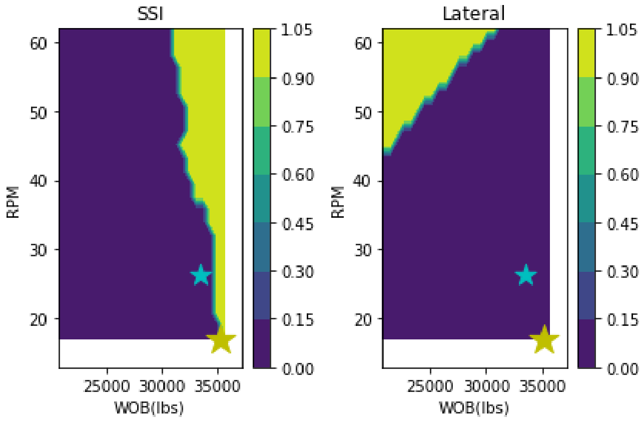

Figure 7.17: 2D representation of Optimization space for Mission Canyon Limestone. The area shaded in blue includes drilling control parameters which will not result in excessive SSI or lateral vibrations – the constrained optimization space. Optimal parameters obtained upon constraining the optimization space – with the use of the SSI model – are plotted in yellow. The unconstrained optimization – which can result in excessive SSI – is plotted with a blue star. (Left) SSI optimization window (Right) Lateral vibration window

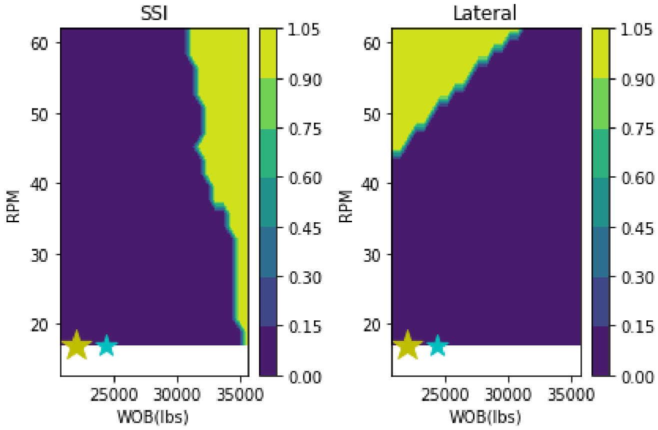

Figure 7.7: 2D representation of Optimization space for Mission Canyon Limestone for MSE optimization. The area shaded in blue includes drilling control parameters which will not result in excessive SSI or lateral vibrations – the constrained optimization space. Optimal parameters obtained upon constraining the optimization space – with the use of the SSI model – are plotted in yellow. The unconstrained optimization – which can result in excessive SSI – is plotted with a blue star. (Left) SSI optimization window (Right) Lateral vibration window

7.7 Conclusions

This chapter introduced two drilling optimization models the TVOPT and CMOPT. The former improved upon a method of modeling which has existed in the industry for over half a century. Drilling vibrations have affected ROP improvements but have never been adequately addressed for drilling optimization. The TVOPT model predicts ROP using traditional models. The modeled ROP along with drilling control parameters are used in a machine learning classification model to classify the optimization space. An optimization algorithm (gradient ascent) is used to find optimal parameters – which can be used to improve ROP ahead of the bit. However, values of the optimal parameters are restricted by the drilling vibrations classification model – which ensures that the optimal parameters do not induce excessive vibrations upon implementation. The use of this model resulted in a ROP improvement of 10 ft/hr (14.8%) on average across all formations. TVOPT was successful in improving ROP for all formations except one (Figure 7.4) – Tyler Sandstone. A closer investigation into the ROP improvement for each formation showed that the underestimation of the ROP model caused this apparent decrease in ROP performance. Overall, the TVOPT model provides a method to find optimal parameters for ROP improvement; these optimal parameters are safe – will not induce excessive drilling vibrations. A similar workflow can be used to keep axial and lateral vibrations in check.

The CMOPT model introduced a modeling method to optimize any drilling metric (or objective) – ROP, MSE or a combination – using a coupled modeling scheme. Individual models for ROP, TOB, and SSI are built using the random forest algorithm. Training data are used to build models – ROP, TOB, MSE and SSI – for a given formation. An optimization algorithm – particle swarm method or Bayesian optimization– is used to determine optimal control parameters which will maximize or minimize the metric concerned. The optimization space is constrained using the SSI model to ensure that the optimal parameters do lead to high SSI while drilling thus avoiding drilling dysfunction. Optimization of ROP and MSE has been evaluated in this paper. ROP was improved by 36% (19ft/hr) and MSE was reduced by 49%(19363 psi). The optimal parameters in both cases were constrained to low vibration regions of the optimization space. The coupled model was used to study the effect of ROP/MSE optimization on other drilling metrics. Optimizing for ROP led to an increase in ROP (36%) and decrease in MSE (30%), whereas optimizing for MSE resulted in a smaller increase in ROP (21%) and larger decrease in MSE (49%). A case study on the Charles limestone formation was used to illustrate the entire optimization process. Add more lateral and axial vibrations models into the workflow was shown with a case study in the Mission Canyon limestone formation. In conclusion, it has been shown that end-to-end coupled drilling optimization using machine learning model was successful in optimizing drilling.

Chapter 8: Conclusions

This chapter concludes the dissertation. The original research work has been discussed in Chapters 2 through 7. The objective of this chapter is to review the contributions of this dissertation and propose avenues for future work.

8.1 Conclusions

Drilling optimization always will be an area of interest in the oil and gas industry. The current techniques used for drilling optimization date back many decades. Today, the increase in data collection, computational tools, and incentive for optimization has introduced many tools which can be utilized to improve the methodology and workflow involved. Machine learning (or data science or statistical learning) can be invaluable tools utilized in drilling, considering the volume of data collected on the rig – surface and downhole.

This dissertation started by reviewing current techniques applied in the field: ROP modeling and optimization. Since ROP remains to be the most commonly used field metric, many ROP models have been proposed. Chapter 2 introduced a robust method to predict ROP while drilling wells: using the random forest algorithm (in the form of a data-driven model). This technique outperformed empirical models as well as other machine learning algorithms. This algorithm is particularly powerful for low-dimensional data – as recorded during drilling. It is important to note that algorithms which are better suited to the problem at hand are preferred to popular algorithms such as neural networks which are designed for high dimensional data problem (such as images, text, and videos). Torque-on-bit (TOB) an important metric was successfully modeled using a similar methodology. ROP and TOB are modeled as functions of drilling control parameters – WOB, RPM, Flowrate, and UCS. Modeling these metrics together using a machine learning algorithm helps couple the models. The random forest model resulted in an average normalized error of 12% and R^2 of 0.84 for all formations analyzed. Similarly, TOB predictions using the random forest model resulted in an average error of 16%.

The accuracy of data-driven models is contrasted with the loss of interpretability of the model. The ROP model – predicted by random forests – cannot be visualized in the form of an equation (a new concept for many). Empirical models on the other hand have high interpretability, but also low accuracy. Data-driven models have high accuracy but low interpretability. A trade-off exists where an increase in accuracy (compared to empirical models) and an increase in interpretability (compared to data-driven models) is possible. This class of models are called hybrid models, obtained by combining empirical and data-driven modeling techniques. Empirical coefficients in deterministic models are calculated by conditioning the models to the data – the training set. Data-driven models, on the other hand, rely purely on the data. Hybrid models combine the two analogies. Two algorithms for building hybrid ROP models have been discussed: hybrid-One and hybrid-N. The hybrid-One model combines different versions (or realizations) of a single deterministic model – providing an alternative method to determine empirical constants. The hybrid-N model combines predictions from different deterministic models using an ensembling algorithm. The application of this model is used in cases where an inference of the model may be required. A case study was used to illustrate the application of such a tool while drilling.

ROP models are created with the intention of using them to optimize ROP. While many ROP models exist in literature, not many papers pen down the how the models can be optimized to provide tangible results. Chapter 4 discussed strategies to optimize ROP models: empirical and data-driven. Empirical ROP models are easy to optimize since they modeled as power-law functions. Gradient ascent was used to find optimal parameters for each ROP model; these optimal parameters were used to estimate the improvement in ROP on each formation. The optimization of traditional models showed an average improved ROP of over 28% for all models (much higher in a few cases). However, the ROP predictions themselves were not accurate (model accuracy was low when these models were employed for ROP prediction), hence the projected ROP improvements using optimal values cannot be fully trusted. In the case of data-driven models, the absence of an equation makes it difficult to find a suitable algorithm for its optimization. This problem persists in all cases where the model being optimized is data-driven – hence the solution proposed is a general solution which can be applied to any data-driven model. This inverse problem is commonly solved using meta-heuristic algorithms. Two simple algorithms and three meta-heuristic algorithms were evaluated in this dissertation based on their ability to improve ROP ahead of the bit (higher the better) and computational run-time. Simple algorithms, like the eyeball method and random search, performed well, improving the ROP by 20% and 21% on average over all the formations tested. Their run-time was trivial (<100 ms) making them effective algorithms for real-time use with zero data-lag. Advanced meta-heuristic algorithms: simplex method, differential evolution and particle swarm method improved the ROP by 30%, 40% and 45% (on average). The computational run-times (mean) were 5.93 s for simplex, 18.475 s for DE and 196.41 s for PSO. These run-times can have adverse effects on the amount of data used to train the model in real-time (or data-lag).

So far, the metric in the dissertation used for drilling optimization was ROP. However, it is known that ROP is not always the best parameter to optimize. Other drilling metrics such as MSE, TOB, and cost-per-foot were evaluated for drilling optimization in Chapter 5. Based on the objective function being optimized, the results can vary. Cost-per-foot is not a good objective function for algorithmic optimization since it just translates to

ROP optimization. Good engineering decisions and improved management practices can make up for the aspects of cost-per-foot not covered by ROP. The best objective function to use for a well will depend on several factors such as the operator, the costs of drilling, equipment, operating costs, history of drilling dysfunctions. The coupled model was used to evaluate the changes in each metric due to optimization. Using ROP as an objective function leads to an improvement of ROP by an average of 31%, whereas MSE increases by 4% and torque on average increased by 10%. When TOB is used as an objective function, reduction in TOB resulted in a low ROP improvement and an increase in MSE which was undesirable. An average of 10% reduction in TOB was noted with a 16% increase in ROP and 15 % increase in MSE. Bayesian optimization can lead to faster convergence for optimization of MSE as compared to the particle swarm algorithm. Optimizing the MSE objective function led to an average decrease of MSE by15%, an increase of ROP by 20% and reduction of torque by 7%. By far, using MSE as an objective function has the most balanced improvement for drilling – an increase in ROP, reduction in torque and MSE at the same time.

Vibrations are the main inhibitor of ROP improvement, hence, a method to predict and control vibrations is crucial for efficient drilling operations. Techniques to classify drilling vibrations are described in Chapter 6. Different classification algorithms were evaluated for the classification of axial, lateral and torsional vibrations. Overall the random forest algorithm outperformed other evaluated algorithms for classification of stick-slip-index (SSI) when evaluated with AUC and F-1 score. Classifiers for axial and lateral vibrations performed well resulting in high values of AUC score, F1-score, and classification accuracy. Classifiers were only built for formations in which vibrations exceeded the threshold. These classifiers can be easily incorporated into a ROP optimization workflow in addition to the SSI classifier.

Chapter 7 combines all the individual models and theory described in the earlier chapters to introduce two drilling optimization models: TVOPT and CMLOPT. TVOPT improves upon a method of modeling which has existed in the industry for over half a century. The TVOPT model predicts ROP using traditional models. The modeled ROP along with drilling control parameters are used in a machine learning classification model to classify the optimization space which ensures that the optimal parameters do not induce excessive vibrations upon implementation. The use of this model resulted in a ROP improvement of 10 ft/hr (14.8%) on average across all formations. The CMOPT model introduces a method to optimize any objective – ROP, MSE or a combination – using a coupled modeling scheme fully based on machine learning. Individual models for ROP, TOB, and SSI are built using the random forest algorithm. The optimization space is constrained using the SSI model to ensure that the optimal parameters do not lead to high SSI while drilling thus avoiding drilling dysfunction. Optimization of ROP and MSE has been evaluated in this paper. ROP was improved by 36% (19ft/hr) and MSE was reduced by 50%(19363 psi).

The dissertation has shown how statistical (or machine) learning techniques can be used to adequately address the problems encountered in drilling optimization. The model developed in this dissertation builds upon the state-of-the-art model currently available.

8.2 Future Work

The work conducted as a part of this dissertation marks the beginning of a new era in drilling optimization. These models can be used within a tool on the rig to optimize all wells drilled by an oil and gas operator. The work in this dissertation covered the use of modeling for improvement of ROP or MSE in a given well. This can be extended to multiple wells, where pad wells are drilled more efficiently utilizing the data collected in the initial well. A Bayesian prior can be used to model the transition between use of data for modeling drilling in a given well.

The use of well logs collected during logging-while-drilling (LWD), along with drilling data can be used to infer formation change. When drilling parameters remain stationary and vibrations low, this task can be achieved by looking for a change in MSE. However, the development of a score similar to the drilling score present in DAS can be beneficial to detect formation change or onset of drilling dysfunction.

The ideas presented in this dissertation can be used to conduct drilling simulations. Geostatistical modeling can be used to create a map of rock strengths in a given reservoir. Once a map of UCS is obtained, drilling can be simulated using different ROP models. This can be used to help identify well paths to reduce the time and length of a required well to reach the target formation. Additionally, reinforcement learning algorithms can be incorporated into these models to better plan new wells.

The main purpose of drilling is to be able to reach the reservoir. The use of well logs collected during logging-while-drilling (LWD) can be used to identify target formations in real-time. Maintaining borehole quality can be important while drilling target zones; this can lead to improved production and shorter completion times.

Appendices

Appendix A: Confidence Intervals for Random Forests

This section explores confidence intervals of random forest ROP models: an added advantage of using a data-driven approach to model ROP (or any other drilling metric). Confidence intervals are analogous to the uncertainty of the predicted ROP. A confidence interval is an interval associated with a random variable yet to be observed or predicted (ROP in this case), with a specified probability of the random variable lying within the interval. A more mathematical definition of confidence intervals for random forests and its calculation is out of the scope here (Wager, Hastie, & Efron, 2014). Figure A-1 plots ROP predictions in Tyler sandstone using the random forests algorithm. The model is trained on data (shown as data points without error bars) and then used to predict ROP for the rest of the formation (data points with error bars). Error bars around the ROP predictions is the 95% confidence interval for the predictions. The confidence interval is special to data-driven models, i.e. it is not possible to estimate confidence intervals for traditional or deterministic models. These confidence intervals provide an additional advantage over traditional models since they estimate the uncertainty of the ROP prediction. Uncertainty in ROP prediction can be very useful from an engineering perspective to make drilling decisions.

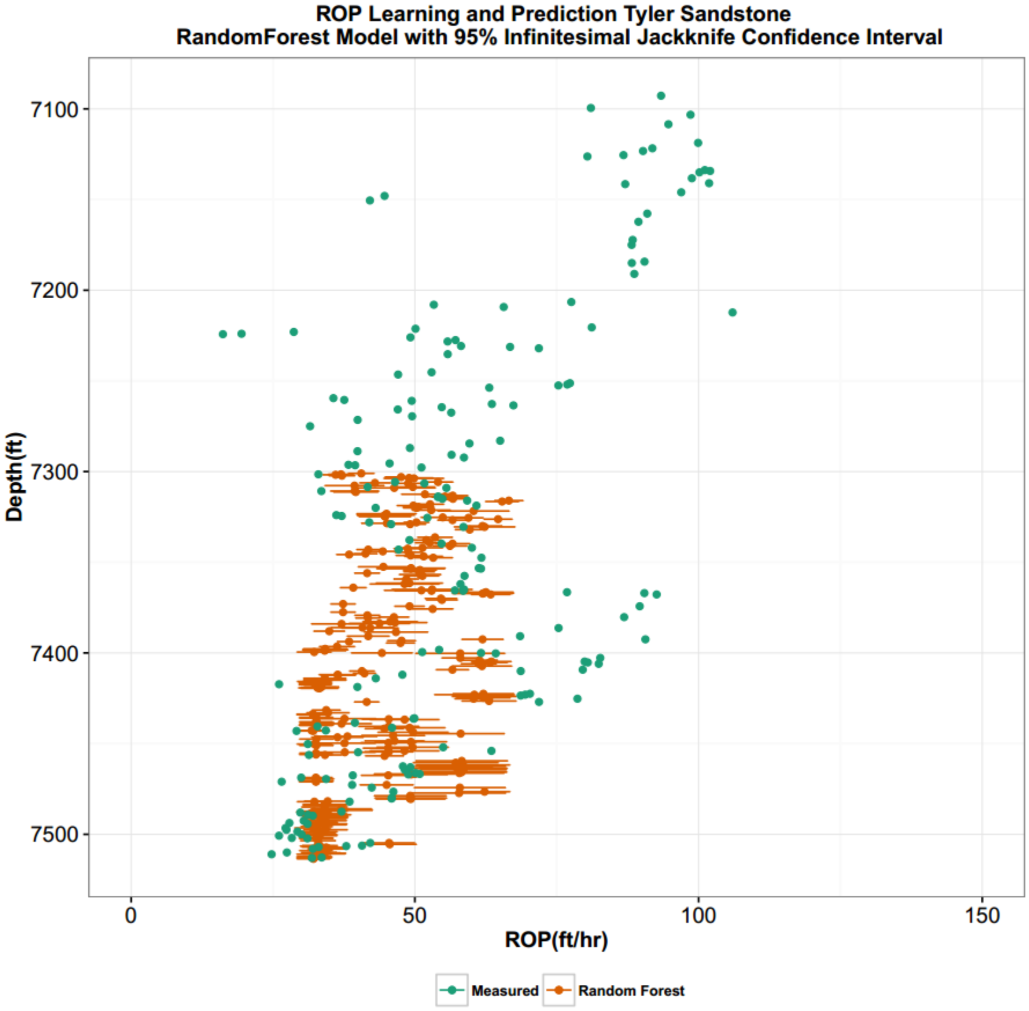

Figure A-1: Confidence interval for ROP predictions using infinitesimal jackknife approach. The first half of the formation is used to train the model. The model is used to predict ROP along with confidence intervals.

Another way to compute a confidence or prediction interval for random forests is to describe them using a quantile random forest. Quantile regression forests are an extension of the popular random forests; however, they describe conditional probabilities instead of fixed predictions. Quantile random forests can be used to quantify the complete conditional distribution of the entity at hand (ROP in our case). A complete conditional description of the variable – $P(Y \leq y \mid X = x)$ – can be extremely useful, since it can be manipulated to yield very interesting quantities: outliers, prediction intervals quantile estimation, and probability (P_{10}, P_{50}, P_{90}) estimates of ROP. Meinhausen (Meinshausen, 2006) formulated and described quantile regression forests.

Trees are built for a random variable - as described in chapter 2; however, all the data are retained, not just the average (like in a conventional random forest algorithm). Instead of computing $F(y|X=x)$, $F(y|X<x)$ is calculated. Computing $F(y|X<x)$ for the 2.5th and 97.5th percentile will yield a 95% confidence interval. The confidence interval for ROP predictions for Tyler sandstone using quantile regression forests has been plotted (Figure A-2). The model is trained on data (shown as data points without error bars) and then used to predict ROP for the rest of the formation (data points with error bars).

Since this approach can be used to represent the variance in the ROP predictions, it can be combined or propagated to determine the confidence interval of the average improvement of ROP. Figures 7.3 and 7.9 calculate the confidence interval for the average improvement in ROP considering ROP predictions as a deterministic value. Since the random forest is stochastic, additional information or model prediction uncertainty from each ROP prediction can be propagated to improve this bound. This can be calculated using Monte Carlo based simulations.

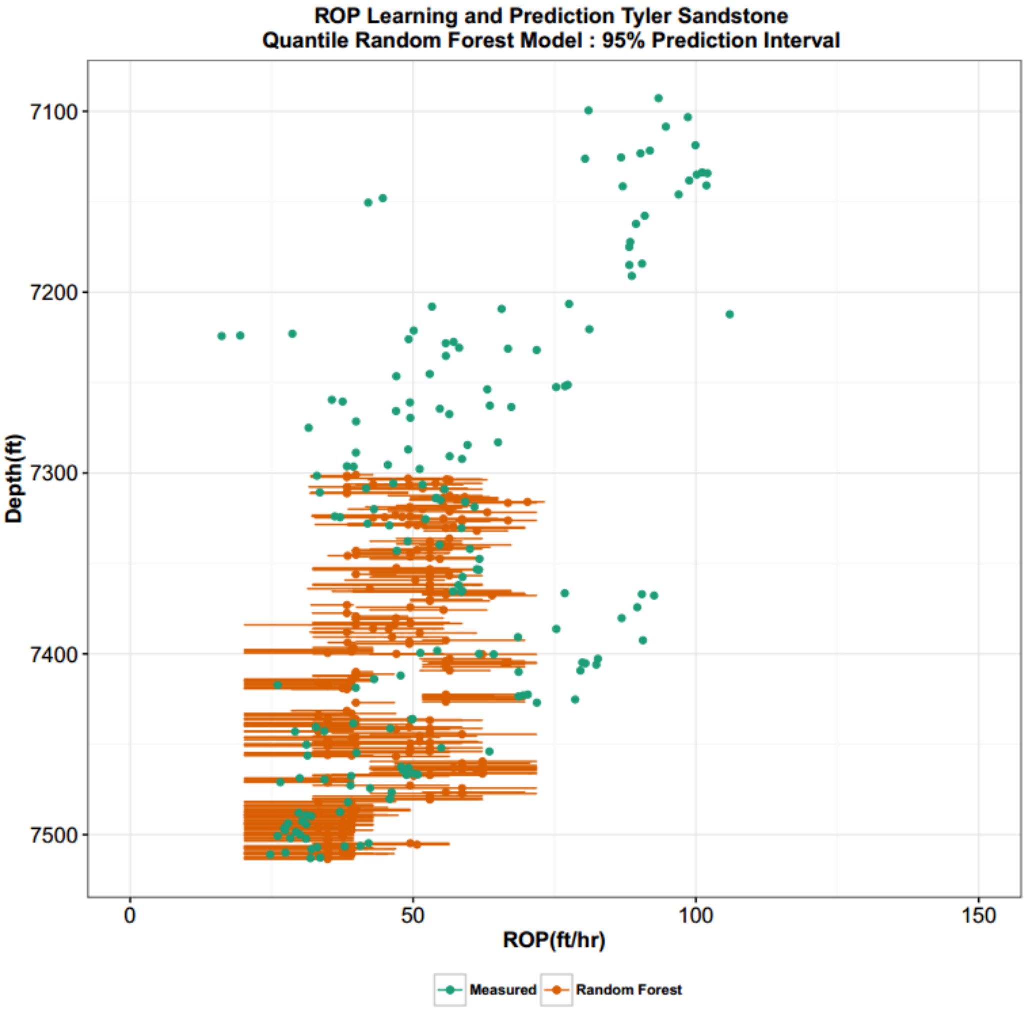

Figure A-2: Prediction interval for ROP predictions using quantile random forest approach. The first half of the formation is used to train the model. The model is used to predict ROP along with confidence intervals.

Appendix B: ROP Prediction using Time Series Forecasting

Time series approaches are very popular in finance, economics, and industrial engineering for modeling data in which are collected in a sequence. Drilling data is collected from sensors, sequentially in equally space-time intervals. The time intervals for the collection of data are often measured in terms of frequency (Hz) – the number of times a sample is collected in one second. In drilling, the sampling frequencies range from 1-15 Hz (sometimes as high as 128 Hz) but are often stored or transmitted at lower rates (Lesso, Ignova, Zeineddine, Burks, & Welch, 2011). A much more detail review and discussion of data collected in drilling have been covered in literature (Baumgartner, 2017). Hence, given the nature collection of drilling data, it is worth exploring whether ROP prediction is efficient using time series analysis.

B.1: THEORY

The main difference between time series analysis and linear regression is that the data are not necessarily independent and/or identically distributed. – which is an important assumption made while using linear regression. One defining characteristic of time series is that this is a list of observations where the ordering matters since there is a dependency and changing the order could change the meaning of the data being analyzed. The main time series modeling method approached in this section is the autoregressive integrated moving average (ARIMA) model.

Time series analysis (Fuller, 2009) deals with three main components: trend, seasonality, and residuals. The trend is the way the date moves long term, does it increase, decrease, or remain the same. Random noise, for example, would not have any trend, whereas drilling data where ROP is continuously decreasing (along the length of a well as rock gets more compacted) would have a decreasing trend (Figure B-1). Seasonality is a phenomenon which can occur due to the repeated nature of a measurement. For example, if a formation is laminated and drilled with constant drilling parameters, the ROP would be expected to show seasonality, where it would decrease while drilling through a hard formation and increase while drilling through a soft formation in a cyclic manner. In general, the dataset analyzed in this section does not contain such formations, hence further seasonality is not discussed further. In general, the seasonality component must be pre-determined and known.

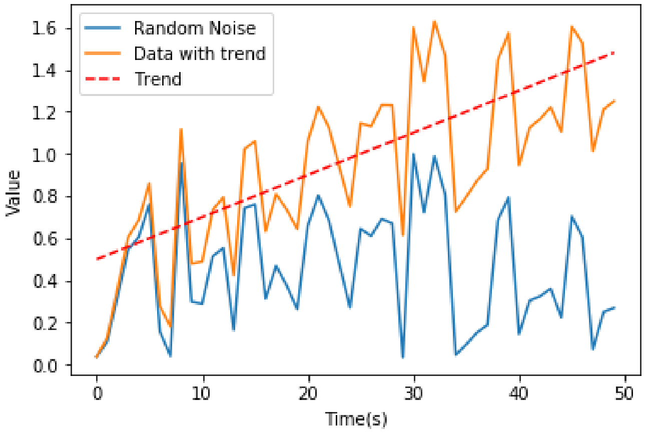

Figure B-1: Generated data showing two separate time series measured. The first series has no trend (blue) and the second series has a trend (orange) drawn with a red line.

The residuals are defined as data obtained after removing the trend and seasonality of the measurements. Modeling residuals accurately can result in a good prediction and are the primary responsibility of the ARIMA model.

Data used in ARIMA models must be stationary. Stationary data: data whose unconditional joint distribution does not change when shifted with time; the mean and variance of the data do not change over time. Hence, stationarity can be measured by checking if the mean and variance of the data change with time. In this section, a Dickey-

Fuller test is used to test for stationarity (Dickey & Fuller, 1979). The non-stationary time series can be transformed into a stationary one using: differencing, trend-removal, or transformations.

Differencing is to difference the data. From a given time series Z_t, a new time series can be defined as: $Y_i = Z_t - Z_{t-1}$. If the data contains a trend, a curve fit to the data can be subtracted from the time series to obtain the residuals. Since the purpose of the fit is to simply remove long term trend, a simple fit, such as a straight line, is typically used (Croarkin, Tobias, & Zey, 2002). If data have non-constant variance, a transformation such as the logarithm or square root transformations may stabilize the variance(Casella & Berger, 2002). Data are typically analyzed for trend and seasonality. Once removed from the data, the residuals are further analyzed or modeled.

The most common approach is the ARIMA model which is composed of two models itself, the autoregressive model (AR) and the moving average model (MA). The autoregressive model attempts to model the time series using the previously predicted value (Equation B-1).

$X_t = \delta + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} \ldots + \varphi_p X_{t-p-1} + A_t$, (Equation B-1)

where, X_t is the time series, A_t is white noise, and $\delta = (1 - \sum_{i=1}^{p} \varphi_i)\mu$ (μ is the process mean). An autoregressive model is a linear regression of the current value of the series against one or more prior values of the series. The number of dependent variables or value of *'p'* is called the order of the AR model. These models can be fit using standard least squares algorithms.

A moving average model is conceptually a regression model of the current value of the series against the white noise or random shocks of one or more prior values of the series. The random shocks at each point are assumed to come from the same distribution, typically a normal distribution, with zero mean and unit variance. The distinction in this model is that these random shocks are propagated to future values of the time series. Fitting the MA estimates is more complicated as compared to AR models because the error terms are not observable. This means that iterative non-linear fitting procedures need to be used in place of linear least squares. In essence, the residual obtained by subtracting the mean of the residual of the differenced data is itself fitted with a model.

$$X_t = \mu + \ \varepsilon_t + \theta_1 \varepsilon_{t-1} \ldots + \theta_q \varepsilon_{t-q}$$

where X_t is the time series, μ is the mean of the series, ε_{t-i} are white noise terms, and $\theta_1 \ldots \theta_q$ are the parameters of the model. The value of q is called the order of the MA model.

The two models are combined to model the time series in the form of an ARMA model (Box, Jenkins, Reinsel, & Ljung, 2015). The ARMA model assumes that the time series is stationary; if not, differencing can be used to enforce stationarity on the time series. An ARMA model with differencing is called ARIMA where 'I' stands for integrated.

The fitting of an ARIMA model – assuming single differencing – the parameters or orders for the AR and MA models are determined using autocorrelation and partial autocorrelation plots (Box et al., 2015). The autocorrelation plot maps the autocorrelation function of the time series: a correlation of the signal with a time-delayed version of itself. The time delay is often called lag. The partial autocorrelation at lag *k* is the autocorrelation between X_t and X_{t-k} that is not accounted for by lags *1* through *k−1*. Once the ARIMA model is fitted to the data, forecasts can be used to predict the next observation of the time series.

B.2 ROP MODELING USING ARIMA

Data from one formation – lodgepole limestone – is analyzed in this section for ROP prediction. The data are modeled as a time series, with time used as a proxy for depth. Since drilling 0.25 ft of depth occurs every 30 seconds on average, the data in this formation can be used for time series analysis (Figure B-2). For simplicity, the starting time of drilling has been set to 01-01-1970 at 00:00. The data is tested for stationarity using a Dickey-Fuller test (Table B-1). Since the p-value of the test is above 0.05 indicating that the data are not stationary, differencing is used to induce stationarity. The differenced time series has a p-value of 0; the stationary time series can be used for ARMA modeling.

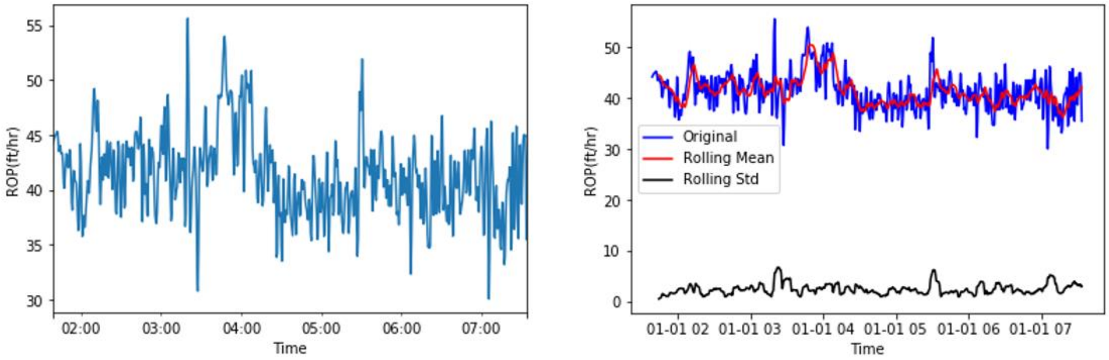

Figure B-2: (left) Measured ROP plotted as a time series; (right) time series imprinted with a mean (trend) and rolling standard deviation.

Table B-1: Results of the Dickey-Fuller test

	Time Series	Differenced Time Series
Test Statistic	-2.84	-7.61
p-value	0.0520	2.22E-11
#Lags Used	20	20
Number of Observations	689	688

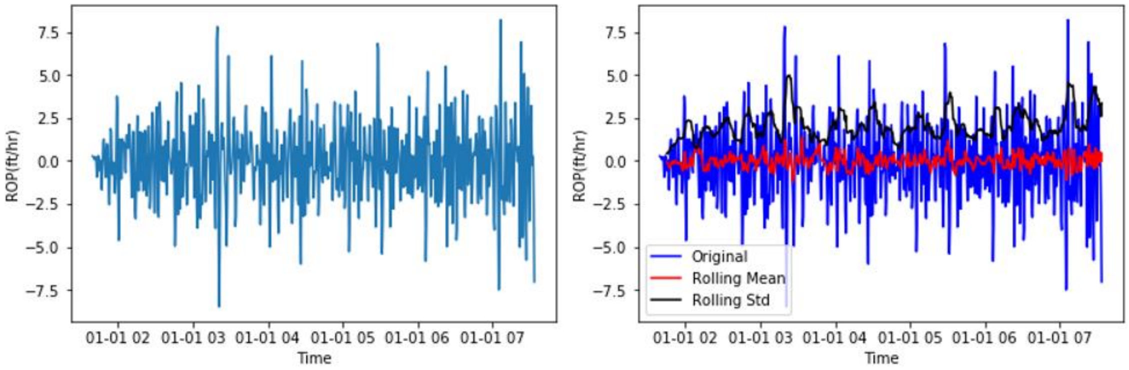

Figure B-3: (Left) Measured ROP plotted as a differenced time series; (Right) differenced time series imprinted with rolling mean (trend) and rolling standard deviation.

The autocorrelation and partial autocorrelation plots for the differenced time series have been plotted as shown in Figure B-4. This plot can be used to infer and determine the order of the ARMA model. Experience and value of the standard error to coefficient ratio guide picking the order of the ARMA model. Looking at Figure B-4, the autocorrelation plot suddenly spikes up, after which it faces an exponential decline. The partial autocorrelation function cyclically fluctuates. This is a characteristic which is commonly seen in time series (Box et al., 2015; Croarkin et al., 2002) and is best approached by using both AR and MA components in the model. More complicated methods of model selection using the Akaike Information Criterion (AIC) have not been discussed (Brockwell & Davis, 2016).

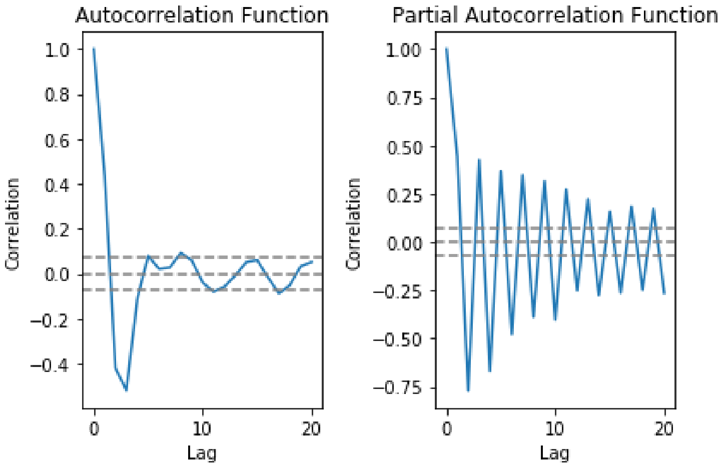

Figure B-4: (Left) Autocorrelation function plot for differenced time series; (Right) Partial autocorrelation plot for differenced time series.

The data are split into training and test sets using an 80/20 split. The best chosen parameters for this data-analysis was an AR(0) and MA(1) model. An out-of-sample forecast on the training test performs really well with low mean squared error (Figure B-5). The out-of-sample forecast is made one step at a time; the forecast length is only 0.25 ft. Despite this high accuracy, this forecast is not practically very valuable. Since the model used is of the order MA(1), only one preceding data point is useful for future predictions. The predicted value can be used for future prediction to obtain a long-term forecast (Figure B-5). This forecast is poor; the model after a few iterations just predicts a general ROP trend.

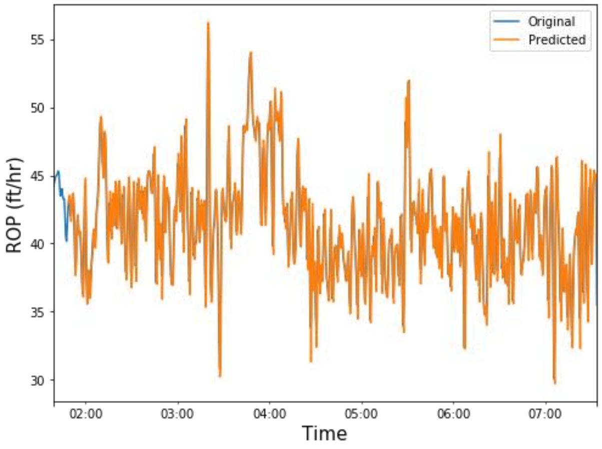

Figure B-5: One-step forecast of ARMA model for ROP time series in Lodgepole Limestone

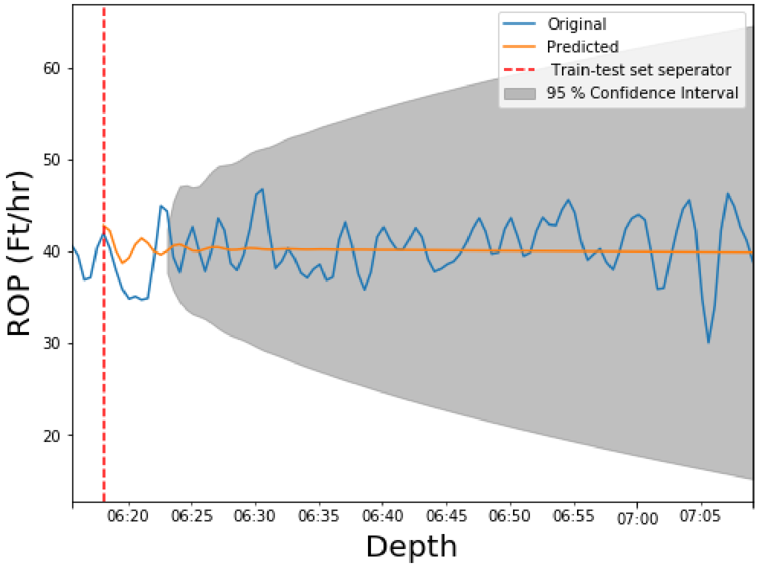

Figure B-6: Short term or test set forecast of ROP using the time series model

B-3 Conclusions

This section presented ROP prediction using time series modeling methods. A one-step forecast for ROP showed good predictions, whereas, short term or test set forecasts only returned a general ROP trend as the prediction. This analysis does not incorporate the physics of the wellbore but treats the ROP measured as an independent signal. While this analysis may work for short term or one-step predictions, it fails practically as shown in Figure B-6. It can be concluded that data-driven models are better suited for ROP modeling as compared to time series analysis.

Appendix C: Sampling during Training

The modeling and optimization of ROP or MSE using machine learning is dependent on the quality of the training data. This determines the optimization space available for optimizing these parameters. In chapter 4, the optimization space or window was restricted to the data observed in the training set to prevent model extrapolation. Additionally, since machine learning models are built purely based on training data, having a good training data set with varying parameters is imperative to the success of the optimization model. This section discusses sampling during training or data collection in the training set. The development of the DAS (Chang et al., 2014) by ExxonMobil discusses the concept of sampling the parameter space. Essentially, control parameters – RPM, WOB, and Flowrate –should be varied sufficiently during training. This expands the applicability of the model developed in earlier chapters. The DAS system (Chang et al., 2014) recommends that the maximum and minimum RPM, WOB and flow-rate window be set collectively based on the judgment of the drillers, drilling engineers, and rig supervisors. This window will depend on the bit, BHA, formation, hole section properties, and experience with previous wells drilled. The objective of defining a window is to avoid extrapolation by defining an optimization space limited by that window (Figure C-1). In addition to the parameter window, a step size of change is defined which can be used to determine the number of samples required to sufficiently cover the parameter space. For example, if the current WOB entering a formation is 15000 lbf, setting the minimum step size as 1000 lbf, would change the WOB to 14000 or 16000 lbf in the next iteration.

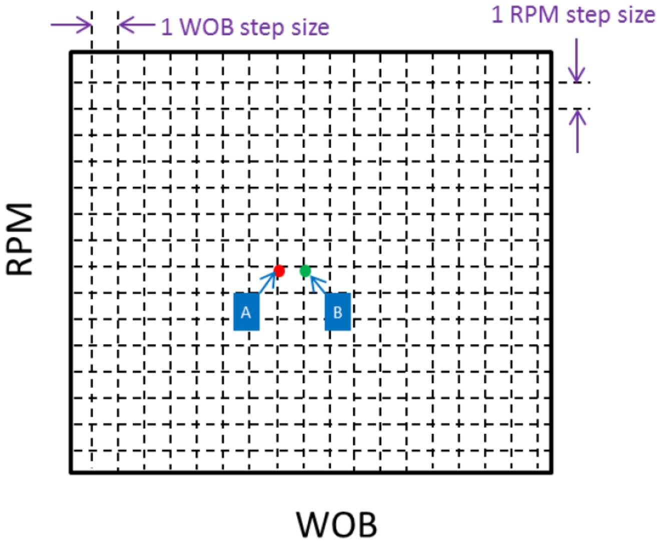

Figure C-1: Parameter space (or optimization space) as defined by drillers and engineers (Chang et al., 2014). The step size has been defined and plotted in the form of a grid.

Sampling the entire optimization space during training can take many iterations and a lot of time – if only one parameter is changed at a time. For example, if the number of levels are set to 10 for each factor (or drilling parameter) – WOB, RPM, and flowrate – then a total of 1000 “settings” would be required to sufficiently sample the entire optimization space. Additionally, this method can result in missing the optimum location completely (Dunn, 2010). Theory from the design of experiments (Dunn, 2010) can be used to aid this sampling process during training.

Design of experiments (DOE) can be applied here to decide the most optimal sampling method in the training set. Factors are parameters of interest, which are varied to change the outcome of the experiment; for drilling optimization, factors evaluated are WOB, RPM, and flow-rate. Levels are the total number of possible values these factors can take within an experiment; in this case, the level will be determined based on the optimization space window and the step size. For example, if the window for WOB lies between 10 and 20 klbs with a step size of 1 lb, the number of levels is 10.

A full factorial design runs through all possible levels of each factor. Hence the number of experiments needed are: $Levels_{WOB} * Levels_{RPM} * Levels_{FlowRate}$. This can increase exponentially depending on the number of levels. If each factor – WOB, RPM, flow-rate – has 10 levels each, a total of 1000 experiments or "setttings" will need to be evaluated. Incorporating the amount of time, it takes to see the change in ROP or MSE at the bit due to a newly applied WOB or RPM on the surface and operational time, this is infeasible.

One option is to reduce the number of factors – which can cause a decreased performance in the algorithm. Alternatively, a fractional factorial design can be utilized. Rather than sample at all locations, the fractional factorial design conducts a reduced number of experiments. Advanced methods like Plackett-Burman (Plackett & Burman, 1946) design can be used when dealing with a higher number of factors (most often not required for drilling operations). Alternatively, the driller and engineer can manually decide to vary different parameters based on intuition and field-based experience – as suggested by the DAS. Other algorithms which can potentially be used to explore the training set are Bayesian optimization and multi armed bandit algorithms.

The sampling of training data points for the data used in this formation are plotted below in the form of a 3-dimensional scatter plot in Figure C-2. Figures C-3, C-4, and C-5 plot the two-dimensional scatter plot for different control parameters (RPM, WOB, and flowrate). RPM and WOB have been well sampled in the training data based on their scatter. However, flowrate is varied much during training. This should be considered during ROP model optimization.

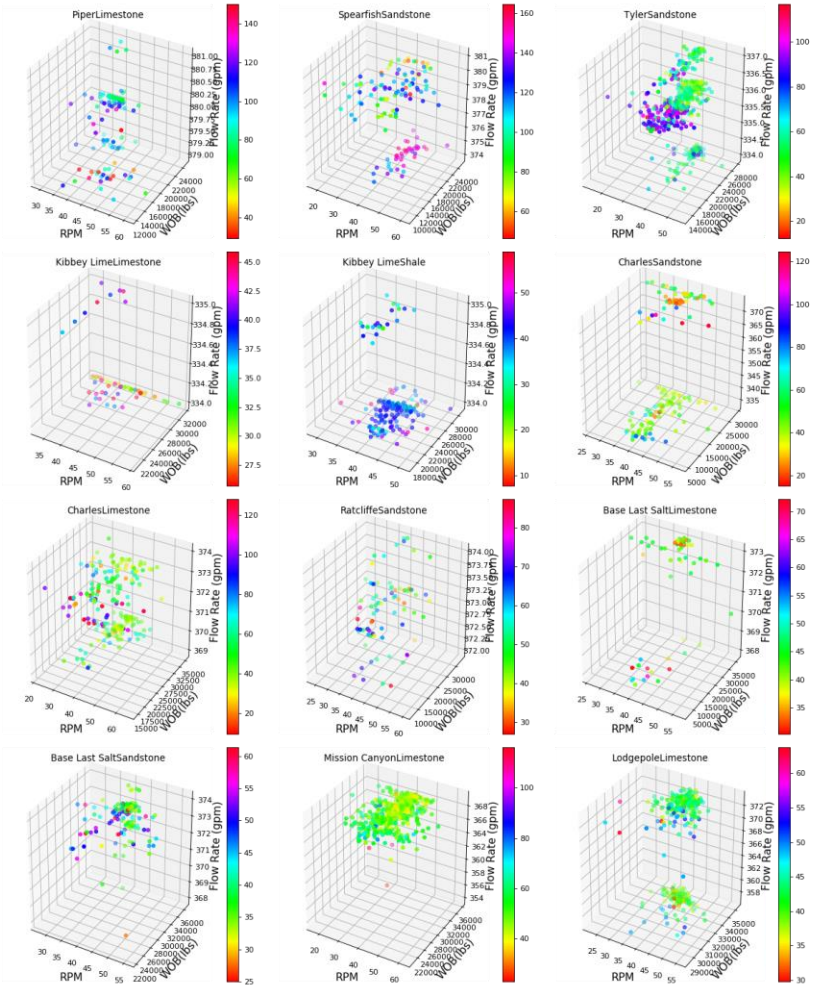

Figure C-2: Each subplot contains the 3-dimensional scatter plot for each feature involved in ROP or MSE optimization for all formations analyzed. The input feature space provides additional insight into the data collection during training. This can be useful in determining bounds for the optimization feature space since extrapolating to scarcely sampled regions can be dangerous. The plot shows that the scatter is well spread out in the training region of the dataset for RPM and WOB. However, flow rate has not been well explored during training

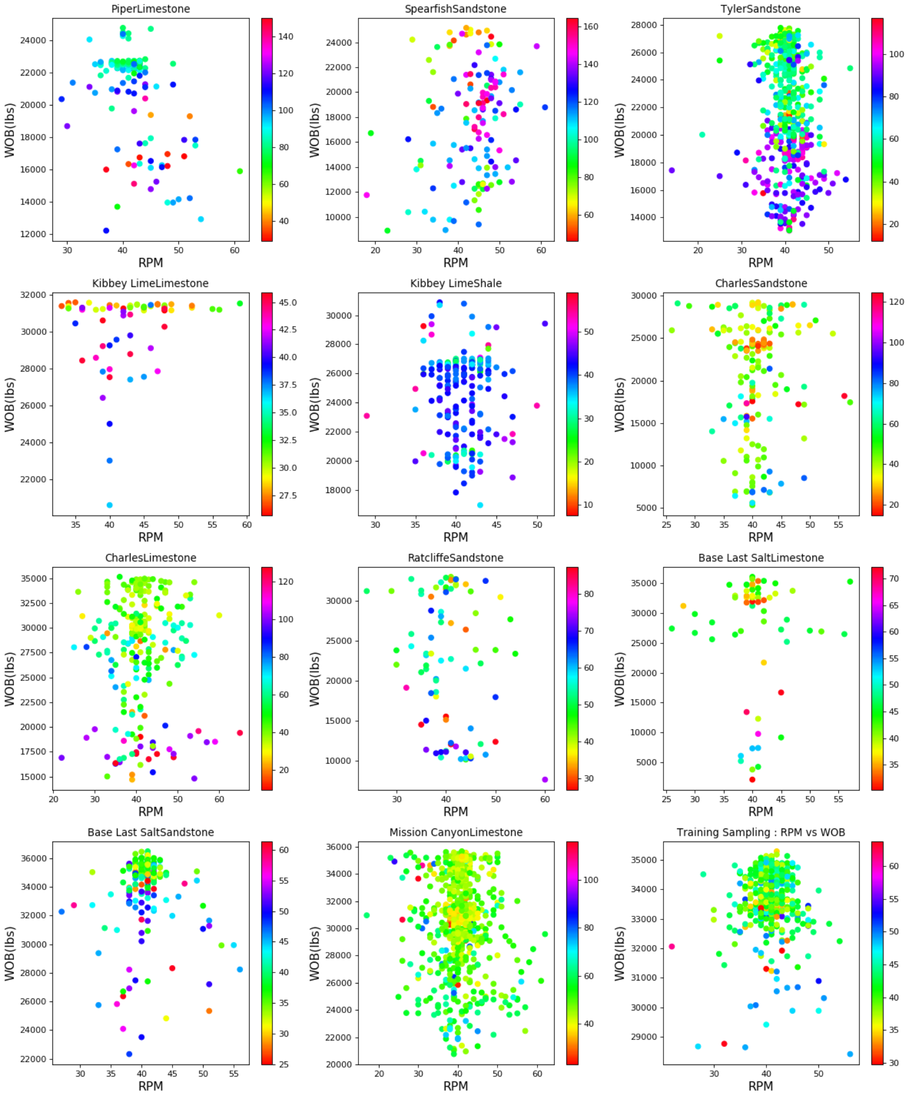

Figure C-3: WOB and RPM feature space plot for all formations. Each subplot contains the 2-dimensional scatter plot for RPM and WOB. The plots show that these features have been explored well. A uniform sampling method when applied in the training region can help avoid model extrapolation

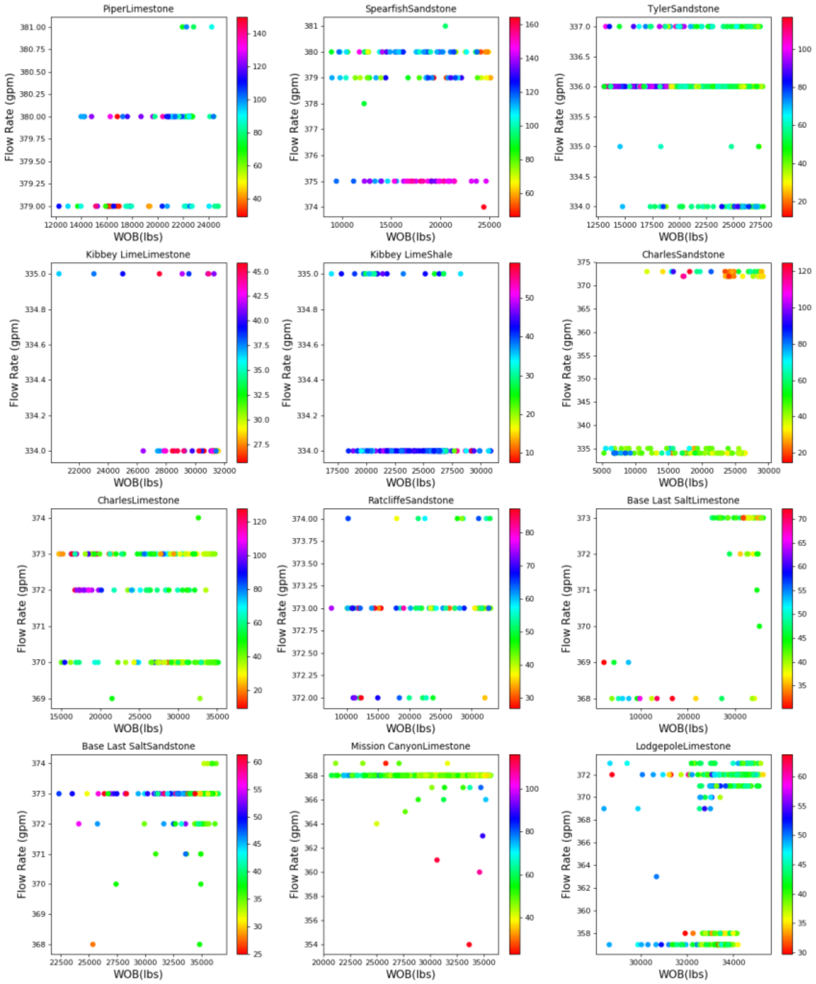

Figure C-4: WOB and Flowrate feature space plot for all formations. Each subplot contains the 2-dimensional scatter plot for flow rate and WOB. The plots show that flowrate has not been explored well. This should be considered while defining the feature space for parameter optimization

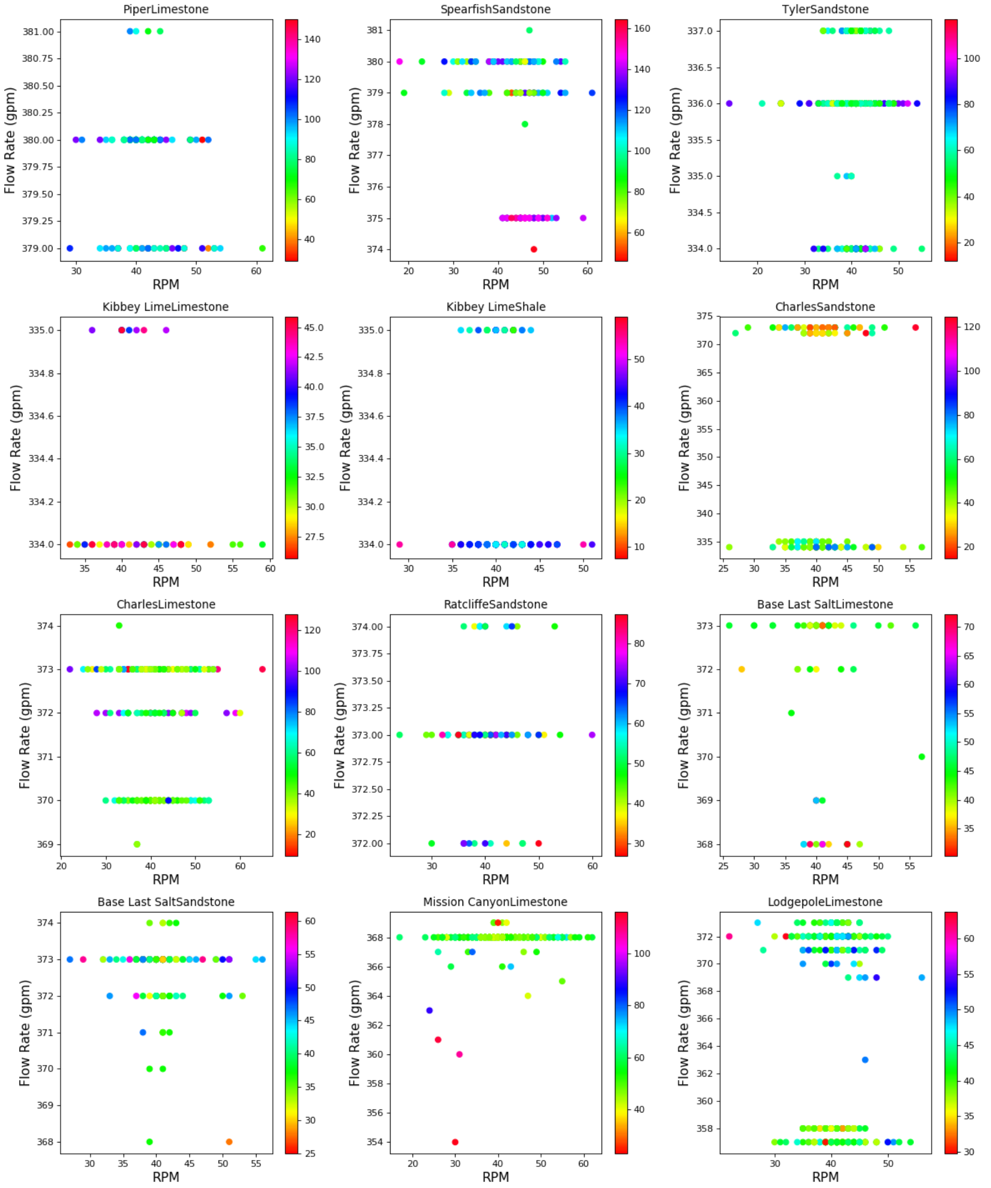

Figure C-5: Flowrate and RPM feature space plot for all formations. Each subplot contains the 2-dimensional scatter plot for RPM and flow rate. The plots show that flowrate has not been explored well. This should be considered while defining the feature space for parameter optimization

Restricting the optimization space for features that are not sampled well can help reduce erroneous model predictions ahead of the bit. In this case, flowrate can be restricted to only the values measured in the training dataset.

Appendix D: Drilling Model Optimization

Following the discussion of different optimization algorithms, this section aims to provide a pseudo code for the algorithms covered in chapter 4. The pseudo code is meant to be used as a template for each algorithm. The pseudo code provided here is a variant of each method and this may not match pseudo code provided in other text books or papers. The aim of this section is to provide a better intuition of each algorithm to the reader.

D.1 Eyeball Method

```
Loop over training set
    Find high ROP in training set (best X%)
    Determine drilling parameters which resulted in high ROP (optimum drilling parameters)
Best = Average optimum drilling parameters
Return Best
```

D.2 Random Search

```
Best (some initial random candidate solution)
Create a vector of samples using latin hyper cube sampling algorithm (S)
Loop over S :
    $S_i$ = random candidate solution
    If $S_i$ > Best:
        Best = $S_i$
Until we run out of time
Return Best
```

D.3 Simplex Method

```
Initialize 3 points (a,b,c)
Repeat
    EvalROP(a,b,c) and determine best, intermediate and worst
    Determine centroid
    reflected = reflect (worst) along centroid
    expanded = reflect worst along centroid & expanded > best
    contracted = reflect worst along centroid & contracted < best
```

```
        If (worst > refected & expanded & contracted) :
                Shrink
        Else:
                 Choose best amoung reflected, expanded and contracted
Until time runs out or three vertices converge
Return Best
```

D.4 Differential Evolution

```
Set alpha{mutation rate/rate of tweaking}
Set Popsize {population size}
P=[]
Q= Val
For I form 1 to Popsize:
        Pi= new random individual
Best= Max(ROP(Pi))
Repeat
        For each element in Pi
                EvalROP(Pi)
                If Q!= Val or  ROP(Qi) > ROP(Pi) :
                        Pi=Qi
                If Best = Val or ROP(Pi) > ROP(Best):
                        Best = Pi
        Q=P
        For each element in Q:
                Determine a,b,c {random vectors}
                D= a + alpha(b-c)
                Pi = new sample tweaked from (D,Qi)
Until we run out of time
Return Best
```

D.5 Particle Swarm Optimization

```
Set swarmsize {size of swarm}
Set alpha {proportion of velocity to be retrained}
Set beta {proportion of personal best to be retrained}
Set gamma {proportion of informant's best to be retrained}
Set delta {proportion of global best to be retrained}
Set epsilon {jump size of particle}
P=[]
For swarmsize times do
        P=P U {new random particle x with random initial velocity}
Best = Value
```

```
Repeat
    For each particle in P:
        evalROP(Pi)
        if Best != Value or ROP(Pi) > Best :
        Best = Pi
    For each particle in P:
        $X^g$= best particle location globally
        $X^l$ = best local particle location
        $X^p$ = best personal particle location
        For each dimension do
            b = rand(0,beta)
            c= rand(0,gamma)
            d= rand(0, delta)
            $vi = alpha * vi + b *(X_i^p - X_i) + c*(X_i^l – X_i) + d*(X_i^g - X_i)$
        For each Pi in P:
            Xi= Xi + Epislon x V
    Until we run of out time
Return Best
```

Appendix E: Retraining Frequency

The optimal parameters returned by the drilling optimization model can be applied ahead of the bit. It is recommended that the model is updated at a regular frequency to ensure that the model is up to date and best parameters are being used to drill. The model can be updated at a frequent interval (say every 4 stands) or when there is an increase in MSE or drilling vibrations.

This model update depends on the data collected for training. In general, once training is complete, after drilling the first stand to collect training data, the optimal parameters are used ahead of the bit are generally implemented using an automatic driller. During this implementation, surface parameters – WOB, RPM, flowrate – are held constant. Hence the data collected in "application mode" cannot be used as training data to re-train the model in case of a formation change. Varying surface parameters to collect training has been discussed in Appendix C. In order to update the model effectively, a "training mode" would have to be adapted where drilling surface parameters – WOB, RPM, Flow-rate – are changed to collect good training data. Once enough training data are collected – determined by sufficiently accurate ROP, TOB, and drilling vibrations models – optimal parameters can be recalculated and implemented.

Theoretically, retraining of these parameters is required if there is a formation change. Assuming that optimal parameters were initially calculated when the formation was first entered by the bit and then implemented using the auto driller, the MSE should remain relatively constant as long as the geology remains constant. The variability of the MSE should not exceed the variability seen in the collected torque data. A sharp change in MSE can indicate a formation change, drilling dysfunction or increased drilling vibrations which would require retraining. If MSE changes significantly – as determined using a hypothesis test – then the model can be retrained. In the case of re-training, practices discussed in Appendix C must be adopted. If the model is instead updated after drilling 5-10 stands, the last stand (or last joint) before which the model is updated should be used in "training mode" to gather good training data for the model update. It is suggested that the model is updated periodically (even if the formation remains the same) to account for dynamically changing downhole conditions such as mud weight and bit wear.

An alternative approach to determining formation change is to use a machine learning model for formation classification. Known formation boundaries from drilling pad wells can be used to train a machine learning model to classify formation based on surface and downhole measured drilling data. Such a model has been trained to classify 3 formations – Mission Canyon, Lodgepole, and Base Last Salt – using drilling surface and downhole parameters (WOB, RPM, Flow-rate, UCS, gamma log) as input features using the random forests algorithm. The accuracy of the classifier is high at 95% (with an F-1 score of 0.95). The misclassifications are not really a problem since they appear alone – not a continuous set of misclassifications. By using a logic-based rule – a minimum of 20 out of 25 data points in a sequence have to be classified correctly – formation change can be predicted very accurately during drilling (Figure E-1). This classifier can also be coupled with the change in MSE to make a more informed decision.

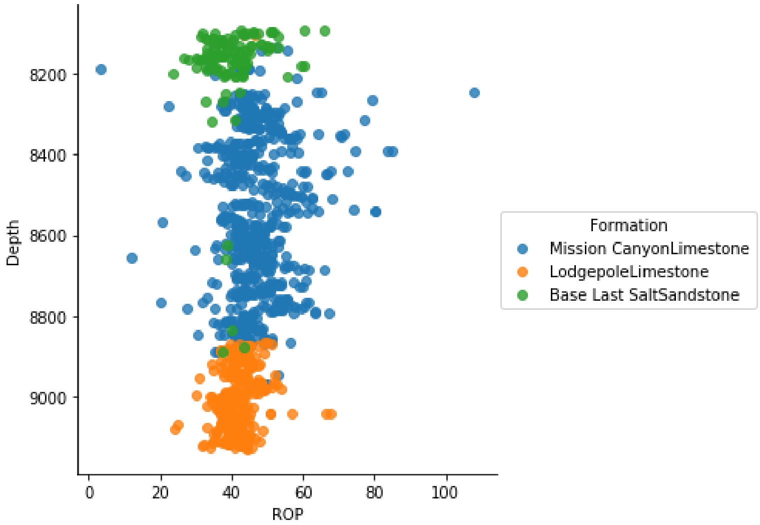

Figure E-1: Formation change classifier using surface and downhole drilling data

Appendix F: SMARTdrill – a tool which can optimize drilling

This section describes the development of an optimization tool – SMARTdrill – which can be used on the rig for drilling optimization. This tool has three major components (or layers): the input layer, the analytics layer and the output layer (Figure 1). The analytics layer is composed of the algorithms and models utilized for drilling optimization – which has been the topical focus of this dissertation (Chapter 2-7). The input and output layers are addressed in this section. The aim is to be able to use SMARTdrill as a stand-alone tool on the rig for drilling optimization.

Input Layer
Analytics Layer
Output Layer

Figure F-1: General schematic of the SMARTdrill tool

F.1 Input Layer

The input layer or data pipeline ensures that relevant data are given to the analytics layer for calculation. These data are gathered from different sources on the rig and used for modeling and calculation of optimal parameters by the analytics layer. The main sources of data required – by the analytics layer – to calculate optimal parameters are collected by the surface sensors (RPM, WOB, Flow-rate) and downhole sensors (downhole RPM, lateral and axial vibration).

Surface measured data are collected by surface sensors which process measurements, encode and send data (sometimes via the WITS protocol at the sensor itself) to the electronic drilling recorder (EDR) where it is displayed. The EDR acts as a data acquisition system for surface sensors. The data from the EDR is sent to a server directly or an intermediate data aggregator – which aggregate all data collected on a rig.

The downhole sensors which measure measured-while-drilling (MWD) parameters are sent to the MWD trailer operated by the service provider. The data from the downhole sensor is corrected and converted to the WITS-0 protocol using a computer and then transferred to a visual display such as the EDR. This data is also transferred directly to a server or an intermediate data aggregator – from which the operator can extract the data. Generally, the correction and transfer of downhole data takes a significant amount of time and cannot be obtained in real-time for most cases. However, technology is progressing rapidly – given the interest in drilling automation and data-driven decisions – and this can become a reality soon.

Alternatively, given the current complications in obtaining downhole data in real-time (or even close to real-time), better-equipped surface sensors which sample at a moderate frequency (5-150 Hz) can be used as a proxy. These surface sensors – such as Sting Sense$^{\text{TM}}$ – can be used in place of downhole data. While downhole data is preferred, it may not always be available on the rig – due to technological and cost constraints. Since the analytics layer consists of a data-driven model, a proxy for downhole sensor measurements can be easily incorporated into the model.

The main purpose of the input layer is to aggregate data collected from various sources, convert it to a format which can be processed by the analytics layer. The surface data can be received from the EDR or surface sensors directly in the WITS format; MWD computers or a corresponding EDR can transmit the downhole sensor data in a WITS or WITSML format. Once the input layer receives all data, it is converted to an amenable format – arrays or data frames – which can be directly used by the analytics layer. The collected data is then transferred to the analytics layer for computation and calculation. A flow chart of input data is visualized in Figure E-2. Alternatively, MO drill's input layer can retrieve data from an intermediate data aggregator if present on the rig.

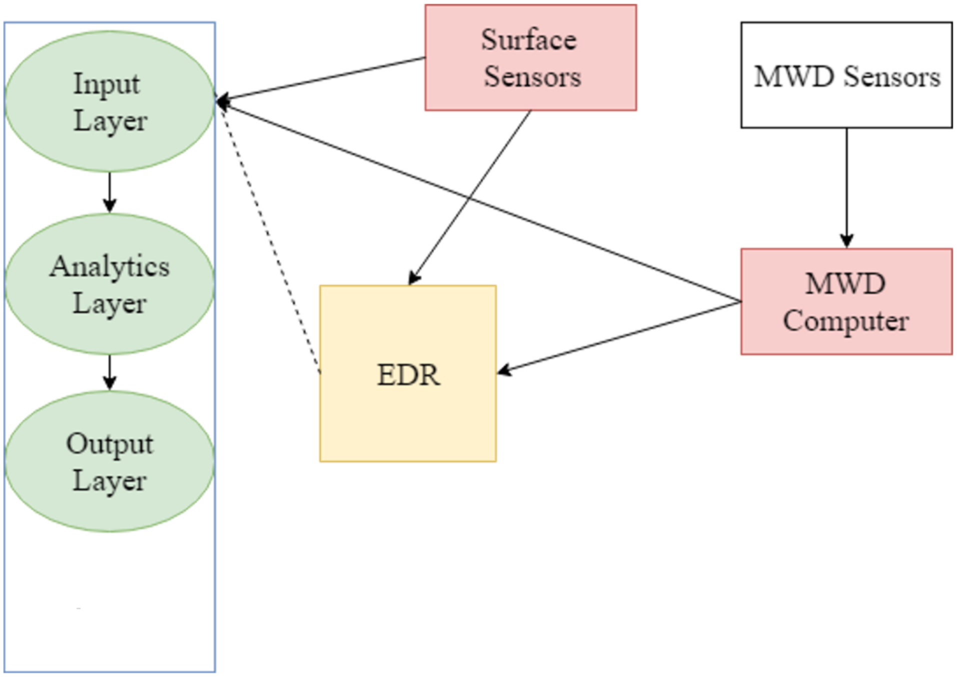

Figure F-2: Flowchart of input dataflow on the rig to SMARTdrill

F.2 Analytics Layer

Data from the input layer is transferred to the analytical layer of SMARTdrill. This data can be readily analyzed by a python script (or any other programming language). The data are used to calculate optimal drilling parameters which are passed to the output layer. The SMARTdrill tool is self-contained; a processing unit present in this tool can be used to perform the calculations as described in the earlier chapter using the TVOPT or CMOPT model. The processing or computing power which can be embedded in a tool may be significantly lower than those available on the cloud. Hence algorithms which are computationally efficient will be preferred. With advances in technology, more powerful processing tools can be embedded in SMARTdrill – expanding its capabilities. The outputs of the optimization model – optimal drilling parameters and any additional data can be passed to the output layer.

F.3 Output Layer

Calculated results and other relevant data are passed to the output layer. Practically the SMARTdrill tool would be can be applied either as a recommender system or controlling system.

The recommender system would advise the driller and engineer; specific settings of drilling operations parameters which would improve the current state of drilling using the TVOPT or CMLOPT model are displayed. This would require the output layer to communicate results to a visualization platform similar to the EDR (or the EDR itself if programmed to handle additional data). The role of this tool, in this case, would purely be advisory. The driller or engineer can choose to use these suggested drilling parameters ahead of the bit (Figure E-3). If accepted, these changes can be communicated to a tool such as the auto driller for implementation.

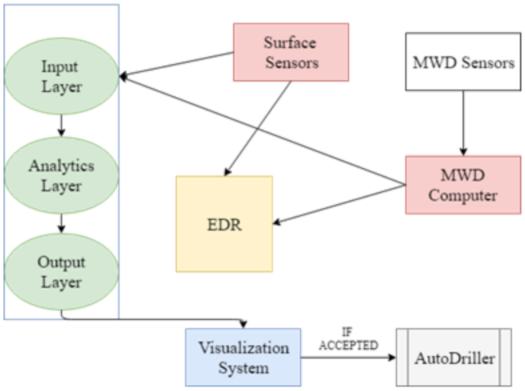

Figure F-3: Flowchart for SMARTdrill to be used as a recommender system

A controlling system would require the output layer to automatically communicate the drilling parameters to the auto driller so that it can be automatically implemented. Retraining and implementation are completely handled by the tool when required (as discussed in Appendix E). This would result in the tool being used as a part of a completely automated drilling system. The driller and/or engineer can take up an observatory role stepping in and manually override the system if required (Figure E-4).

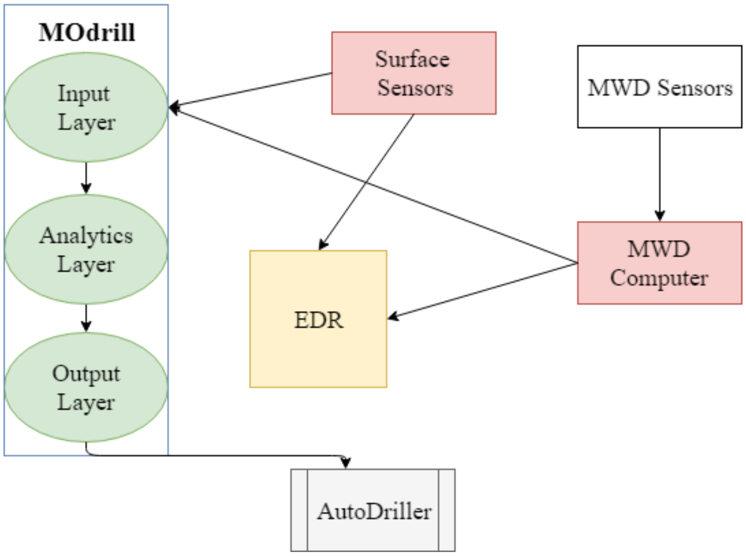

Figure F-4: Flowchart of MO drill being used as a controlling system

The output layer can encode data into the WITS (or WITSML) format to be passed to a server for storage. This can also include the models (ROP, TOB, MSE and vibrations), all calculated parameters, and optimal drilling parameters. This system of recording and storing this data can be useful for further analysis in the office or planning future wells.

References

Abdul Rahman, N., Mohaideen, A., Bakar, F. H., Tang, K. H., Maury, R., Cox, P., … Subroto, B. (2012). Solving Stick-Slip Dilemma: Dynamic Modeling System Significantly Reduces Vibration, Increases ROP by 54%. Society of Petroleum Engineers. https://doi.org/10.2118/161155-MS

Aerts, Guus; Brun, A. J. M. (2015). How to achieve 50 percent reduction in offshore drilling costs. Retrieved from https://www.mckinsey.com/industries/oil-and-gas/our-insights/how-to-achieve-50-percent-reduction-in-offshore-drilling-costs.

Arevalo, Y., & Fernandes, A. (2011). SPE 147747 Quantification of Drillstring Integrity Failure Risk Using Real-Time Vibration Measurements.

Arevalo, Y. I., Medina, Y., & Naslausky, A. (2011). Offshore Europe update :

Ashley, D. K., McNary, X. M., & Tomlinson, J. C. (2001). Extending BHA life with multi-axis vibration measurements. In *SPE/IADC drilling conference*. Society of Petroleum Engineers.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463). ACM press New York.

Bailey, J. R., Biediger, E. A. O., Gupta, V., Ertas, D., Elks, W. C., & Dupriest, F. E. (2008). Drilling Vibrations Modeling and Field Validation. *IADC/SPE Drilling Conference*, (1984), 1–15. https://doi.org/10.2118/112650-MS

Baumgartner, T. H. (2017). Maximizing the value of information from high-frequency downhole dynamics data.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Bilgesu, H. I. I., Tetrick, L. T. T., Altmis, U., Mohaghegh, S., & Ameri, S. (1997). A new approach for the prediction of rate of penetration (ROP) values. In *SPE Eastern Regional Meeting*. Society of Petroleum Engineers. https://doi.org/10.2118/39231-MS

Bingham, M. G. (1964). A new approach to interpreting Rock Drillability. *The Oil and Gas Journal*.

Bishop, C. M., & Christopher, M. B. (2016). *PATTERN RECOGNITION AND MACHINE LEARNING.* Springer-Verlag New York.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters*, *24*(6), 377–380.

Bourgoyne Jr, A. T., & Young Jr, F. S. (1974). A multiple regression approach to optimal drilling and abnormal pressure detection. *Society of Petroleum Engineers Journal*, *14*(04), 371–384.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Boyd, S., & Vandenberghe, L. (2010). *Convex Optimization*. *Optimization Methods and Software* (Vol. 25). https://doi.org/10.1080/10556781003625177

Bratli, R. K., Hareland, G., Stene, F., Dunsaed, G. W., & Gjelstad, G. (1997). Drilling optimization software verified in the North Sea. In *Latin American and Caribbean Petroleum Engineering Conference*. Society of Petroleum Engineers.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv Preprint ArXiv:1012.2599*.

Brockwell, P. J., & Davis, R. A. (2016). *Introduction to time series and forecasting*. springer.

Bybee, K. (2011). Real-Time Optimization of Drilling Parameters. *Journal of Petroleum Technology*, *63*(02), 48–49.

Caicedo, H. U., Calhoun, W. M., & Ewy, R. T. (2005). Unique ROP predictor using bit-specific coefficient of sliding friction and mechanical efficiency as a function of confined compressive strength impacts drilling performance. In *SPE/IADC Drilling Conference*. Society of Petroleum Engineers.

Casella, G., & Berger, R. L. (2002). *Statistical inference* (Vol. 2). Duxbury Pacific Grove, CA.

Censor, Y. (1977). Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, *4*(1), 41–59.

Chang, D. D.-L., Payette, G. S., Pais, D., Upstream, E., Wang, L., Bailey, J. R., & Mitchell, N. D. (2014). Field Trial Results of a Drilling Advisory System. In *IPTC 2014: International Petroleum Technology Conference* (pp. 1–13).

Chapman, C. D., Sanchez, J. L., De Leon Perez, R., Yu, H., De, L. P., & Yu, H. (2012). Automated Closed-Loop Drilling with ROP Optimization Algorithm Significantly Reduces Drilling Time and Improves Downhole Tool Reliability. *IADC/SPE Drilling Conference and Exhibition*, (2006). https://doi.org/http://dx.doi.org/10.2118/151736-MS

Christoforou, A. P., & Yigit, A. S. (2001). Active control of stick-slip vibrations: The role of fully coupled dynamics. In *SPE middle east oil show*. Society of Petroleum Engineers.

Cox, D. D., & John, S. (1992). A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on* (pp. 1241–1246). IEEE.

Croarkin, C., Tobias, P., & Zey, C. (2002). *Engineering statistics handbook*. NIST iTL.

Davis, J., Smyth, G. F., Bolivar, N., & Pastusek, P. E. (2012). Eliminating Stick-Slip by Managing Bit Depth of Cut and Minimizing Variable Torque in the Drillstring. In *IADC/SPE Drilling Conference and Exhibition*. Society of Petroleum Engineers.

Detournay, E., Richard, T., & Shepherd, M. (2008). Drilling response of drag bits: theory and experiment. *International Journal of Rock Mechanics and Mining Sciences*, *45*(8), 1347–1360.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, *74*(366a), 427–431.

Drillstring Vibrations and Vibration Modeling. (2010). Retrieved from https://www.slb.com/~/media/Files/drilling/brochures/drilling_opt/drillstring_vib_br

.pdf

Dufeyte, M. P., & Henneuse, H. (1991). Detection and monitoring of the slip-stick motion: field experiments. In *SPE/IADC drilling conference*. Society of Petroleum Engineers.

Dunlop, J., Isangulov, R., Aldred, W. D., Sanchez, H. a, Flores, J. L. S., Herdoiza, J. a, … Luppens, J. C. (2011). Increased Rate of Penetration Through Automation. *SPE/IADC Drilling Conference and Exhibition*, 1–11. https://doi.org/10.2118/139897-MS

Dunn, K. (2010). Process Improvement Using Data. *Online], Http://Learnche. Org/Pid*.

Dupriest, F. E., & Koederitz, W. L. (2005). Maximizing drill rates with real-time surveillance of mechanical specific energy. *SPE/IADC Drilling Conference*, (SPE/IADC 92194), 1–10. https://doi.org/10.2118/92194-MS

Dupriest, F. E., Witt, J. W., & Remmert, S. M. (2005). Maximizing ROP With Real-Time Analysis of Digital Data and MSE. *International Petroleum Technology Conference*, (1), 1–8. https://doi.org/10.2523/10607-MS

Ebrahimi, M., & Novieri, E. (2010). Cost Per Foot Reduction by Bit Run Optimization: A Simulation Study. In *Trinidad and Tobago Energy Resources Conference*. Society of Petroleum Engineers.

Eckel, J. R. (1967). Microbit studies of the effect of fluid properties and hydraulics on drilling rate. *Journal of Petroleum Technology*, *19*(04), 541–546.

Efron, B. (1987). Better bootstrap confidence intervals. *Journal of the American Statistical Association*, *82*(397), 171–185.

Efron, B., & Hastie, T. (2016). *Inference, Computer Age Statistical Inference : Algorithms Evidence, And Data Science*.

EIA. (2016). *Trends in U.S. Oil and Natural Gas Upstream Costs*. Retrieved from https://www.eia.gov/analysis/studies/drilling/pdf/upstream.pdf

Ertas, D., Bailey, J. R., Wang, L., & Pastusek, P. E. (2013). Drillstring Mechanics Model for Surveillance, Root Cause Analysis, and Mitigation of Torsional and Axial Vibrations. In *SPE/IADC Drilling Conference*. Society of Petroleum Engineers. https://doi.org/10.2118/163420-MS

Ertas, D., Bailey, J. R., Wang, L., Pastusek, P. E., Ertas, D., Wang, L., … Pastusek, P. E. (2013). Drillstring Mechanics Model for Surveillance, Root Cause Analysis, and Mitigation of Torsional and Axial Vibrations. *Society of Petroleum Engineers*, (December). https://doi.org/10.2118/163420-MS

Ertas, D., Bailey, J., Wang, L., & Pastusek, P. E. (2014). Drillstring Mechanics Model for Surveillance, Root Cause Analysis, and Mitigation of Torsional Vibrations. *Society of Petroleum Engineers*. https://doi.org/10.2118/163420-PA

Ertas, M. D., Bailey, J. R., Burch, D. N., Wang, L., Pastusek, P. E., & Sundararaman, S. (2015, March 10). Methods to estimate downhole drilling vibration amplitude from surface measurement. United States of America: Google Patents.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*

(Vol. 1). Springer series in statistics New York.

Fuller, W. A. (2009). *Introduction to statistical time series* (Vol. 428). John Wiley & Sons.

Gandelman, R. A. (2012). Prediçao da ROP e otimizaçao em tempo real de parâmetros operacionais na perfuraçao de poços de petróleo offshore. Ph. D. thesis, UFRJ.

Gerbaud, L., Menand, S., & Sellami, H. (2006). PDC bits: all comes from the cutter rock interaction. In *IADC/SPE Drilling Conference* (p. 1).

Ghasemloonia, A., Rideout, D. G., & Butt, S. D. (2015). A review of drillstring vibration modeling and suppression methods. *Journal of Petroleum Science and Engineering*, *131*, 150–164.

Gjelstad, G., Hareland, G., Nikolaisen, K. N., & Bratli, R. K. (1998). The Method of Reducing Drilling Costs More Than 50 Percent Optimizing, 1–7.

Goldstein, E. B., Coco, G., Murray, A. B., & Green, M. O. (2014). Data-driven components in a model of inner-shelf sorted bedforms: a new hybrid model. *Earth Surface Dynamics*, *2*(1), 67.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157–1182.

Hareland, G., & Rampersad, P. R. (1994). Drag-bit model including wear. In *SPE Latin America/Caribbean Petroleum Engineering Conference*. Society of Petroleum Engineers.

Hegde, C., Daigle, H., & Gray, K. (2018). Performance comparison of algorithms for real-time rate of penetration optimization in drilling using data-driven models. *SPE Journal*.

Hegde, C., Daigle, H., Millwater, H., & Gray, K. (2017). Analysis of rate of penetration (ROP) prediction in drilling using physics-based and data-driven models. *Journal of Petroleum Science and Engineering*, *159*. https://doi.org/10.1016/j.petrol.2017.09.020

Hegde, C., Daigle, H., Millwater, H., & Gray, K. (2017). Analysis of rate of penetration (ROP) prediction in drilling using physics-based and data-driven models. *Journal of Petroleum Science and Engineering*, *159*, 295–306. https://doi.org/10.1016/j.petrol.2017.09.020

Hegde, C., & Gray, K. E. (2017). Use of machine learning and data analytics to increase drilling efficiency for nearby wells. *Journal of Natural Gas Science and Engineering*, *40*, 327–335. https://doi.org/10.1016/j.jngse.2017.02.019

Hegde, C. M. (2016). *Application of statistical learning techniques for rate of penetration (ROP) prediction in drilling*. The University of Texas at Austin.

Hegde, C. M., Awan, O. K., & Wiemers, T. (2018). Application of Real-time Video Streaming and Analytics To Breakdown Rig Connection Process. In *Offshore Technology Conference*. https://doi.org/10.4043/28742-MS

Hegde, C. M., & Gray, K. E. (2018). Evaluation of coupled machine learning models for drilling optimization. *Journal of Natural Gas Science & Engineering*.

Hegde, C. M., Wallace, S. P., & Gray, K. E. (2015). Use of Regression and

Bootstrapping in Drilling Inference and Prediction. In *SPE Middle East Intelligent Oil and Gas Conference and Exhibition*. Society of Petroleum Engineers.

Hegde, C., Wallace, S., & Gray, K. (2015). Real time prediction and classification of torque and drag during drilling using statistical learning methods. *SPE Eastern Regional Meeting*, *2015–Janua*. https://doi.org/10.2118/177313-MS

Hegde, C., Wallace, S., & Gray, K. (2015). Using trees, bagging, and random forests to predict rate of penetration during drilling. In *SPE Middle East Intelligent Oil and Gas Conference and Exhibition*. Society of Petroleum Engineers.

Jaggi, A., Upadhaya, S., & Chowdhury, A. R. (2007). Successful PDC/RSS Vibration Management Using Innovative Depth-of-Cut Control Technology: Panna Field, Offshore India. In *SPE/IADC Drilling Conference*. Society of Petroleum Engineers.

Jahanbakhshi, R. . K. R. (2012). Real-time Prediction of Rate of Penetration During Drilling Operation In Oil And Gas Wells. *American Rock Mechanics Association*, *53*(3), 127.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Jansen, J. D. (1991). Non-linear rotor dynamics as applied to oilwell drillstring vibrations. *Journal of Sound and Vibration*, *147*(1), 115–135.

Jansen, J. D. (1993). *Nonlinear dynamics of oilwell drillstrings*.

Janwadkar, S. S., Fortenberry, D. G., Roberts, G. K., Kramer, M., Trichel, D. K., Rogers, T., … Isbell, M. R. (2006). BHA and Drillstring Modeling Maximizes Drilling Performance in Lateral Wells of Barnett Shale Gas Field of N. Texas. In *SPE Gas Technology Symposium*. Society of Petroleum Engineers.

Johancsik, C. A., Friesen, D. B., & Dawson, R. (1984). Torque and drag in directional wells-prediction and measurement. *Journal of Petroleum Technology*, *36*(06), 987–992.

Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760–766). Springer.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Kyllingstad, A., & Halsey, G. W. (1987). A Study of Slip-stick Motion of the Bit. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Ledgerwood III, L. W., Jain, J. R., Hoffmann, O. J., & Spencer, R. W. (2013). Downhole measurement and monitoring lead to an enhanced understanding of drilling vibrations and polycrystalline diamond compact bit damage. *SPE Drilling & Completion*, *28*(03), 254–262.

Lesso, W. G., Ignova, M., Zeineddine, F., Burks, J. M., & Welch, J. B. (2011). Testing the Combination of High Frequency Surface and Downhole Drilling Mechanics and Dynamics Data Under a Variety of Drilling Conditions. In *SPE/IADC Drilling Conference and Exhibition*. Society of Petroleum Engineers.

Li, Z., & Guo, B. (2007). Analysis of longitudinal vibration of drill string in air and gas drilling. In *Rocky Mountain Oil & Gas Technology Symposium*. Society of Petroleum Engineers.

Luke, S. (2009). *Essentials of metaheuristics* (Vol. 113). Lulu Raleigh.

Lummus, J. L. (1970). Drilling optimization. *Journal of Petroleum Technology*, *22*(11), 1–379.

Macpherson, J. D., Mason, J. S., & Kingman, J. E. E. (1993). Surface Measurement and Analysis of Drillstring Vibrations While Drilling. *SPE/IADC Drilling Conference*, (SPE/IADC 25777), 953–963. https://doi.org/10.2118/25777-MS

Mahyari, M. F., Behzad, M., & Rashed, G. R. (2010). Drill string instability reduction by optimum positioning of stabilizers. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *224*(3), 647–653.

Marquez, M. B. S., Boussaada, I., Mounier, H., & Niculescu, S.-I. (2015). Analysis and Control of Oilwell Drilling Vibrations: A Time-Delay Systems Approach, 282. https://doi.org/10.1007/978-3-319-15747-4

Maurer, W. C. (1962). The" perfect-cleaning" theory of rotary drilling. *Journal of Petroleum Technology*, *14*(11), 1–270.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(Jun), 983–999.

Menand, S., & Mills, K. (2017). Use of Mechanical Specific Energy Calculation in Real-Time to Better Detect Vibrations and Bit Wear While Drilling. *Paper AADE-17-NTCE-0332017 Proceedings of the 2017 AADE National Technical Conference and Exhibition Held at the Hilton Houston North Hotel, Houston, Texas, April 11-12,*.

Menand, S., Sellami, H., Tijani, M., Stab, O., Dupuis, D. C., & Simon, C. (2006). Advancements in 3D drillstring mechanics: from the bit to the topdrive. In *IADC/SPE drilling conference*. Society of Petroleum Engineers.

Meng, C. U. I., Haige, W., Jinying, Z., Liu, C. U. I., & Zhixue, C. (2015). Optimizating Drilling Operating Parameters With Real-Time Surveillance and Mitigation System of Downhole Vibration in Deep Wells. *Advances in Petroleum Exploration and Development*, *10*(1), 22–26.

Minasny, B., & McBratney, A. B. (2005). The Matérn function as a general model for soil variograms. *Geoderma*, *128*(3–4), 192–207.

Miranda, L. J. V. (2018). PySwarms: a research toolkit for Particle Swarm Optimization in Python. *Journal of Open Source Software*, *3*, 21.

Mockus, J. (1994). Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, *4*(4), 347–365.

Motahhari, H. R., Hareland, G., & James, J. A. (2010). Improved drilling efficiency technique using integrated PDM and PDC bit parameters. *Journal of Canadian Petroleum Technology*, *49*(10), 45–52.

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, *7*(4), 308–313. https://doi.org/10.1093/comjnl/7.4.308

Ng, A. (2000). CS229 Lecture notes. *CS229 Lecture Notes*, *1*(1), 1–3.

Ng, A., Ngiam, J., Foo, C. Y., Mai, Y., Suen, C., Coates, A., … Sameep Tandon. (2015).

Deep Learning Tutorial. *University of Stanford*. https://doi.org/10.1007/s13218-012-0198-z

Nygaard, R., Hareland, G., Budiningsih, Y., Terjesen, H. E., & Stene, F. (2002). Eight Years Experience with a Drilling Optimization Simulator in the North Sea. *IADC/SPE Asia Pacific Drilling Technology*, (IADC/SPE 77247).

P. Murphy, K. (1991). *Machine Learning: A Probabilistic Perspective*. *Machine Learning: A Probabilistic Perspective*. https://doi.org/10.1007/SpringerReference_35834

Pastusek, P. E., Brackin, V. J., & Lutes, P. J. (2005). A fundamental model for prediction of hole curvature and build rates with steerable bottomhole assemblies. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Patil, P. A., & Teodoriu, C. (2013). Model development of torsional drillstring and investigating parametrically the stick-slips influencing factors. *Journal of Energy Resources Technology*, *135*(1), 13103.

Pavone, D. R., & Desplans, J. P. (1994). Application of high sampling rate downhole measurements for analysis and cure of stick-slip in drilling. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Payette, G. S., Pais, D., Spivey, B., Wang, L., Bailey, J. R., Pastusek, P., & Owens, M. (2015). Mitigating Drilling Dysfunction Using a Drilling Advisory System: Results from Recent Field Applications. *International Petroleum Technology Conference*. https://doi.org/10.2523/IPTC-18333-MS

Payette, G. S., Spivey, B. J., Wang, L., Upstream, E., Bailey, J. R., Company, D., … Systems, P. (2017). A Real-Time Well-Site Based Surveillance and Optimization Platform for Drilling: Technology, Basic Workflows and Field Results. In *SPE/IADC Drilling Conference and Exhibition*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Pees, S. T. (2004). Oil history. *Accessed On*, *6*(25), 4.

Petroleum, B. (2016). BP Energy Outlook 2016 edition. *Outlook To*, *2035*.

Plackett, R. L., & Burman, J. P. (1946). The design of optimum multifactorial experiments. *Biometrika*, *33*(4), 305–325.

Rashidi, B., Hareland, G., & Nygaard, R. (2008). Real-Time Drill Bit Wear Prediction by Combining Rock Energy and Drilling Strength Concepts. *Abu Dhabi International Petroleum Exhibition and Conference , Abu Dhabi, UAE*.

Robnett, E. W., Heisig, G., McGinley, P. J., & Macpherson, J. D. (2002). Real-Time Downhole Drilling Process Data Complement Surface Data in Drilling Optimization. *IADC/SPE Asia Pacific Drilling Technology*, (IADC/SPE 77248), 1–10. https://doi.org/10.2118/77248-MS

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive Modeling*, *5*(3), 1.

Sanderson, D., Energy, X. T. O., Payette, G. S., Spivey, B. J., Upstream, E., Bailey, J. R., … Systems, P. (2017). Field Application of a Real-Time Well-Site Drilling

Advisory System in the Permian Basin. In *Unconventional Resources Technology Conference (URTeC)*. https://doi.org/10.15530/urtec-2017-2670861

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Self, R. V. V, Atashnezhad, A., & Hareland, G. (2016). Use of a Swarm Algorithm to Reduce the Drilling Time through Measurable Improvement in Rate of Penetration. *50th US Rock Mechanics/Geomechanics Symposium*, *1*.

Shuttleworth, N. E., Van Kerkoerle, E. J., Folmer, D. R., & Foekema, N. (1998). Revised drilling practices, VSS-MWD tool successfully addresses catastrophic bit/drillstring vibrations. In *IADC/SPE drilling conference*. Society of Petroleum Engineers.

Shyu, R.-J. (1989). Bending vibration of rotating drill strings. Massachusetts Institute of Technology.

Sill, J., Takács, G., Mackey, L., & Lin, D. (2009). Feature-weighted linear stacking. *ArXiv Preprint ArXiv:0911.0460*.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems* (pp. 2951–2959).

Soares, C. (2015). *Development and applications of a new system to analyze field data and compare rate of penetration (ROP) models*. The University of Texas at Austin.

Soares, C., Daigle, H., & Gray, K. (2016). Evaluation of PDC bit ROP models and the effect of rock strength on model coefficients. *Journal of Natural Gas Science and Engineering*, *34*, 1225–1236. https://doi.org/10.1016/j.jngse.2016.08.012

Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, *29*(2), 143–151.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

Taguchi, G. (1986). *Introduction to quality engineering: designing quality into products and processes*.

Tansev, E. (1975). A Heuristic Approach to Drilling Optimization. *Spe*.

Theloy, C. (2014). *Integration of geological and technological factors influencing production in the Bakken play, Williston Basin*. Colorado School of Mines.

Thomson, W. (1996). *Theory of vibration with applications*. CRC Press.

Trichel, D. K., Isbell, M., Brown, B., Flash, M., McRay, M., Nieto, J., & Fonseca, I. (2016). Using Wired Drill Pipe, High-Speed Downhole Data, and Closed Loop Drilling Automation Technology to Drive Performance Improvement Across Multiple Wells in the Bakken. In *IADC/SPE Drilling Conference and Exhibition*. Society of Petroleum Engineers.

VanderPlas, J. (2016). *Python data science handbook: Essential tools for working with data*. " O'Reilly Media, Inc."

Wager, S., Hastie, T., & Efron, B. (2014). Confidence intervals for random forests: the jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*, *15*(1), 1625–1651.

Walker, B. H., Black, A. D., Klauber, W. P., Little, T., & Khodaverdian, M. (1986). Roller-bit penetration rate response as a function of rock properties and well depth. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Wallace, B. C., & Dahabreh, I. J. (2014). Improving class probability estimates for imbalanced data. *Knowledge and Information Systems*, *41*(1), 33–52.

Wallace, S. P., Hegde, C. M., & Gray, K. E. (2015). A System for Real-Time Drilling Performance Optimization and Automation Based on Statistical Learning Methods. In *SPE Middle East Intelligent Oil and Gas Conference and Exhibition*. https://doi.org/10.2118/176804-MS

Wang, J., Kumaran, K., Xu, P., Sowers, S. F., Wang, L., Bailey, J. R., … Bangaru, S. S. (2010, June 28). Drilling Advisory Systems and Methods Based on At Least Two Controllable Drilling Parameters. Google Patents.

Winters, W. J., Warren, T. M., & Onyia, E. C. (1987). Roller bit model with rock ductility and cone offset. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.

Yi, P., Kumar, A., & Samuel, R. (2015). Realtime rate of penetration optimization using the shuffled frog leaping algorithm. *Journal of Energy Resources Technology*, *137*(3), 32902.

Zhu, X., Tang, L., & Yang, Q. (2014). A Literature Review of Approaches for Stick-Slip Vibration Suppression in Oilwell Drillstring. https://doi.org/10.1155/2014/967952