Exploring Parameter Interactions and Deep Learning for Modeling Pressure Related Downhole Safety Conditions during Drilling

By

©Augustine Uhunoma Osarogiagbon

A dissertation submitted to the School of Graduate Studies

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

St. John's, Newfoundland, Canada

June 2021

DEDICATION

This work is dedicated to almighty God who made this possible, my wife, Esther Chinaza Osarogiagbon, and my son, Eghosa Ebubechukwu Osarogiagbon who were with me and encouraged me during my studies. I want to specially appreciate my mother for all she has done for me right from my birth.

ABSTRACT

Drilling for petroleum is technically engaging, considering the potentially huge risk involved. Blowout due to uncontrolled kick represents a scenario that is to be avoided due to disastrous consequences, e.g. huge financial loss, environmental damage, and death of personnel. Kick occurrence can be prevented if the pore pressure is correctly estimated and the proper drilling mud weight employed. Pore pressure prediction is done in shale lithology; hence a fast means of proper lithology identification is important for pore pressure prediction. Monitoring downhole for pore pressure related hazard therefore includes but is not limited to: monitoring for kick occurrence, monitoring for abnormal pore pressure, and monitoring for changes in lithology for adequate pore pressure prediction.

In the field of data science, deep learning is gaining significant interest, which is likely due to its potentials and successful applications. Researchers have begun to explore deep learning in several areas with close affinity to drilling engineering, such as lithology identification, drilling rig state determination, generating logging/other drilling parameters, detecting downhole events, and detecting abnormality in data. Therefore, this serves as a motivation to take advantage of deep learning capabilities in monitoring downhole conditions during drilling to prevent pore pressure based hazardous events.

In this dissertation, a novel methodology for kick detection using drilling parameters is presented. Likewise, a novel methodology for predicting the shaliness of a rock formation using drilling parameters is also presented. These methodologies utilized deep learning algorithms in order to achieve the desired objectives. Results obtained using field data justified the development of methodologies with the capability to capture sequential dependencies. Cost represents a significant factor for utilizing drilling parameters in comparison to the use of highly

sophisticated/expensive downhole sensors. As part of this dissertation, a novel approach for pore pressure prediction from porosity and resistivity measurement is presented. The aim of combining porosity and resistivity is to explore how the interrelationship between them can enhance pore pressure prediction. The methodology developed for combining porosity and resistivity performed better than the conventional approach based on field data. Machine learning was also employed for pore pressure prediction and better result was also achieved in comparison to conventional approach based on the same field data. In summary, this dissertation presents several novel methodologies for monitoring different aspects of downhole conditions from downhole lithology to downhole drilling events which are important for improved drilling safety.

ACKNOWLEDGEMENTS

I want to express my gratitude to my Supervisors: Dr. Venkatesan Ramachandran for his mentorship both in academic, career, and character-building; Dr. Faisal Khan for his support in terms of academic mentorship, time management, accommodation, and financial support; and Dr. Paul Gillard for his academic mentorship, encouragement and career development. I want to thank Dr. Olalere Oloruntobi for his immense advice in the area of drilling engineering. I also appreciate Dr. Steven Butt and members of the Drilling Technology Laboratory for the knowledge I gained through the numerous presentations I participated. I thank members of the Centre For Risk, Integrity And Safety Engineering for the numerous advice on research methodologies and article publications during my studies. I want to thank Dr. Dennis Peters, Dr. Cheng Li, Dr. Cecilia Moloney, Dr. Samuel Ike, and Dr. Oliver Stueker for their support and encouragement during My PhD program. I want to thank my friends Charles Osaretin, Emmanuel Otasowie, Isi Edeoghon, Jeffery Odianosen, John Unegbu, Kenneth Edomwandekhoe, Nixon Obi, Osasuyi Odia, Peter Ogban, Prophet Aikhoje, Somadina Muojeke, Sylvester Nkebakwu, Wilson Ozah, and many others too numerous to mention.

I want to acknowledge Dr. Jincai Zhang and Dr. Olalere for assisting me with field data for my PhD, as well as the owners of the software WebPlotDigitizer and UN-SCAN-IT which I also used in obtaining data for my research work. I want to specially thank Niger Delta Development Commission for offering me a scholarship towards my PhD, as well as the financial support provided through Canada Research Chair Tier 1 Program in Offshore Safety and Risk Engineering.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

1. Introduction

1.1 Background

Drilling thousands of feet into the ground in search of resources for human energy needs can be very expensive and risky. Popular forms of drilling include drilling sedimentary rocks for hydrocarbon and drilling geothermal wells to explore for heat energy (Allahvirdizadeh, 2020; Lukawski et al., 2014). In this dissertation, the focus is on drilling sedimentary rock for petroleum. When drilling for petroleum, the pore pressure or geopressure which refers to the pressure of the fluid within the pores of the rock can pose a serious threat to the safety of lives, equipment and environment within the vicinity of the drilling operation (Abimbola, 2016; Oloruntobi, 2019). Pore pressure, which can result in kick, represents a critical safety factor. One of the early important articles in this regard was by Eaton, where methodologies to predict geopressure using sonic travel time, conductivity and corrected d-exponent in shale region were presented (Eaton, 1975). Accurate prediction of pore pressure (geopressure) facilitates the use of safe drilling fluid density (Brahma & Sircar, 2018; Y. Feng et al., 2015). There is the chance that pore pressure will exceed mud pressure during drilling, which can lead to kick. Detecting the occurrence of kick is important as uncontrolled kick can lead to blowout (Abimbola et al., 2014; Khakzad et al., 2013; Khoshnaw et al., 2014; Zhang & Yin, 2017). Downhole pressure sensors as well as high resolution flow meters offer useful means of monitoring for kick. Use of downhole sensors requires appropriate design for upward transmission of information. Also, cost, installation and maintenance represents challenges of using high resolution flow meter (Reitsma, 2011). This makes it a worthwhile to study how drilling parameters/surface sensors can be used for automatic kick detection.

Ensuring safety during drilling involves prediction/detection of events, e.g. kick occurrence, and taking appropriate control, e.g. controlling kick. This dissertation does not include drilling/well control for safety purposes; instead, the dissertation only focuses on prediction/detection. Several fault trees showing probability of blowouts occurring as a consequence of abnormal pore pressure and kick control can be found in the following: Tamim et al., (2019), Bijay et al., (2020) and Abimbola, (2016).

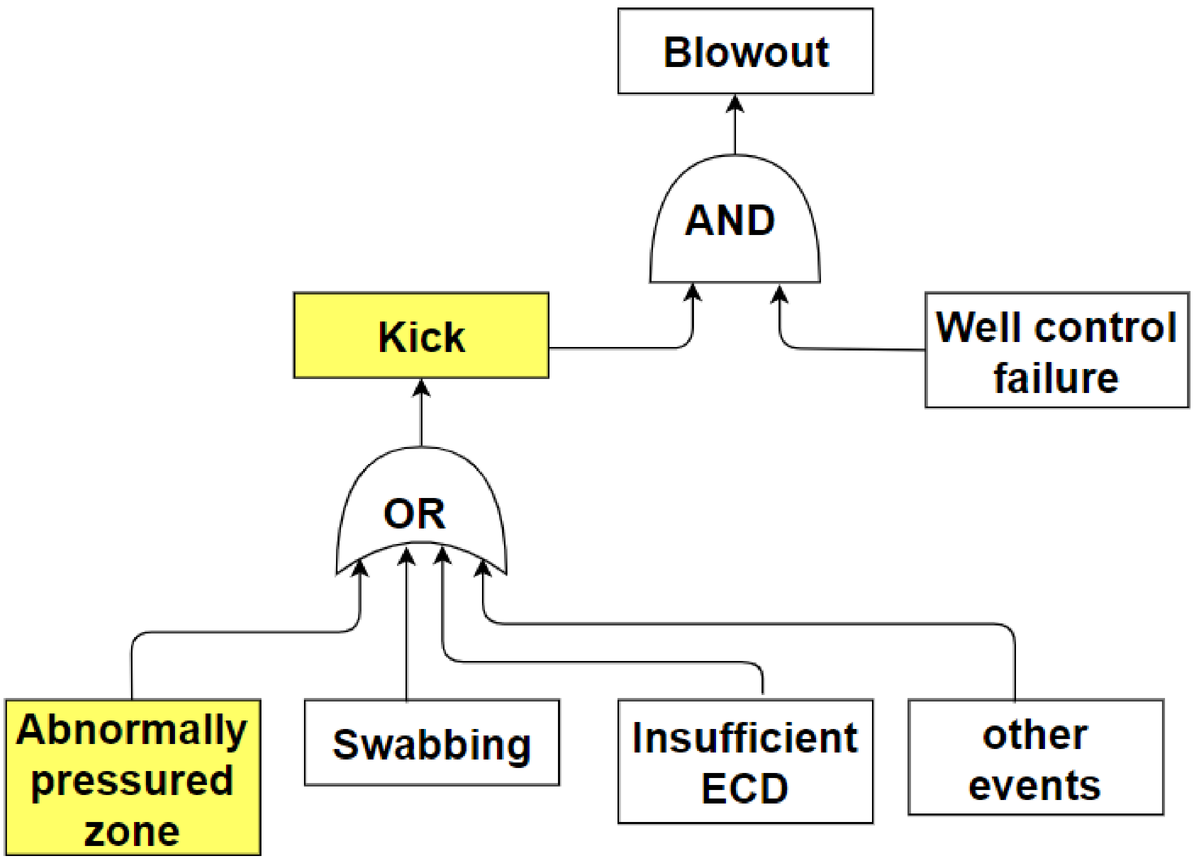

Fig. 1.1 Fault tree model for drilling operations.

Fig. 1.1 is obtained by slightly modifying part of Fig. 2 in the article by Abimbola et al., (2014). Fig. 1.1 is a fault tree which shows the events that can lead to blow out. In the same article by Abimbola et al., (2014), twenty different kick causing events (abnormally pressured zone, swabbing, insufficient equivalent circulating density (ECD), loss circulation, inadequate hole fill up, operator error, bad cementing, stuck pipe etc.,) and their probabilities were presented. Among these events, abnormally pressured zone had the highest probability. Fig. 1.2 shows the relative percentage of the top events that can cause kick based on their probabilities.

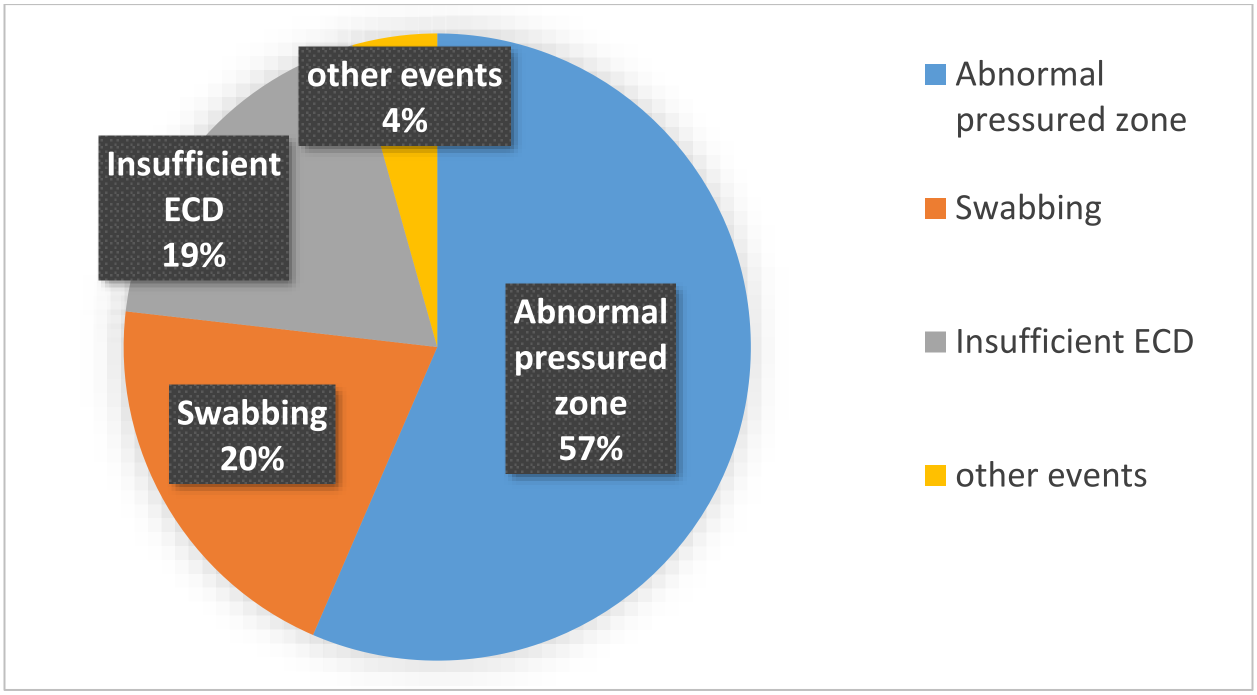

Fig. 1.2 Probability of kick causing events.

From Fig. 1.2, it can be seen that abnormally pressured zone with a relative percentage of 57% is majorly responsible for kick. This shows the benefit of accurately predicting pore pressure. The light-yellow coloured sections of Fig. 1.1 shows the focus of this dissertation towards enhancing drilling safety.

Predicting pore pressure (in order to identify abnormally pressured zone) also has its challenges. There are uncertainties in important variables which influence pore pressure prediction such as wireline log data, porosity and lithology (Oughton et al., 2018). Overpressure mechanisms can be due to compaction disequilibrium, fluid expansion or transfer mechanism (Tingay et al., 2009). Overpressures in sedimentary rocks are primarily caused by compaction disequilibrium (Tingay et al., 2009; Zhang, 2013); this is often observed as higher than the porosity expected at a given depth based on the normal porosity trend for the rock (Zhang, 2013). Compaction disequilibrium as a means of observing changes in pore pressure has been used in estimating pore pressure using several log measurements such as sonic log and resistivity log (Zhang, 2011). Fig. 1.3 was obtained by slightly modifying part of Figure 2 in the article by Oughton et al., (2018). Fig. 1.3 is a pore pressure-porosity parameter dependency network model based on mechanical compaction. The focus of Fig. 1.3 is on porosity. Other articles consulted for Fig. 1.3 are: Tingay et al., (2009), Maxwell, (1964) and Saleh et al., (2013).

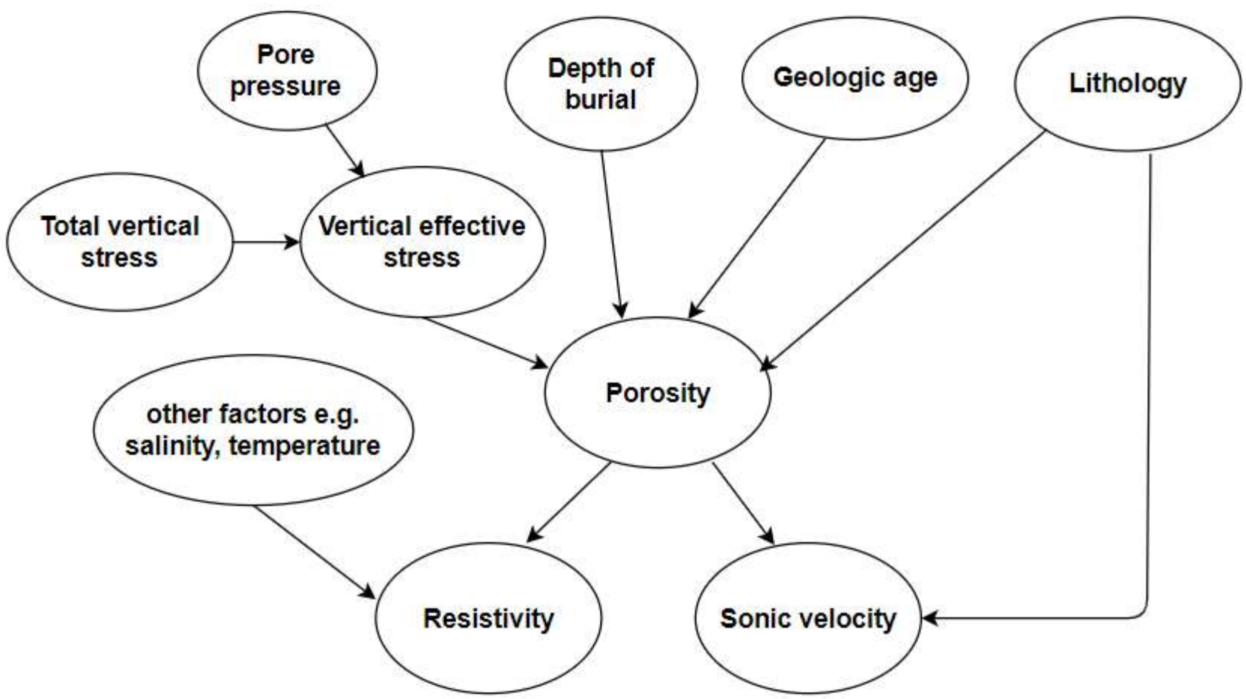

Fig. 1.3 Porosity dependency network model for pore pressure prediction

The arrows in Fig. 1.3 (also in Fig. 1.4) shows the direction of influence between parameters. For example, the porosity in a section of a rock will influence the resistivity and sonic velocity/transit time that can be measured in that section of the rock. As shown in Fig. 1.3, pore pressure variations can be captured by sonic and resistivity logs because these logs (sonic and resistivity) are sensitive to porosity anomaly caused by pore pressure variations (Tingay et al., 2009). Although porosity deviation from normal trend (compaction disequilibrium) represents the conventional means by which overpressure is estimated, Fig. 1.3 shows that lithology also influences porosity and improper lithology identification can cause erroneous pore pressure prediction from porosity deviation. Thus, proper lithology identification plays a significant role in achieving accurate pore pressure prediction. Based on this, this dissertation investigates both pore pressure prediction and lithology identification.

A parameter dependency network which utilizes parts of Fig. 1.3 and Fig. 1.1 is presented as Fig. 1.4. This figure shows the research focus of this dissertation.

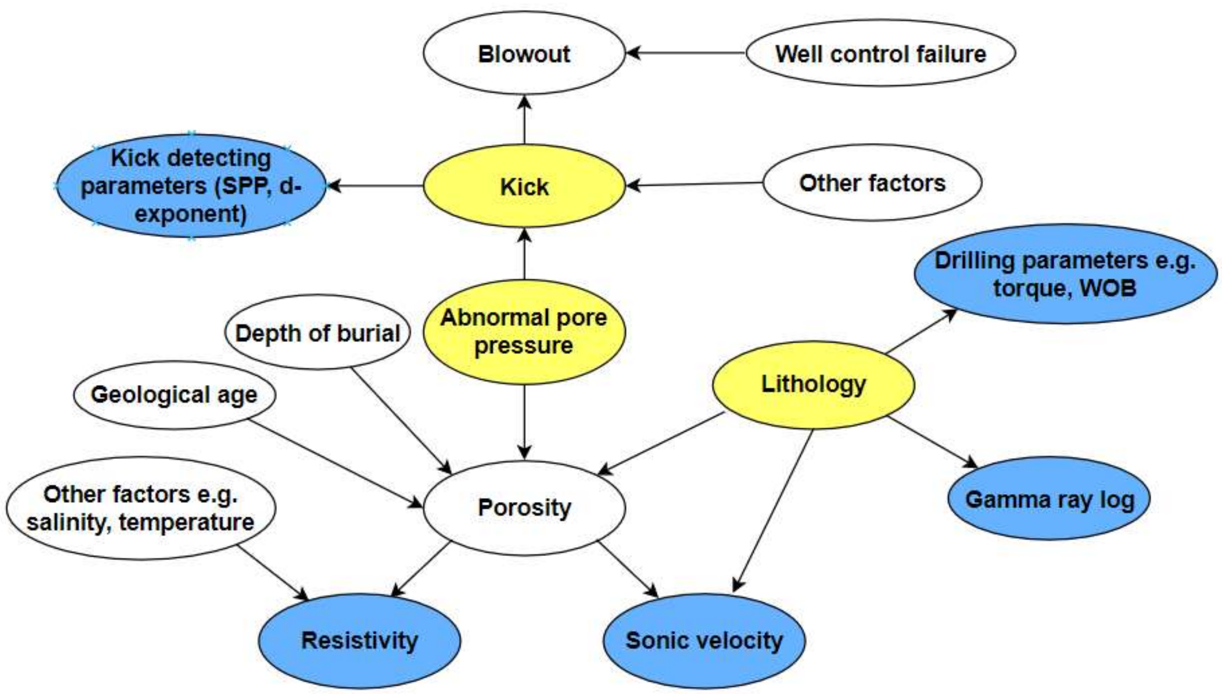

Fig. 1.4 Parameter dependency network of research focus

The light blue shaded sections of Fig. 1.4 show input parameters which would be used to predict/detect the light yellow shaded sections of Fig. 1.4. Lithology will be identified using drilling parameters and gamma ray log, pore pressure will be predicted using resistivity log and sonic derived porosity log, and kick detection will be done using Standpipe pressure (SPP) and d-exponent data. More details are available in Chapter 1.6 of this dissertation.

1.1.1 Brief introduction to machine learning

Humans possess the abilities to perform complex intelligent tasks for which there are no simple equations e.g. ability to observe resemblance among siblings. The field of machine learning is an evolving field which seeks to artificially recreate human intelligence. Machine learning therefore represents an interesting area of artificial intelligence which involves developing algorithms to learn from data (Freeman & Chio, 2018; Mohammed et al., 2017; Sze et al., 2017).

Machine learning can operate in supervised mode (if each of the training data are labelled as input and output) or unsupervised mode (data are not labelled as input or output). Machine learning is designed to perform the task of classification or regression (if the output data is discrete or continuous). Several machine learning algorithms have been developed such as simple neural networks, support vector machines, K nearest neighbour, learning decision tree, naïve Bayes, convolutional neural networks, etc. Machine learning algorithms differ in terms of their capability, ease of implementation and how easy it is to explain the outcome of their learning (Freeman & Chio, 2018; Hagan et al., 1996; Haykin, 2009; Mohammed et al., 2017; Russell & Norvig, 2010).

Deep learning is achieved by exploiting multiple levels of interrelationships among input parameters. This enables the interaction of input parameters at different hierarchical levels to be utilized for learning. Deep learning can be implemented by using a multiple hidden layer network. The multilayer can be structured in a form to enable hierarchical learning (Sze et al., 2017; Zhang et al., 2018B). Deep learning has achieved state of the art success in several domains such as image recognition and audio processing (Chung et al., 2014; Kumar et al., 2017; Sundermeyer et al., 2015). Although deep learning has achieved state of the art status in comparison to other forms of machine learning algorithms, the performance of deep learning would only surpass those of shallow neural networks, medium neural network and traditional machine learning when very high amount of data is used for training (Tang et al., 2018). The capability of deep learning to utilize hierarchical learning reduces the need for handcrafted engineering features. Deep learning is recommended when data is large, the system been modeled or data changes rapidly and there is lack of availability of human experts. (Alom et al., 2019; LeCun et al., 2015; Sze et al., 2017; Zhang et al., 2018B).

Deep learning requires a huge amount of data for efficient performance. Therefore, transfer learning and pre-training can be done to augment data availability as well as ease up on training time. In drilling engineering, pre-training offers the opportunity to utilize data from other petroleum fields for a field with limited data. Although deep learning offers the benefit of extracting relevant features of data for training, the capabilities of deep learning still hinges on the size of training data used.

Some questions that need to be addressed in the use of machine learning are: what machine learning algorithms are suitable for drilling operations, what are the current size of data being used in drilling engineering in comparison to other fields such as computer vision, how has the machine learning use progressed with respect to different machine learning algorithms, and what can be done to make the best use of machine learning for drilling data.

1.1.2 Kick during drilling and machine learning

Kick refers to the situation when formation fluid flows into the bottom hole as a result of the formation pressure exceeding the bottom hole pressure. While flow measurement indicators are good source of kick indication, false alarm represents a major challenge for flow measurement indicators. Several windowed threshold based algorithms, e.g. cumulative sum (CUSUM) algorithm, have been implemented to reduce false alarm rate (Hargreaves et al., 2001). In addition, more work has also been done to optimize kick detection (early detection with reduced false alarm) with the use of machine learning algorithms.

In order to optimally improve kick detection in noisy drilling data, Hargreaves et al., (2001) utilized the Bayesian probabilistic approach. The methodology involves developing models for inflow and outflow which represents different conditions including steady state, pipe movement, pumps on and kick. A Bayesian frame work is used to match the drilling data with the models and the probability outcomes can be used to indicate the event that the set of drilling data belongs to. Although testing showed some occurrences of false alarm, comparison with CUSUM algorithm showed improved sensitivity.

Nybo et al., (2008A) aimed at establishing a proof of concept that machine learning algorithms can be used to augment the performance of a physical model for false alarm reduction in kick detection. This approach combines an artificial intelligence (AI) approach called echo state network (ESN) with a physical model in order to reduce false alarm rate in kick prediction from fluid flow measurements using time series data. The physical model was an advanced dynamic flow model found in Petersen et al., (2006). The ESN which is a kind of artificial neural network with memory capabilities was chosen due to time dependent relationship in the data of the training parameters. For the combined AI and physical model, the ESN is used to predict a residual which will not be accounted for by the output from the physical model. The residual is obtained from the training data by subtracting physical model prediction of flow from actual measured flow. The ESN was trained to predict residual using pump rate and mud density. Experiment result shows that combining both physical and AI yielded a lower false alarm rate in comparison to stand alone physical model or standalone AI approach (Nybo et al., 2008A). Nybo et al., (2008B) also went ahead to show the benefit of using knowledge embedded in time series representation of drilling data for the purpose of mitigating false kick alarm (i.e., temporal relationship in data), as opposed to using basic threshold detection. For this purpose, two algorithms with data-memory capabilities (Auto Regressive Integrated Moving Average algorithm and the ESN) were used separately. For training, change in active volume is taken as output and pump rate is taken as input. The results of the use of these algorithms showed improvement in mitigating false Alarm.

Kamyab et al., (2010) utilized focused time-delay neural network (FTDNN) which is an algorithm with data-memory capabilities for kick detection. Data from 4 wells of 3 different fields were used for training, and the data was divided such that training data had 2 kicks and validation data also had two kicks. Several input parameters numbering 13 where used individually for predicting kick detection using the neural network. The input parameters considered were flow in, mud weight, total SPM (strokes per minute), RPM (revolutions per minute), torque, pump pressure, drag forces, weight on bit, hook load, rate of penetration, drilled depth, suction pit and active pits totalizer. Results show that the pit volume indicators (suction pit, active pits totalizer) were the most important for detection followed by pump pressure. In using the FTDNN, several factors are to be considered for optimal performance. These are: sampling frequency of the data, method used for data normalization and kick probability threshold value.

Conventional surface sensors could suffer from calibration drift. In order to improve the robustness of conventional sensors, Pournazari et al., (2015) utilized machine learning algorithms for pattern recognition. The machine learning implementation can also be used to augment the performance of a sensor which performs detection using physics-based calibration. Kick detection using the machine learning algorithms was based on trends in pit volume and flow out which occurs during drilling, pipe tripping and connections. The algorithms for training were intended for event classification and rig activity classification. The types of event include kick, lost circulation and fracture breathing. For drilling activity classification, time series data for bit speed, flow rate, pump stroke and pump stroke rate of change were used. In order to detect abnormal deviations in pit volume and flow-out signals, moving window averaging technique in combination to symbol aggregate approximation was employed. Machine learning algorithms can then be trained based on the patterns obtained from pit volume and flow out signals for event classification. Three different classifiers (Naïve Bayes, Decision tree and Random forest) were employed and the random forest classifier performed best (Pournazari et al., 2015).

In the article by Xie et al., (2018), wavelet neural network was utilized for kick prediction. Wavelet analysis offers several benefits such as denoising and analysis of time series data with non-stationary data at different frequencies. Genetic algorithm was used for optimizing the neural network. The following input data were used in training for kick detection: rate of penetration, mass per unit volume of a drilling fluid (synonymous with mud density), mud weight of circulating fluid, mud weight going into the well, mud weight going out of the well and mud depth. According to the article, the use of genetic algorithm made training faster with improved prediction. Fjetland et al., (2019) utilized long short-term memory recurrent neural network (LSTM-RNN) on simulated data using the following input parameters Flow rate in, flow rate out, standpipe pressure, choke pressure, choke opening, bit pressure and bit depth for kick detection.

It can be observed that input parameters such as flow rate, standpipe pressure, mud density, pit volume, rate of penetration, torque, weight on bit have been explored for kick detection using machine. Further work would be to get new measurable or derived parameter which can be used for kick detection in order to improve reliability of kick detection.

1.1.3 Pore pressure and drilling

For conventional drilling, it is desirable to keep mud weight gradient above pore pressure gradient in order to prevent kick. However, formation fracture can result when the mud weight gradient exceeds formation fracture gradient. It is therefore important to accurately predict pore pressure in order to design a safe drilling mud weight (density) window (Brahma & Sircar, 2018; Feng et al., 2015; Osarogiagbon et al., 2021).

Pore pressure prediction is carried out in shale lithology based on expected normal hydrostatic values and actual measured values of the following indicating parameters: resistivity, porosity, velocity, d-exponent, hydromechanical specific energy etc., (Eaton, 1975; Oloruntobi, 2019;

Zhang, 2011). Equation (1.1) represents the foundation from which pore pressure estimation is done using the indicating parameters (resistivity, porosity, velocity etc.,).

$$S_v = S_e + \alpha P. \tag{1.1}$$

In Equation (1.1), P refers to pore pressure, S_v refers to overburden stress, S_e refers to vertical effective stress and α refers to effective stress coefficient. α is usually assumed to be 1 (Zhang, 2013). However, for deep reservoir or reservoirs with high cementation, α may vary significantly from 1 (Amiri et al., 2019; Dassanayake et al., 2015; Mao et al., 2018; Sayers et al., 2002; Zhang, 2013). Core data have been obtained for which effective stress coefficient value as low as 0.55 was estimated (Civan, 2021).

Determining the value of α can be challenging because it can be influenced or modelled as a function of factors such as porosity, confining pressure, pore geometry, pore pressure, cementation, clay content etc., (Alam et al., 2012; Frempong & Butt, 2006; Luo et al., 2015; Xu et al., 2006).

In addition to the challenge of determining the value of α, changes in the indicating parameters may also be due to other factors other than pore pressure, e.g. resistivity may change due to changes in salt concentration of brine in the rock, temperature, fluid type, shale content, texture, and type of clay (Saleh et al., 2013). Based on this, several indicating parameters can be utilized to improve accuracy. The type of indicating parameters as well as the manner in which the indicating parameters can be efficiently combined can also be explored.

1.1.4 Lithology identification during drilling

Pore pressure prediction can be done using either velocity or resistivity or porosity values obtained in shale lithology (Eaton, 1975; Zhang, 2011). This shows the importance of appropriate lithology determination. Gamma ray log represents a typical means of detecting shale lithology in siliciclastic environment (Assaad, 2008; Clavier et al., 1971; Larionov, 1969; Olayiwola & Bamford, 2019; Oloruntobi & Butt, 2019; Oloruntobi, 2019; Stieber, 1970; Yusuf et al., 2019). Although gamma ray measurements can be obtained directly through logging while drilling (LWD), there are benefits of obtaining gamma ray measurements with the use of drilling parameters. Such benefits include economy (because drilling parameters are already part of standard parameters obtained while drilling) and improved reliability due to chances of LWD failing (Salehi et al., 2017; Zhang et al., 2018A). Several drilling parameters such as the rate of penetration (ROP) and d-exponent have been used for lithology identification. However, there is still the need to factor in the effect of other parameters such as bit type, bit wear, torque etc., (Oloruntobi & Butt, 2020). Based on this, it is important to explore means of capturing the relationship between gamma ray log and all relevant drilling parameters.

1.2 Research problem

In an attempt to improve overall safety during drilling, this research focuses on developing methodologies for predicting/detecting downhole conditions/events such as kick occurrence, pore pressure and lithology during drilling. In doing this, some principal research questions of concern are:

1. What are the regular data sizes available for use?

2. What input parameters should be considered?
3. What useful features/attributes and interrelationship among input parameters should be exploited?
4. How can deep learning be best used with input parameters or attributes of input parameters?

An attempt was made to answer these questions macroscopically (with a broad view) by carrying out a survey (Chapter 2). In Chapters 3-5, attempts will be made to provide more detailed solutions to some or all of these questions as pertaining to kick, pore pressure and lithology.

1.3 Research objective

The primary research objective is the application of deep learning in modeling downhole conditions during drilling for safety enhancement. In order to achieve this objective, attempt will be made to develop methodologies for extracting useful features/attributes from measuring/logging parameters obtained during drilling. Deep learning algorithms can then utilize the features/attributes for discrete/continuous event detection/prediction. The interrelationship among input parameters will also be explored by analysing the parameters. This is useful especially when the data size is not large enough for adequate deep learning implementation. Based on the primary objective, sub-objectives will involve modeling the following: Kick detection, pore pressure prediction and shale lithology identification.

1.4 Research scope

1. The method of learning utilized in this dissertation is the supervised machine learning mode.
2. For kick detection, the output event to be detected is qualitatively classified as kick occurred or no-kick occurred. Further work could be to estimate kick severity or hydrocarbon influx rate into the well during drilling using deep learning. The input parameters to be considered are those that can be measured near the surface (not near bit pressure sensors) such as rate of penetration, rotary speed, standpipe pressure etc.
3. For pore pressure prediction, resistivity log and porosity derived from sonic log will be utilized. Other possible means of deriving porosity log include neutron and density log (Kamel & Mabrouk, 2003). Porosity log developed from different sources (sonic, neutron etc.,) could significantly perform differently for pore pressure prediction (Tingay et al., 2009).
4. For shale lithology identification, gamma ray log generation using a methodology based on deep learning will be implemented in this dissertation. Other lithology identification logs such as photoelectric absorption log, self-potential log and neutron log (Ehsan & Gu, 2020; Fertl, 1987) could be utilized in future work.

1.5 Research tasks

1. Conduct a survey to understand the gaps in the use of supervised machine learning for pressure related safety/hazardous events during drilling as well as to observe the progress in the use of deep learning for drilling activities. This will provide information on the suitability of drilling data size currently used for deep learning, relevant input

parameters and what needs to be done to make the best use of deep learning with reference to detecting/predicting drilling related safety events.

2. Explore the use of deep learning for kick detection using relevant attributes of d-exponent data and standpipe pressure data.
3. Explore the interrelationship between porosity and resistivity for pore pressure prediction.
4. Explore the use of deep learning on gamma ray log generation using interrelationship between drilling parameters.

1.6 Dissertation structure

The methodologies developed in this dissertation can be included into conventional drilling approaches in order to improve safety conditions during drilling. Fig. 1.5. shows integration of the different aspects of this dissertation.

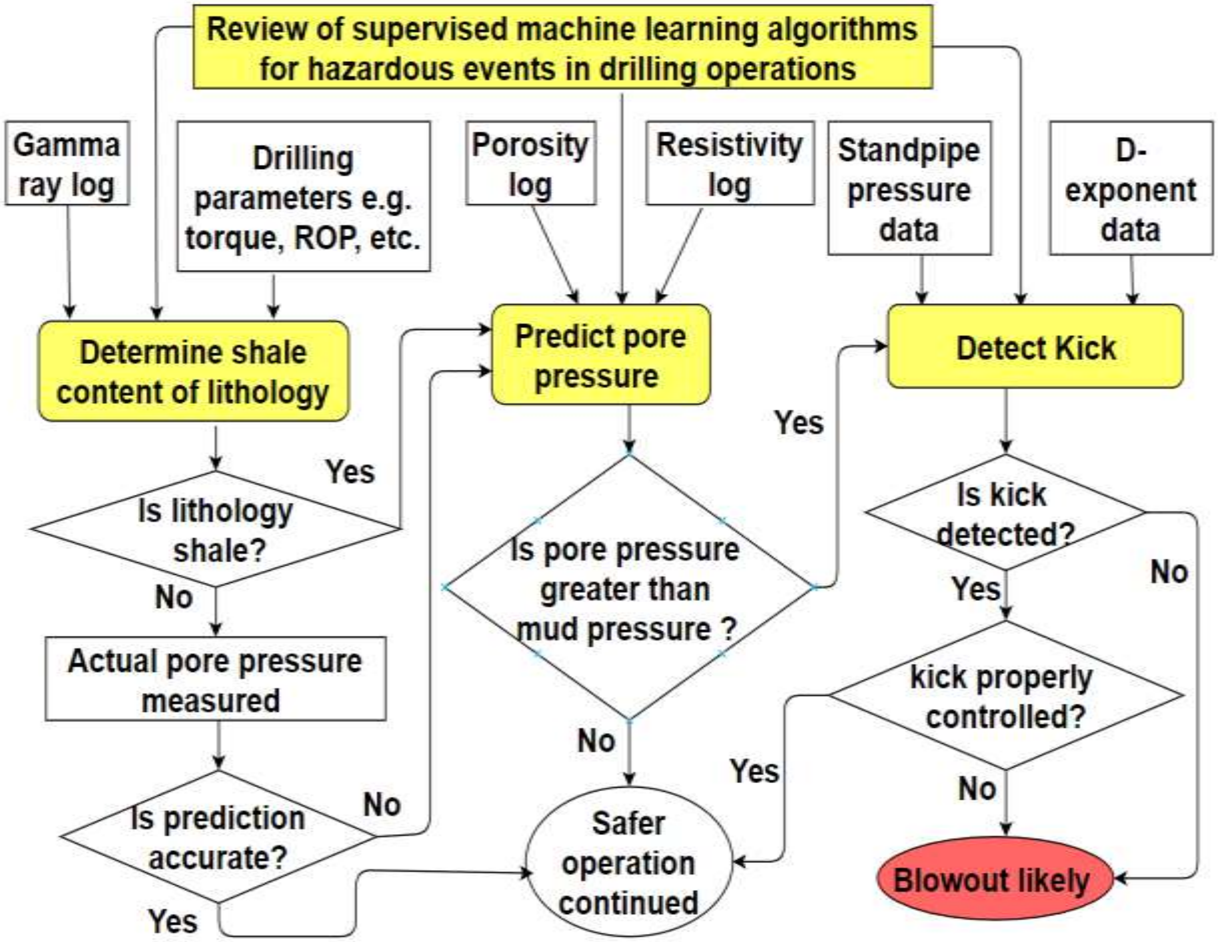

Fig. 1.5 Integrating the different research modules of this dissertation

In Fig. 1.5, the boxes with light yellow background show the contribution of this dissertation towards improving safety during drilling, the light red background show the event that the contribution of this dissertation aims to prevent. It should be noted that lithology determination, pore pressure prediction and kick detection (three of the four light yellow background in Fig. 1.5) are activities that are constantly being carried out during drilling because of their critical effect on safety. The dissertation is organized using manuscript style. Chapters 2 to 5 correspond to four journal articles which report the research described in this PhD dissertation. In addition, the Thesis conclusion is presented in Chapters 6. The structure of this dissertation is described in Table 1.1.

Table 1.1 Dissertation chapters and their titles

Chapter	**Title**
2	Review and analysis of supervised machine learning algorithms for hazardous events in drilling operations
3	A new methodology for kick detection during petroleum drilling using long short-term memory recurrent neural network
4	Combining porosity and resistivity log for pore pressure prediction
5	Gamma ray log generation from drilling parameters using deep learning
6	Thesis conclusion

Chapter 2 involved studying the use of supervised machine learning in a wide range of pressure based hazardous drilling events such as kick, kick, fracture, lost circulation and stuck pipe. Because the events are pressure based, the study also included the use of supervised machine learning on pore pressure, equivalent circulation density and bottom hole circulating pressure. By considering a wide number of events (e.g. kick, fracture, lost circulation etc.,) instead of only focusing on kick and pore pressure, a more robust conclusion on the use of machine learning can be obtained. Chapter 2 also involved studying deep learning and its use on several drilling activities such as lithology identification, drilling rig state determination, generating logging/other drilling parameters and detecting abnormality in data. This also served as a guide in developing the methodology for generating gamma ray log for lithology identification.

In Chapter 3, a survey was done to identify a new parameter which can be used for kick detection using a deep learning methodology and d-exponent was identified. The survey identified two groups of kick identification parameters which are: drilling parameter group (the same as d-exponent which is derived from weight on bit, rate of penetration, rotary speed and bit size) and flow parameter group (which utilizes flow in, flow out, standpipe pressure). It was observed that the flow paddles for measuring flow in and flow out rate can more easily become faulty. Hence only the Standpipe pressure was selected from the flow parameter group. A methodology which extracted relevant attributes from d-exponent and Standpipe pressure and trained these attributes to detect kick with the use of long short term recurrent neural network (a renowned deep learning algorithm) was developed and the methodology was tested with field data.

For pore pressure prediction, increase in porosity above the normal trend indicates higher pore pressure, this also corresponds directly to a proportional decrease in resistivity when Archie's cementation exponent is constant. However, a disproportionate change in resistivity when porosity changes, indicates changes in Archie's cementation exponent. Cementation represents a significant factor that can influence pore pressure prediction. In Chapter 4, the goal was to combine resistivity and porosity for pore pressure prediction by deriving an equation which utilizes porosity and resistivity in a way that captures changes in cementation effect. Field data was used for testing.

In Chapter 5, the goal was to able to generate gamma ray log (an established shale lithology identifier) from drilling parameters due to advantages of improved reliability at little to no cost and the ability to obtain data at bit point. The hydro-mechanical specific energy (HMSE) equation which incorporates several drilling parameters (torque, weight on bit, bit diameter, rotary speed, rate of penetration, flow rate, bit pressure drop at the nozzle and hydraulic energy reduction factor) will be presented as a robust means of capturing lithology changes with the use of drilling parameters. A methodology was developed which utilized HMSE and deep learning for gamma ray log generation. Several deep learning algorithms were explored in the methodology development in order to understand their suitability. Field data was used for testing.

1.7 Novelty and contributions

The main contributions of this PhD dissertation are:

1. A survey which points out the current progress and challenges in the use of supervised machine learning for pressure-based drilling hazard.
2. The development of a new methodology for kick detection which uses long short-term memory recurrent neural network on relevant features of standpipe pressure and d-exponent data for kick detection.
3. The development of a methodology which combines porosity and resistivity for pore pressure prediction by accounting for changes in cementation effect.
4. The development of a methodology for generating gamma ray log from drilling parameters using deep learning for lithology identification.

1.8 Conclusions

The purpose of this dissertation is to improve safety of drilling operation by focusing on pore pressure-based hazards. The dissertation focuses on developing methodologies for detecting/predicting downhole conditions, the results of which can serve as input to drilling decisions for the overall safety of operations.

This chapter introduces the challenges and approach adopted in order to achieve the purpose of this dissertation. The overall structure of the dissertation is also presented in this chapter.

1.9 Nomenclature

α	Effective stress coefficient
P	Actual pore pressure to be predicted (psi)
S_e	Vertical effective stress (psi)
S_v	Overburden stress (psi)

1.10 Acronyms

CUSUM	cumulative sum
ECD	equivalent circulating density
ESN	echo state network
FTDNN	focused time-delay neural network
HMSE	hydro-mechanical specific energy
LSTM	long short-term memory
LWD	logging while drilling
ROP	rate of penetration

RPM	revolutions per minute
RNN	recurrent neural network
SPP	standpipe pressure
SPM	strokes per minute

1.11 References

Abimbola, M., Khan, F., & Khakzad, N. (2014). Dynamic safety risk analysis of offshore drilling. Journal of Loss Prevention in the Process Industries, 30, 74–85.

Abimbola, M. O. (2016). Dynamic safety analysis of managed pressure drilling operations. Memorial University of Newfoundland.

Alam, M. M., Fabricius, I. L., & Christensen, H. F. (2012). Static and dynamic effective stress coefficient of chalk: Geophysics, 77. L1–L11.

Allahvirdizadeh, P. (2020). A review on geothermal wells: Well integrity issues. Journal of Cleaner Production, 124009.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. Electronics, 8(3), 292.

Amiri, M., Lashkaripour, G. R., Ghabezloo, S., Moghaddas, N. H., & Tajareh, M. H. (2019). 3D spatial model of Biot's effective stress coefficient using well logs, laboratory experiments, and geostatistical method in the Gachsaran oil field, southwest of Iran. Bulletin of Engineering Geology and the Environment, 78(6), 4633–4646.

Assaad, F. A. (2008). Field methods for petroleum geologists: A guide to computerized lithostratigraphic correlation charts case study: Northern Africa. Springer Science & Business Media.

Bijay, B., George, P., Renjith, V. R., & Kurian, A. J. (2020). Application of dynamic risk analysis in offshore drilling processes. Journal of Loss Prevention in the Process Industries, 68, 104326.

Brahma, J., & Sircar, A. (2018). Design of safe well on the top of Atharamura anticline, Tripura, India, on the basis of predicted pore pressure from seismic velocity data. Journal of Petroleum Exploration and Production Technology, 8(4), 1209–1224.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv Preprint ArXiv:1412.3555.

Civan, F. (2021). Effective-Stress Coefficients of Porous Rocks Involving Shocks and Loading/Unloading Hysteresis. SPE Journal, 26(01), 44–67.

Clavier, C., Hoyle, W., & Meunier, D. (1971). Quantitative interpretation of thermal neutron decay time logs: part I. Fundamentals and techniques. Journal of Petroleum Technology, 23(06), 743–755.

Dassanayake, A. B. N., Fujii, Y., Fukuda, D., & Kodama, J. (2015). A new approach to evaluate effective stress coefficient for strength in Kimachi sandstone. Journal of Petroleum Science and Engineering, 131, 70–79.

Eaton, B. A. (1975). The equation for geopressure prediction from well logs. Fall Meeting of the Society of Petroleum Engineers of AIME.

Ehsan, M., & Gu, H. (2020). An integrated approach for the identification of lithofacies and clay mineralogy through Neuro-Fuzzy, cross plot, and statistical analyses, from well log data. Journal of Earth System Science, 129(1), 1–13.

Feng, Y., Arlanoglu, C., Podnos, E., Becker, E., & Gray, K. E. (2015). Finite-element studies of hoop-stress enhancement for wellbore strengthening. SPE Drilling & Completion, 30(01), 38–51.

Fertl, W. H. (1987). Log-derived evaluation of shaly clastic reservoirs. Journal of Petroleum Technology, 39(02), 175–194.

Fjetland, A. K., Zhou, J., Abeyrathna, D., & Gravdal, J. E. (2019). Kick detection and influx size estimation during offshore drilling operations using deep learning. 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2321–2326.

Freeman, D., & Chio, C. (2018). Machine learning and Security. O'Reilly Media: Newton, MA, USA.

Frempong, P. K., & Butt, S. D. (2006). Non-destructive techniques to determine the effective stress coefficient of sandstone formations. Journal of Canadian Petroleum Technology, 45(07).

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). Neural network design. PWS Publishing Co.

Hargreaves, D., Jardine, S., & Jeffryes, B. (2001). Early kick detection for deepwater drilling: New probabilistic methods applied in the field. SPE Annual Technical Conference and Exhibition.

Haykin, S. (2009). Neural networks and learning machines,–3rd ed., Copyright by Pearson Education. Inc., Upper Saddle River, New Jersey, 7458.

Kamel, M. H., & Mabrouk, W. M. (2003). Estimation of shale volume using a combination of the three porosity logs. Journal of Petroleum Science and Engineering, 40(3–4), 145–157.

Kamyab, M., Shadizadeh, S. R., Jazayeri-rad, H., & Dinarvand, N. (2010). Early kick detection using real time data analysis with dynamic neural network: A case study in iranian oil fields. Nigeria Annual International Conference and Exhibition.

Khakzad, N., Khan, F., & Amyotte, P. (2013). Quantitative risk analysis of offshore drilling operations: A Bayesian approach. Safety Science, 57, 108–117.

Khoshnaw, F., Jaf, P., & Farkha, S. (2014). Pore, abnormal formation and fracture pressure prediction. WIT Transactions on Ecology and the Environment, 186, 579–593.

Kumar, A., Kim, J., Lyndon, D., Fulham, M., & Feng, D. (2017). An ensemble of fine-tuned convolutional neural networks for medical image classification. IEEE Journal of Biomedical and Health Informatics, 21(1), 31–40.

Larionov, V. V. (1969). Radiometry of boreholes. Nedra, Moscow, 127.

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

Lukawski, M. Z., Anderson, B. J., Augustine, C., Capuano Jr, L. E., Beckers, K. F., Livesay, B., & Tester, J. W. (2014). Cost analysis of oil, gas, and geothermal well drilling. Journal of Petroleum Science and Engineering, 118, 1–14.

Luo, X., Were, P., Liu, J., & Hou, Z. (2015). Estimation of Biot's effective stress coefficient from well logs. Environmental Earth Sciences, 73(11), 7019–7028.

Mao, X., Liu, Y., Guan, W., Liu, S., & Li, J. (2018). A new effective stress constitutive equation for cemented porous media based on experiment and derivation. Arabian Journal of Geosciences, 11(13), 337.

Maxwell, J. C. (1964). Influence of depth, temperature, and geologic age on porosity of quartzose sandstone. AAPG Bulletin, 48(5), 697–709.

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2017). Machine learning: algorithms and applications. Crc Press.

Nybo, R., Bjorkevoll, K. S., & Rommetveit, R. (2008). Spotting a false alarm. Integrating experience and real-time analysis with artificial intelligence. Intelligent Energy Conference and Exhibition.

Nybo, R., Bjorkevoll, K. S., Rommetveit, R., Skalle, P., & Herbert, M. C. (2008). Improved And Robust Drilling Simulators Using Past Real-Time Measurements And Artificial Intelligence. Europec/EAGE Conference and Exhibition.

Olayiwola, A. M., & Bamford, K. M. (2019). Depositional environment and reservoir characterization of the deep offshore Upper Miocene to Early Pliocene Agbada Formation, Niger delta, Nigeria. Journal of African Earth Sciences, 159, 103578.

Oloruntobi, O., & Butt, S. (2019). The New Formation Bulk Density Predictions for Siliciclastic Rocks. Journal of Petroleum Science and Engineering.

Oloruntobi, O., & Butt, S. (2020). Application of specific energy for lithology identification. Journal of Petroleum Science and Engineering, 184, 106402.

Oloruntobi, O. S. (2019). The pore pressure, bulk density and lithology prediction (Doctoral dissertation, Memorial University of Newfoundland). Memorial University of Newfoundland.

Osarogiagbon, A. U., Khan, F., Venkatesan, R., & Gillard, P. (2021). Review and analysis of supervised machine learning algorithms for hazardous events in drilling operations. Process Safety and Environmental Protection, 147. https://doi.org/10.1016/j.psep.2020.09.038

Oughton, R. H., Wooff, D. A., Hobbs, R. W., Swarbrick, R. E., & O'Connor, S. A. (2018). A sequential dynamic Bayesian network for pore-pressure estimation with uncertainty quantification. Geophysics, 83(2), D27–D39.

Petersen, J., Bjørkevoll, K. S., Frøyen, J., & Rommetveit, R. (2006). A general dynamic model for flow related operations during drilling, completion, well control and intervention. IBP1373_06, Presented at the Rio Oil & Gas Expo and Conference, 11–14.

Pournazari, P., Ashok, P., van Oort, E., Unrau, S., & Lai, S. (2015). Enhanced kick detection with low-cost rig sensors through automated pattern recognition and real-time sensor calibration. SPE Middle East Intelligent Oil and Gas Conference and Exhibition.

Reitsma, D. (2011). Development of an automated system for the rapid detection of drilling anomalies using standpipe and discharge pressure. SPE/IADC Drilling Conference and Exhibition.

Russell, S. J., & Norvig, P. (2010). Artificial Intelligence-A Modern Approach, Third International Edition. Pearson Education London.

Saleh, S., Williams, K. E., & Rizvi, A. (2013). Resistivity-Based Pore-Pressure Prediction-Pitfalls and Solutions. SPE Annual Technical Conference and Exhibition.

Salehi, M. M., Rahmati, M., Karimnezhad, M., & Omidvar, P. (2017). Estimation of the non records logs from existing logs using artificial neural networks. Egyptian Journal of Petroleum, 26(4), 957–968.

Sayers, C. M., Johnson, G. M., & Denyer, G. (2002). Predrill pore-pressure prediction using seismic data. Geophysics, 67(4), 1286–1292.

Stieber, S. J. (1970). Pulsed neutron capture log evaluation-Louisiana Gulf Coast. Fall Meeting of the Society of Petroleum Engineers of AIME.

Sundermeyer, M., Ney, H., & Schlüter, R. (2015). From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(3), 517–529.

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295–2329.

Tamim, N., Laboureur, D. M., Hasan, A. R., & Mannan, M. S. (2019). Developing leading indicators-based decision support algorithms and probabilistic models using Bayesian network to predict kicks while drilling. Process Safety and Environmental Protection, 121, 239–246.

Tang, A., Tam, R., Cadrin-Chênevert, A., Guest, W., Chong, J., Barfett, J., Chepelev, L., Cairns, R., Mitchell, J. R., & Cicero, M. D. (2018). Canadian Association of Radiologists white paper on artificial intelligence in radiology. Canadian Association of Radiologists Journal, 69(2), 120–135.

Tingay, M. R. P., Hillis, R. R., Swarbrick, R. E., Morley, C. K., & Damit, A. R. (2009). Origin of overpressure and pore-pressure prediction in the Baram province, Brunei. Aapg Bulletin, 93(1), 51–74.

Xie, H., Shanmugam, A. K., & Issa, R. R. A. (2018). Big Data Analysis for Monitoring of Kick Formation in Complex Underwater Drilling Projects. Journal of Computing in Civil Engineering, 32(5), 4018030.

Xu, X., Hofmann, R., Batzle, M., & Tshering, T. (2006). Influence of pore pressure on velocity in low-porosity sandstone: Implications for time-lapse feasibility and pore-pressure study. Geophysical Prospecting, 54(5), 565–573.

Yusuf, B., Oloruntobi, O., & Butt, S. (2019). The formation bulk density prediction for intact and fractured siliciclastic rocks. Geodesy and Geodynamics. https://doi.org/https://doi.org/10.1016/j.geog.2019.05.005

Zhang, D., Chen, Y., & Meng, J. (2018). Synthetic well logs generation via Recurrent Neural Networks. Petroleum Exploration and Development, 45(4), 629–639.

Zhang, J. (2011). Pore pressure prediction from well logs: Methods, modifications, and new approaches. Earth-Science Reviews, 108(1–2), 50–63.

Zhang, J. (2013). Effective stress, porosity, velocity and abnormal pore pressure prediction accounting for compaction disequilibrium and unloading. Marine and Petroleum Geology, 45, 2–11.

Zhang, J., & Yin, S.-X. (2017). Fracture gradient prediction: an overview and an improved method. Petroleum Science, 14(4), 720–730.

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. Information Fusion, 42, 146–157.

Chapter 2

2. Review and analysis of supervised machine learning algorithms for hazardous events in drilling operations

Preamble

In line with the primary objective of this dissertation as presented in section 1.3 of Chapter 1, this chapter provides a review of applications of machine learning methods in the study of hazardous conditions during drilling, as well as special focus on the use of deep learning in drilling related activities. This study provides insight into the progress and challenges in the use of machine learning methods. The knowledge obtained from this chapter aided in the development of the methodologies of the other chapters of this dissertation. Such aid include choice of machine learning algorithms, manner in which machine learning algorithms should be used and some possible consequences of the machine learning algorithms used.

I (Augustine Uhunoma Osarogiagbon) have contributed to Conceptualization, Methodology, Formal Analysis, Software, Investigation, Writing - Original Draft, and Writing - Review & Editing of this work, while Dr. Ramachandran Venkatesan contributed to Methodology, Formal Analysis, Writing - Review & Editing, Supervision, and Project Administration; Dr. Faisal Khan contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing, Supervision, and Project Administration; and Dr. Paul Gillard contributed to Formal Analysis, Writing - Review & Editing, and Supervision. A version of this chapter is published in the Journal of Process Safety and Environmental Protection, Volume 147, March 2021, Pages 367-384, https://doi.org/10.1016/j.psep.2020.09.038.

Abstract

Results of bibliometric analysis and a detailed review are reported on the use of supervised machine learning to study hazardous drilling events. The bibliometric analysis attempts to answer pertinent questions related to progress in the use of supervised machine learning for hazardous events due to drilling fluid density/mud weight. The analysis indicates artificial neural network as the most popular algorithm among researchers. Also, deep learning, random forest and support vector machine have gained momentum in recent use.

A critical review of literature on hazardous events and supervised machine learning algorithms is presented. This review was done to observe how the algorithms were used, their relative successes, limitations, as well as input parameters which aided in detection or estimation by the machine learning algorithms. An introduction to deep learning and a review of literature on the use of deep learning with respect to operations involving drilling parameters is presented. The review on deep learning and drilling parameters covered the following operations: lithology identification, drilling rig state determination, generating logging/other drilling parameters and detecting abnormality in data.

The study highlights need of publicly accessible large database with data from different oilfields for development of machine learning algorithms. These algorithms could be used globally for the enhancement of machine learning for new fields or fields with limited data. The availability of such large database would aid researchers in improving or customizing deep learning algorithms in line with the unique needs of drilling activities.

Keywords: Machine learning; artificial intelligence; deep learning; bibliometric analysis; drilling operation; drilling safety; petroleum industry

2.1 Introduction

The activities of the oil and gas industry can pertain to upstream, midstream or downstream. Upstream activities involve reservoir characterization, drilling and production of crude products. Midstream mainly involves processing, storage, marketing and transportation of the output from upstream. Downstream activities include receiving outputs from the midstream, refining the oil and performing distribution of petroleum products (PSAC, 2018). Several challenges encountered in the oil and gas industry can benefit from the use of machine learning. For example, in the area of drilling, machine learning can be applied towards pore pressure prediction (Ahmed et al., 2019a), in reservoir characterization, machine learning can be used in predicting reservoir properties at locations without core or appropriate log data (Osarogiagbon et al., 2015) and machine learning can also be used to forecast oil production rate (Mamudu et al., 2020).

Hazardous events are undesirable as they can lead to loss of time, loss of money, loss of human abilities and loss of lives. Sadly, the oil and gas industry has had its share of disastrous accidents which can be traced to certain events. A popular case is the Macondo blowout which resulted in an estimated loss of over 14 billion dollars (Mason, 2019). During drilling, hazardous events which are directly caused or highly influenced by the use of wrong drilling fluid density includes kick, formation fracture, lost circulation and stuck pipe (Abimbola et al., 2015). If these events are not properly monitored and controlled, they could lead to accidents. Predicting the occurrence of these events can be challenging due to the number of influencing parameters. Kick can occur due to several classes of factors related to hydrostatic head (e.g. abnormal pore pressure, insufficient mud density, lost circulation), cement (e.g. inadequate bonding, casing centralization), or pressure control equipment for managed pressure drilling (Tamim et al., 2019). This shows that robust mathematical models might be required to predict or detect kick as a function of causing factors.

The impact of hazardous events on oil and gas operations cannot be over-emphasized. Occurrence of blowouts or shutting/relieving a well to prevent blowout represents huge losses to the oil and gas industry. Thus, there are several publications which present methodologies for early detection, prediction, and mitigation of hazardous events in the oil and gas industry. Some studies showing methodologies for analysis and detection of drilling related hazards can be found in (Abimbola et al., 2015; Sun et al., 2018a). This work aims to summarize the efforts of authors in using supervised machine learning in the area of drilling with regards to hazardous events. This can thus provide pointers to where there are obvious rooms for improvement. Machine learning has attracted considerable interest in the oil and gas industry and Table 1 lists selective review papers on the use of machine learning for petroleum exploration/production applications.

Table 2.1 Selective reviews on the use of machine learning in oil and gas operations.

Source	Focus of articles
(Bravo et al., 2014)	Popularity/acceptability of artificial intelligence among different segments of oil and gas workers.
(Agwu et al., 2018)	Literature review on artificial intelligence and drilling fluid.
(Alkinani et al., 2019a)	The use of artificial neural network in different categories (exploration, drilling, production, and reservoir engineering) of oil and gas operations.

(Noshi & Schubert, 2018)	Descriptions of commonly used machine learning techniques/algorithms in the drilling industry with the aim of exposing the merits and demerits of these algorithms for drilling applications.
(Bello et al., 2015)	Review of artificial intelligence in drilling.

Although several literature reviews on the use of machine learning or artificial intelligence for petroleum applications have been reported, to the best of our knowledge, (i) no bibliometric analysis on trend in usage of supervised machine learning and hazard events in drilling related activities of the petroleum industry is available, (ii) no general review on the use of deep learning on drilling parameters have been done. In addition, this work aims to present the limitations in the current usage of supervised machine learning for hazardous events with respect to drilling.

This work is structured as follows: Section 2.2 gives an introduction to machine learning, Section 2.3 gives an introduction to deep learning, Section 2.4 describes factors that can influence the use of machine learning, Section 2.5 describes bibliometric analysis used to obtain trend in use of machine learning algorithms, Section 2.6 describes results of bibliometric analysis, Section 2.7 presents review of articles on machine learning and hazardous events in drilling engineering, Section 2.8 presents review of articles on deep learning and drilling parameters, Section 2.9 presents gaps in machine learning implementations, and conclusion is given in Section 2.10.

2.2 Introduction to machine learning

It is very convenient to have a simple and well-defined equation which solves a problem. However, it is often difficult to do this for many real-life problems. For example, while it is easy for a person to identify members of a family from their looks, there is no simple equation for this (Russell & Norvig, 2010). These forms of challenges are a key reason why computation is moving in the direction of simulating how human reason. Although machine learning and artificial intelligence are used interchangeably, machine learning is a subset of artificial intelligence. Artificial intelligence deals with the ability of computers/machines to take decisions and act like humans. However, some of these decisions may not require learning, e.g. the door of a house opening when an object is close to it may be termed artificial intelligence, but this could have been done by using a motor that is activated whenever the path between a photo detector and a light emitting diode is obstructed. On the other hand, machine learning refers to the situation where the door control system has to go through some training example in order to realize the appropriate event required for the door to open. In the field of drilling, an example of the application of machine learning is when a kick detection system learns the values of parameters such as standpipe pressure, torque, etc. that indicates the occurrence of kick after feeding the kick detection system with some sets of data. Therefore, machine learning involves the use of algorithms or procedures by an artificial system to learn from data (Freeman & Chio, 2018; Mohammed et al., 2017; Sze et al., 2017).

Machine learning can be classified into supervised learning, unsupervised learning and reinforcement learning. For supervised learning, a system is given some inputs and their corresponding output or targets; the system then tries to build a relationship between the inputs and outputs. The goal of the relationship built through supervised learning is to predict the outputs for a set of inputs that were not used during learning. Unsupervised learning refers to the case whereby a system is given a set of data without any specification (input or output); the system then tries to discover any possible form of relationship among this set of data. Reinforcement learning refers to learning based on experience, i.e. a system interacts with an environment and learns based on the consequence of its actions. An example of reinforcement learning application is a robot learning how to adjust mud weight (mud density) during drilling through simulations. The robot could use the outcome of its choices in deciding the right and wrong values of mud weight to use at different drilling scenarios. There is also semi-supervised learning, which results when some of the data are labeled as in the case of supervised learning and some of the data fall under the category of unsupervised learning (Mohammed et al., 2017). For the purpose of this dissertation, the focus will be on supervised learning. In addition to categorizing machine learning based on learning mode (supervised, unsupervised and reinforcement learning), machine learning can be categorized as performing the task of classification or regression based on the qualitative or quantitative nature of output data. Machine learning performs the task of classification when the output is qualitative in nature. Likewise, machine learning performs regression when the output is quantitative in nature (James et al., 2013). For example, if the machine learning task is to detect the downhole drilling condition at a given point in time if kick occurred or loss circulation occurred or if the downhole is at a balanced drilling condition, then such task is termed classification. Also, when the goal of a machine learning algorithm is to estimate overbalance pressure value or quantity of gas influx per seconds, then such task is termed regression.

Several supervised machine learning algorithms can be found in literature. Some common once are: classification/decision tree, random forest, k nearest neighbor, support vector machine, artificial neural network, naïve Bayes, linear regression and linear discriminant analysis (Freeman & Chio, 2018; Mohammed et al., 2017; Russell & Norvig, 2010). While some machine learning algorithms may be more robust than others in terms of learning accuracy, other factors such as speed of implementation, ease of interpreting results, nature of supervised machine learning task (classification or regression), nature and amount of input and output data, as well as the complexity of the model to be learnt, are important to consider when deciding the machine learning algorithm to employ. For example, linear regression offers the advantage of easy implementation and ease of result interpretation in comparison to artificial neural network. However, linear regression will not perform as good as artificial neural network when the relationship between input and output to be learnt is not linear. More information on machine learning can be found in (Freeman & Chio, 2018; Hagan et al., 1996; Haykin, 2009; Mohammed et al., 2017; Russell & Norvig, 2010).

2.3 Introduction to deep learning

Deep learning results from the application of multi hidden layer neural network for learning (Sze et al., 2017) (Zhang et al., 2018b). The beauty of deep learning is that it works by exploiting several levels of interrelationships among input parameters. With this, deep learning can learn important features of the input parameters at different hierarchical levels of input data interaction. Deep learning has been successfully applied to several domains such as image recognition and audio processing. Deep learning represents the future of machine learning (Pouyanfar et al., 2018), not only because the founding principle is based on exploiting how humans and animal learn (Sze et al., 2017), but its performance is outstanding in comparison to other machine learning algorithms when sufficient data is available (Feng et al., 2019; Zhang et al., 2018b). For example, convolution neural network is a form of deep learning algorithm and it represents the state of art for image classification (Kumar et al., 2017). According to the article by Arulkumaran *et al.,* the application of deep learning in the field of reinforcement learning has taken reinforcement learning to heights that were previously unattainable (Arulkumaran et al., 2017). Due to this, deep reinforcement, which is the outcome of the use of deep learning for reinforcement learning is poised towards revolutionizing the world of artificial intelligence (Arulkumaran et al., 2017).

The major motivation for the need for deep learning can be understood by studying Fig. 2.1.

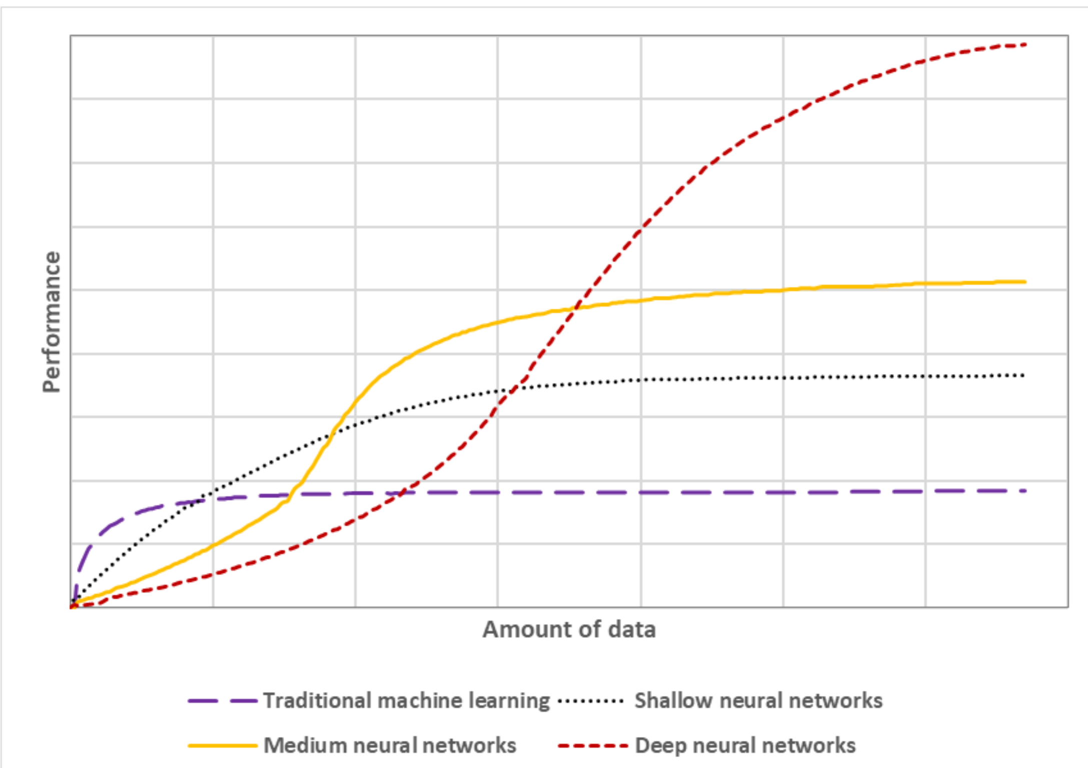

Fig. 2.1 Performance as a function of data availability for different machine learning algorithms

Fig. 2.1 is an approximate reconstruction of machine learning performance as a function of data size shown in (Tang et al., 2018). Fig. 2.1 shows that deep learning becomes more successful than other forms of machine learning when very large amount of data is used for training.

However, for very small data, deep learning is less efficient compared to other machine learning algorithms. Similar performance plot of deep learning in comparison to other machine learning algorithms as a function of amount of data can be found in (Alom et al., 2019; Alyafeai & Ghouti, 2020).

The ability of deep learning to capture hierarchal representation of information in the training data during the course of training reduces the need to use handcrafted engineering features which is common in traditional machine learning approaches. Deep learning is recommended when data is extremely large, data and system behavior changes rapidly and there is significant limitation in human expert/analytical means of solving the task. Based on this, task such as weather forecasting, image and speech recognition provides the avenue for which deep learning can be judiciously explored (Alom et al., 2019; LeCun et al., 2015; Sze et al., 2017; Zhang et al., 2018b). The main draw back with the use of large data for which deep learning is suited for is high computation resources requirement. Thus, deep learning users need to plan for time required for training, memory storage of data, power/energy requirement and cost. Reviews have also been carried out on challenges and progress in hardware and software platforms for deep learning implementation (Alom et al., 2019; Hatcher & Yu, 2018; Sze et al., 2017). It is worth noting that tech giants also offer services (Amazon Web Services, Google Cloud, Azure, Alibaba cloud, Baidu cloud etc.,) to aid with the use of deep learning (Li et al., 2019).

The area of deep learning of recent is gaining increased interest from researchers and based on this, it can be inferred that the applications of deep learning is bound to increase at least in the nearest future. Several reviews have been carried out on deep learning. These reviews focused on several aspects of deep learning including algorithms, fields of applications, challenges and future use. Table 2.2 show some of these review articles and some of the deep learning networks which the articles focused on (indicated by the ✓ sign). The acronyms of terms used in Table 2.2 are: Convolutional Neural Network (CNN), Deep Belief Network (DBN), Restricted Boltzmann Machine/Deep Boltzmann Machine (BM), Recurrent Neural Network (RNN), Recursive Neural Network (ReNN), Generative Adversarial Network (GAN), Auto-Encoder (AE), Stacked Auto-Encoder (SAE), Variational Auto-Encoder (VAE).

Table 2.2 Some general references on deep learning algorithms and applications

Deep learning network	Authors				Some notable aim/features of the deep learning network
	(Pouyanfar et al., 2018)	(Zhang et al., 2018b)	(Mohammadi et al., 2018)	(W. Liu et al., 2017)	
CNN	✓	✓	✓	✓	The architecture of CNN was inspired by the visual cortex organization in animals, which utilizes spatial locality or relationship in image data. Utilizing this form of structure helps to reduce computation cost (W. Liu et al., 2017). CNN has been be used for image and video recognition (Hatcher & Yu, 2018).
BM	✓	✓	✓	✓	The network aims to learn how to reproduce desired or acceptable values of a vector of input parameters connected in the visible layer with high probability (Salakhutdinov & Hinton, 2012; Zhang et al., 2018b). The BM can be used for reduction of data dimensionality (Mohammadi et al., 2018), collaborative filtering (Du et al., 2017) and event classification (Sharan & Moir, 2017).
DBN	✓	✓	✓	✓	The DBN is identical to the BM except that each layer of the DBN can be trained greedily. Thus the connections between two layers in the DBN can be made to be directed, whereas the connections between two layers in BM are undirected. This improves the network's interrelationship based on the trained data for DBN in comparison to BM; however, this could make the DBN

					less robust to new sets of data (Pouyanfar et al., 2018).
RNN	✓	✓	✓		RNN is Built to exploit sequential relationship in data, e.g. previous inputs in time or space could influence current output. Therefore, RNN requires memory for storage of past inputs or computed values (Zhang et al., 2018a). RNN and its variant LSTM-RNN have been used for language modeling (Sundermeyer et al., 2015).
ReNN	✓				This network aims to capture structural relationship (tree-like structure) amongst a set of data (Ditzler et al., 2015; Socher et al., 2011). ReNN have been used for processing natural language sentences (Socher et al., 2011).
GAN	✓		✓		The goal of a generative model of a GAN is to capture the training data statistical distribution. This can be used to generate results that are not part of the training data but can trick an observer to think it belongs to the training data (Goodfellow et al., 2014). GAN can be used for synthesizing human images (Creswell et al., 2018).
AE , SAE & VAE	✓	✓	✓	✓	AE encodes and decodes data with a goal of compressing the data. Doing so results in extracting useful information while filtering out irrelevant information (W. Liu et al., 2017). The basic unit of an AE is the feed forward neural network whose target is the same as the inputs (W. Liu et al., 2017). Stacking several AE results in a SAE (Zhang et al., 2018a). The VAE encodes and decodes data like the SAE, but the latent parameters of VAE are forced to assume a specific statistical distribution (Ma et al., 2020; Pouyanfar et al., 2018). Thus additional data can be obtained by sampling from the latent distribution thereby making it a generative model. VAE was used for anomaly detection in (Sun et al., 2018b).

2.4 Factors that can influence the progress in the use of machine learning

Considering that machine learning requires a computing device to learn from data, the factors to be considered are training data and computing packages.

Training data presents two major forms of problems which are: the availability of sufficient data for training and the time/resources required to train massive amount of data. Fortunately this challenges can be ameliorated by public availability of data sets and the use of transfer learning. One question probably could be how large should a dataset be for deep learning? A general list of datasets in different research areas has been compiled in Wikipedia (Mohammadi et al., 2018; Pu et al., 2020; Wikipedia, 2020). The number of instances (e.g. number of images for learning/testing in image detection based machine learning) corresponding to the dataset with the highest number of instances is selected for each research area and shown in Fig. 2.2.

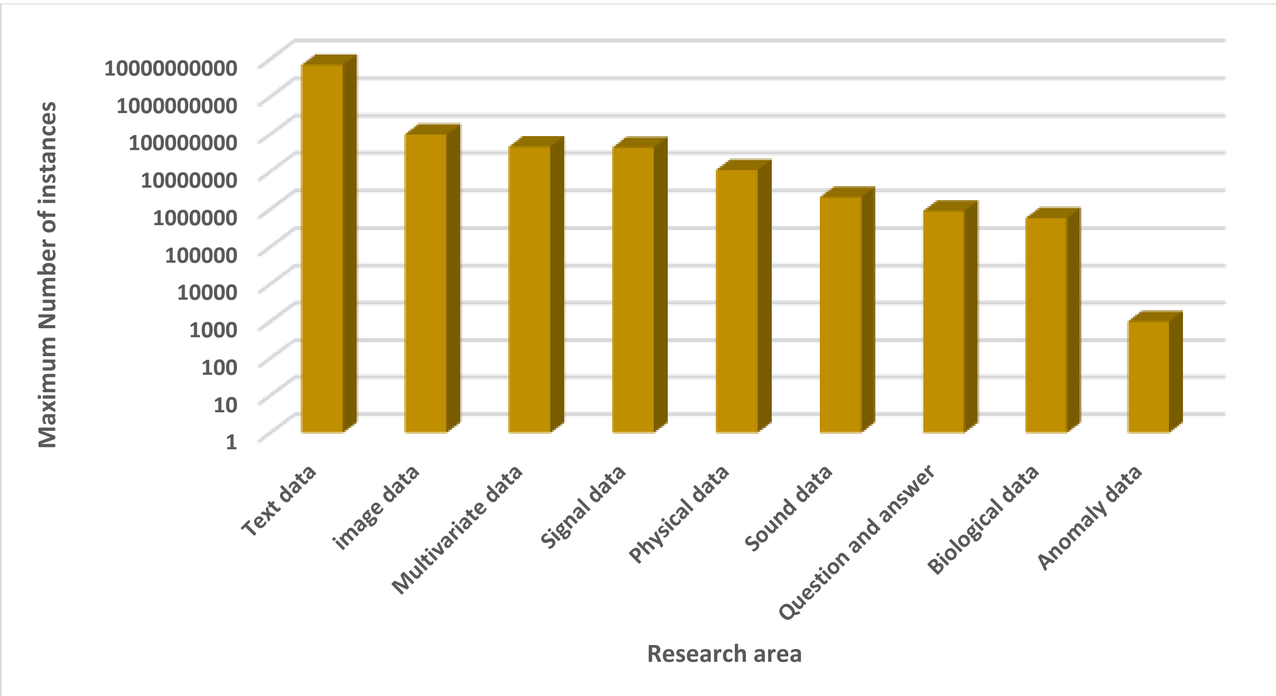

Fig. 2.2 Research areas and their top dataset in terms of quantity of data

Fig 2.2 shows 6 out of the 9 research areas having instances greater than one million. This could serve as a guide in setting up a database in petroleum drilling engineering for deep learning.

With the availability of a large dataset in a research area, transfer learning and pre-training can be exploited. Transfer learning refers to the situation whereby learning achieved by training a machine learning to perform a task in a given domain can be exploited in learning how to perform another task in the same domain or similar task in another domain or another task in another domain. For example, a neural network called X that is trained with data from field X to predict porosity from drilling parameters may become useful for pore pressure prediction using drilling parameters in field X. Likewise, the neural network X may also be useful in porosity prediction in field Y. There is also a possibility that neural network X could aid in pore pressure prediction in field Y. Transfer learning provides the opportunity to gain from previously learnt knowledge and this could translate to continuous improvement in performance when the algorithm is continuously updated with available data. Transfer learning/pre-training is used to mitigate the challenge of obtaining sufficient data for a given task and to save time and energy required to train large data sets (Alom et al., 2019; Pan & Yang, 2010; Weiss et al., 2016). This is even more obvious with deep learning where large amount of data is required for good performance as shown in Fig. 2.1. In deep learning, transfer learning can be implemented by first pre-training with the data from which learning is to be transferred from before any or both of the following is done: (1) fine-tuning some/all of the weights and biases of the deep learning algorithm with the actual data of primary interest (2) freeze the weights and biases but train the final classifier/regression layer. The option to be adopted after pre-training depends on the relative difference in quantity and nature of the pre-training and primary training data. For example, freezing weights and biases is recommended over fine-tuning when the primary dataset is very small in comparison to the pre-training dataset (Alom et al., 2019). A possible application of pre-training in drilling engineering can be recommended with the occurrence of this two scenarios: (1) There is a public or easily accessible drilling dataset called data V which has millions of instances/data points of drilling parameters (e.g. torque, weight on bit etc.,) and sonic compressional velocity from different oil fields, (2) a researcher plans to build a sonic compressional velocity prediction model for field W using deep learning but has only obtained few hundred of instances/data point of drilling parameters and sonic compressional velocity for field W. Pre-training is encouraged in this situation in that there could be a global/general rule governing the relationship between drilling parameters and sonic compressional velocity which should first be learnt from data V. After this, data W can be brought in for the deep learning algorithm to capture the relationship unique to field W.

The availability of user-friendly machine learning frameworks (computing software packages) can encourage researchers to use these algorithms. (Alom et al., 2019) listed 19 popularly used platforms for deep learning, some of which can also be used for other supervised machine learning algorithms. The platforms are: Tensorflow, Caffe, Keras, Theano, Torch, PyTorch, Lasagne, DL4J, Chainer, DIGITS, CNTK, MatConvNet, MINERVA, MXNET, OpenDeep, PuRine, PyLerarn2 and TensorLayer, LBANN. The first year for which each of these platforms were made available to the public was obtained by checking their release date/document availability year. Based on this, a plot was made as shown in Fig. 2.3 to observe the number of platforms available from 2000 to 2019.

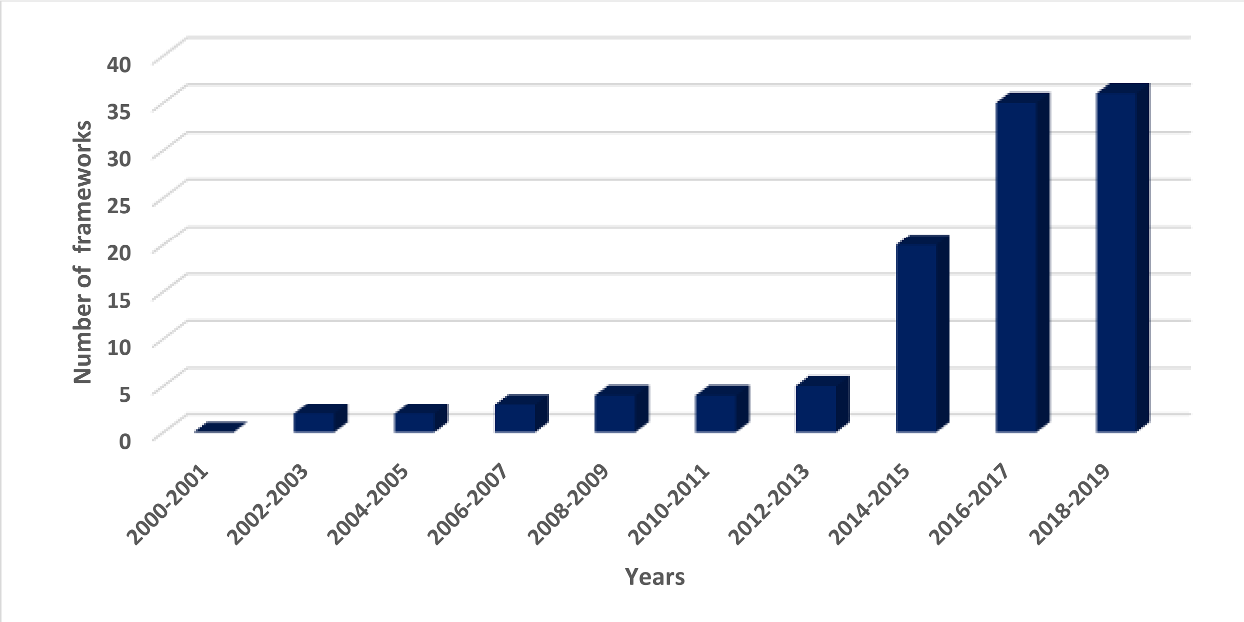

Fig. 2.3 Observation of trend in platform availability

It can be observed in Fig. 2.3 that there is a significant increase in the number of platforms available from 2014 upwards. Bibliometric analyses will be performed to observe if there is an observable correlation between platform availability and machine learning use.

2.5 Bibliometric analysis methodology

The databases considered for this work are Scopus and OnePetro. The main advantage of using Scopus is that it is the largest abstract and citation database of peer reviewed literature (Elsevier, 2020). OnePetro was used because it primarily focuses on oil and gas activities.

It is very tough to anticipate the precise words that authors used in capturing their intentions. Based on this, the search words were selected to obtain a fair representation of most documents of our interest. The database search involves four classes of search words/phrases. The first class was for supervised machine learning algorithms, the second class was aimed at constraining the results to oil and gas operations, the third class was to indicate that drilling represent our area of interest and the fourth class was for the hazardous events. For OnePetro database search, the second class of words/phrases was omitted because OnePetro already focuses on oil and gas operations.

For the first class, nine categories of search words are available depending on the type of supervised machine learning algorithm for which search is to be made for. These categories are:

<u>Linear regression category</u>

"linear regression*" OR "logistic regression*" OR "ordinary least squares regression*" OR "stepwise regression*" OR "multivariate adaptive regression spline*" OR "principal component regression*" OR "partial least squares regression*" OR "projection pursuit regression*" OR "ridge regression*"

<u>Discriminant analysis category</u>

"*linear discriminant analysis" OR "mixture discriminant analysis" OR "quadratic discriminant analysis" OR "flexible discriminant analysis"

<u>Decision tree category</u>

"classification tree*" OR "decision tree learning*" OR "*regression tree*" OR "m5 model tree*" OR "chi-squared automatic interaction detection*" OR "decision stump*"

<u>Random forest category</u>

"decision forest" OR "random forest"

<u>Instance based category</u>

"k nearest neighbo*" OR "k-nearest neighbo*" OR "locally weighted learning*" OR "learning vector quantization*"

Artificial neural network category

"neural network*" OR “radial basis function network*”

Support vector category

"support vector machine*" OR “support vector regression*”

Bayesian based category

"*naive bayes*" OR “averaged one-dependence estimator*” OR “bayesian belief network*” OR “bayesian network*” OR “Hidden Markov model*”OR “Conditional random field*”

Deep learning category

"Auto Encoder*" OR "Adversarial Network*" OR "Recursive Neural Network*" OR "Boltzmann Machine*" OR "Deep Belief Network*" OR "gated recurrent unit*" OR "Long short term memory" OR "Recurrent Network*" OR "Recurrent Neural Network*" OR "Convolutional Neural Network*" OR "Convolutional Network*" OR "deep learning*"

Combining the four classes in order to search for all articles in the linear regression category gives the search words/phrases using Scopus database as shown:

("linear regression*" OR “logistic regression*” OR “ordinary least squares regression*” OR “stepwise regression*” OR “multivariate adaptive regression spline*” OR “principal component regression*” OR “partial least squares regression*” OR “projection pursuit regression*” OR “ridge regression*”)

AND

("petroleum" OR "oil" OR "gas" OR "hydrocarbon*" OR "offshore*" OR "subsea*" OR “reservoir*” OR “formation*” OR “rock”)

AND

("drilling*")

AND

(“kick*" OR “pipe sticking*” OR “stuck pipe*” OR “fracture*” OR “lost circulation*”)

The supervised machine learning algorithms were obtained from (Freeman & Chio, 2018; Mohammed et al., 2017; Russell & Norvig, 2010; Shaier, 2020). The focus is on supervised machine learning algorithms, and as such, algorithms such as k-means clustering, self organizing map, principal component analysis etc., are not included in the search term. We choose not to include terms such as Gradient Boosting or AdaBoost because they are meant to improve the performance of supervised machine learning algorithms such as decision tree. Optimization algorithms such as genetic algorithm, ant colony etc., were not included because optimization algorithms are used as a training tool for supervised machine learning algorithms. Although the artificial neural network category also covers deep learning algorithms such as convolution neural network and recurrent neural network, a separate category was introduced for deep learning in order to observe the relative interest in deep learning. Curly bracket and asterisks were used to enhance accuracy of search. For example, “neural network” will search for “neural network” or “neural-network” as a unit. Excluding curly bracket will select an article that has the words “neural” and “network” not used together. Also, the use of asterisk sign (*) in the example "k nearest neighbo*" will search for "k nearest neighbour", "k nearest neighbor", "k nearest neighbours" and "k nearest neighbors"

Hazardous events that can be directly incurred or can be highly influenced by overbalance or underbalance drilling conditions were selected. These can be obtained from end events of Fig. 3

and Fig. 4 in the article by (M. Abimbola et al., 2015). The title, abstract and keywords fields of the contents of Scopus database were used for searching. Searching was done for all years of available documents in the database. The search method utilized was not case sensitive. Although there are chances of authors using abbreviations without giving the full meaning, abbreviations have different meanings in different contest. Hence, abbreviations were not used as search words.

The significant difference between the search used for OnePetro and that used for Scopus is that the use of asterisk (*) for stem word utilization was absent for OnePetro. Based on this, words and their plural equivalent were used for OnePetro search. For example, the linear regression category search with OnePetro was done by the following:

("linear regression" OR "linear regressions" OR “logistic regression” OR "logistic regressions" OR “ordinary least squares regression” OR "ordinary least squares regressions" OR “stepwise regression” OR "stepwise regressions" OR “multivariate adaptive regression spline” OR "multivariate adaptive regression splines" OR “principal component regression” OR "principal component regressions" OR “partial least squares regression” OR "partial least squares regressions" OR “projection pursuit regression” OR "projection pursuit regressions" OR “ridge regression” OR "ridge regressions")

AND

("Drilling")

AND

("kick" OR "pipe sticking" OR "stuck pipe" OR "fracture" OR "fractured" OR "fractures" OR "lost circulation")

2.6 The results of bibliometric analyses

The bibliometric search was done on 27th July, 2020 and the total search result for the different categories of supervised machine learning algorithms is presented in Fig. 2.4.

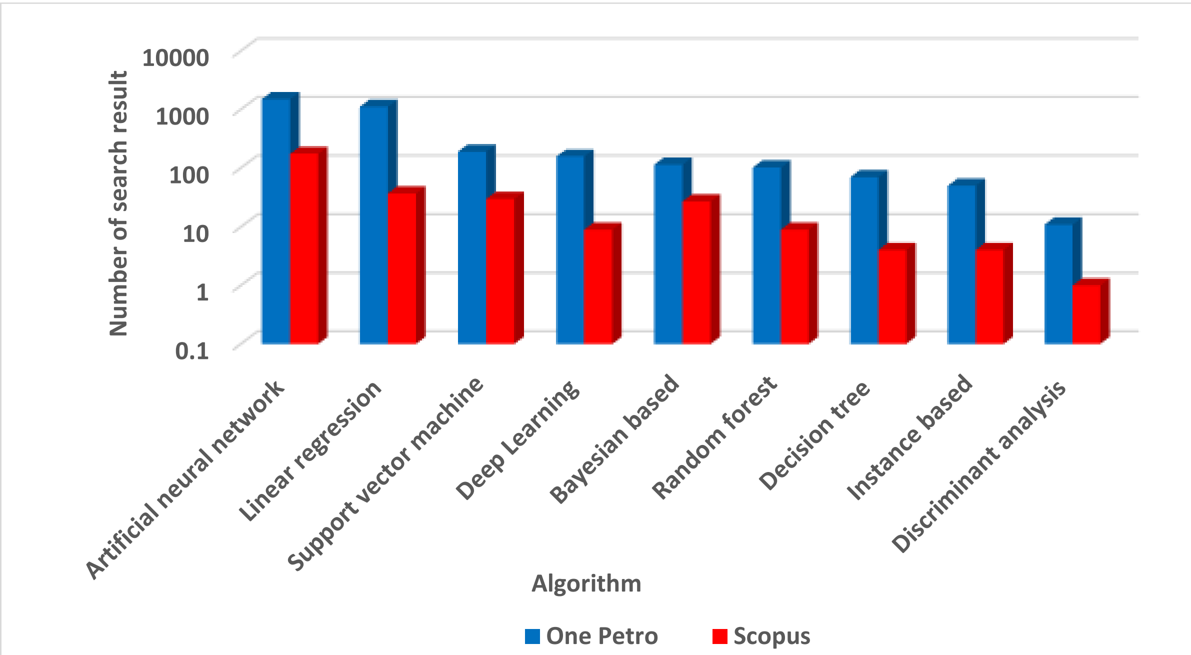

Fig. 2.4 Total number of supervised machine learning algorithms

Fig. 2.4 clearly shows artificial neural network leading in popularity for hazardous events in drilling. A breakdown of the relative use of the algorithms over the years is shown in Fig. 2.5. This gives an indication of usage trend over time.

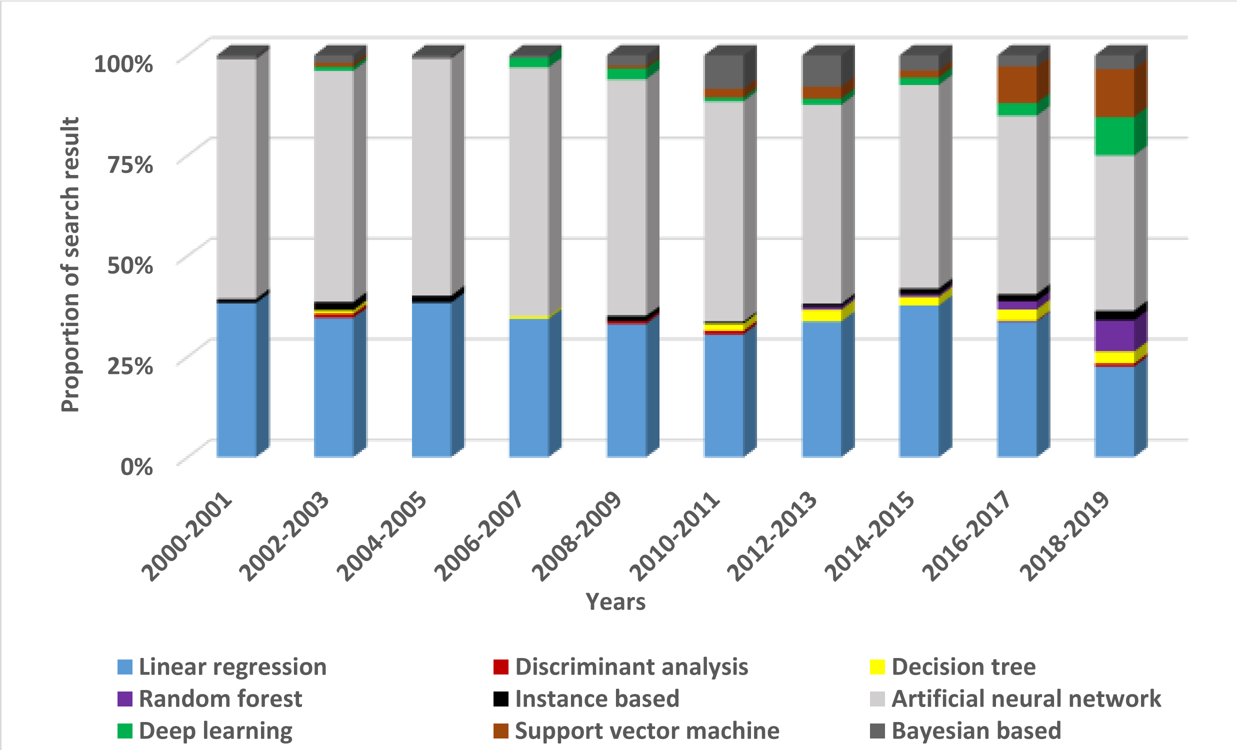

Fig. 2.5 The trend in the use of supervised machine learning algorithms

It can be observed in Fig. 2.5 that the relative use of deep learning, support vector machine and random forest algorithms have recently gained momentum (between 2014 to 2019). This is similar to the observation in Fig. 2.3 where the availability of software platforms gained significant increase from around 2014. Thus, there is the likelihood that researchers in drilling engineering are directly or indirectly benefiting from the availability of more software platforms for machine learning. A justifiable reason why researchers would prefer deep learning, support vector and random forest is because of their performance. While deep learning is recommended for large amounts of data (as discussed in Section 2.3), SVM and random forest can be ranked among the best sets of machine learning classifiers for non-large scale problems as observed in (Fernández-Delgado et al., 2014). In Fernández-Delgado et al., (2014), 179 machine learning classifiers from 17 machine learning family were tested using 121 datasets. Overall, the random forest family came out best followed by the SVM family.

2.7 Review of machine learning and hazardous events in drilling engineering

When pressure gradient due to drilling fluid density is below pore pressure gradient, kick can occur. On the other hand, when pressure gradient due to drilling fluid density is more than formation fracture gradient, formation fracture can also occur. When these events are not properly controlled, blow out or well collapse could occur (Abimbola et al., 2014; Khakzad et al., 2013; Khoshnaw et al., 2014; Zhang & Yin, 2017). This therefore necessitates a drilling fluid density window of operation in order to avoid both extremes i.e. too high or too low drilling fluid densities (Brahma & Sircar, 2018; Feng et al., 2015). Table 2.3-2.6 contains review on the use of machine learning on the following drilling hazardous events: kick, formation fracture, lost circulation and pipe sticking/stuck pipe. In addition, Table 2.7 shows a summary of reviews on the use of machine learning in determining the following parameters: pore pressure, equivalent circulating density (ECD) and bottom hole circulating pressure (BHCP). These parameters were considered because they directly influence the hazardous events of Tables 2.3-2.6.

Table 2.3 Machine learning and kick

Article source	Input parameters	Machine learning algorithm	Data size used	Findings
(Yin et al., 2014)	Stand pipe pressure (SPP) variation, drilling time variation, cell volume variation, outlet flow rate variation, outlet conductivity variation, outlet volume density change, outlet temperature variation, total hydrocarbon gas volume variation and C1 component content variation.	Artificial neural network (ANN)	The section of YYS well from 2560.7 m to 2601.2 m was for training. The section of YY well from 2568.6 m to 2610.1 m and the section of well ZZ from 1374.9 m to 1434.3 m was used for testing.	The algorithm employed achieved accuracy rate of 91.4% and 86.7% for early warning for kick.
(Haibo, L; Tang, Y; Li,	Casing pressure and SPP.	Bayes discriminant	The section of YYL well from 2310.3 m to 2333.7 m was used for training. The section of YY well from 2312.1 m to 2334.6 m was used for testing.	The change in trend of casing pressure and SPP data enabled kick and loss detection using the Bayes discriminant algorithm.
(Liang et al., 2019)	Casing pressure and SPP.	ANN	110 samples were used for training and 10 samples were used for testing.	Using genetic algorithm as an optimization methodology to train the back propagation neural network improved prediction accuracy in comparison to the conventional implementation of back propagation neural network (using gradient of error function which could result in being stuck at a local minima).

(Adedigba et al., 2018)	Mud pressure, fracture pressure and pore pressure.	Bayesian network	Time dependent probability of kick for 35 data points was evaluated.	The Bayesian Tree Augmented Naïve Bayes algorithm provided an efficient approach for predicting the dynamic risk of the drilling operation.
(Zhang et al., 2018a)	The bow-tie model for analysis had 23 parameters which could be classified based on their ability to contribute to any of the following categories: (i) abnormal formation pressure (ii) drop in bottom hole pressure (iii) managed pressure drilling (MPD) system failure (iv) rotating control device failure. For example, the parameter “flow meter failure” contributes to MPD system failure.	Dynamic Bayesian network	Experts knowledge/literature were used in building the network.	Considering the three drilling phases which are, circulating, making connection and tripping out; the tripping out phase has the highest chances of having kick. The following were found to be the most significant root cause of kick: "high speed of tripping out" in the tripping out phase, "backpressure pump failure" in making connection phase and "rig pump failure" in the circulating phase.
(Xie et al., 2018)	ROP, mass per unit volume of a drilling fluid (“Schlumberger definition”), mud weight (MW) going into the well, MW going out of the well, MW of circulating fluid and depth of well.	Genetic wavelet ANN	1440 sensor data records were obtained. 1000 records were used for training and remaining 440 records for testing.	Using genetic algorithm improved training time and prediction accuracy of the neural network.

Table 2.4 Machine learning and fracture/fracture pressure

Article source	Input parameters	Machine learning algorithm	Data size used	findings
(Malallah & Nashawi, 2005)	Pore pressure gradient, depth and rock density.	ANN	21,513 data points from sixteen wells in seven different geologic prospects were available. 20,975 data points from fifteen wells were used for training and 537 data points from sixteen wells were used for testing.	ANN obtained better results than those obtained from existing correlations (Eaton's (1969) correlation) in predicting fracture gradient. Test results of wells from seven regions shows average absolute relative error (%) of 0.31, 7.78, 3.21, 0.09, 0.04, 0.001, 0.021.
(Soroush et al., 2010)	Gamma ray, sonic log, resistivity log, porosity log, density log, photoelectric factor, caliper, MW and in-situ vertical stress.	Bayesian classifier	Data from four drilled wells were used. About 70% of data was used for training and 30% for testing.	Bayesian classifier on de-noised data identified breakout zones and non-breakout zones in wells with average accuracies between 78% and 93% depending on train wells used.
(Tokhmchi et al., 2010)	Energy log of the following: Caliper log, sonic log, density log and PEF (lithology) log.	linear & power regression	All data from each of eight wells were used for linear and power equation fitting (no separate test and train data).	Sonic, caliper and density log had the best correlation with fracture density for eight studied wells.
(Chao et al., 2015)	Seismic wave impedance and overburden pressure data.	ANN	Data from well IE which spans from about 700 m to 2700 m was used for training and testing was done using data of well WZ with two data points of formation pressure obtained by leak-off tests.	Prediction accuracies above 90% were achieved for fracture pressure.

(Abdideh, 2016)	Petrophysical log energy using caliper, sonic, density and lithology (PEF) log obtained in fractured zone.	linear & power regression	Data from one well were used for linear and power equation fitting (no separation between test and train data).	Power regression yielded better correlation coefficient than the linear regression in estimating fracture density.
(Fang et al., 2017)	Acoustic logs, density logs, compensated neutron logs, gamma ray and ant attribute logs from seismic review.	ANN	223 m of core data from 7 wells were available, along with seismic survey covering 60 km^2 and 110 wells of log data.	The ant tracking attribute from seismic review enhanced detection of small faults and large-scale fractures in the buried hill reservoir.
(Roy et al., 2018)	Tensile strength, P-wave velocity and S-wave velocity.	ANN, FIS, ANFIS, MRA	36 data sets were used for training, 10 data sets were used for testing the developed model.	In predicting mode-I fracture toughness, ANFIS (adaptive neuro-fuzzy inference system) performed best, followed by FIS (fuzzy inference system) and ANN. The MRA (multiple regression analysis) performed least.

Table 2.5 Machine learning and lost circulation

Article source	Input parameters	Machine learning algorithm	Data size used	findings
(Alireza et al., 2011)	Eighteen parameters were considered, these include: present depth of well from ground surface, Asmary formation top from ground surface, northing and easting of the considered well, bit size, average output of pump in gallon per minutes, average pump pressure, MW, solid percent of drilling fluid and amount of loss of circulation in day pervious of considered day.	ANN	Data from 32 wells were used. About 70% of input data were used for training, 15% for testing and 15% for validation.	R value for training is 0.95, for testing is 0.76 and for validation is 0.82.
(Toreifi & Rostami, 2014)	Geographic coordinates (east and north), the current depth, depth of formation tip, ROP, formation type, annulus volume, mud pressure, flow rate of mud pump, mud pump pressure, filter cake viscosity, solid content, plastic viscosity (PV), yield point (YP), initial strength and final strength after 10 min.	ANN	1,630 data sets from 38 wells were used. 60% of the data were used for training, 20% of the data were used for validation and 20% of the data were used for testing.	Modular neural network performed better than multilayer perceptron networks in terms of accuracy for the data used.
(Haibo, L; Tang, Y; Li, X; Luo, 2014)	Casing pressure and SPP.	Bayes discriminant	The section of YYL well from 3123.5 m to 3133.6 m was used for training. The section of YY well from 3122.1 m to 3131.4 m was used for testing.	The change in trend of casing pressure and standpipe pressure data enabled kick and loss detection using the Bayes discriminant algorithm.

(Wu et al., 2016)	21 factors where considered. These include: unreasonable design of drilling fluid density, flow meter failure, natural micro fracture, low formation fracture pressure, large formation porosities, casing failure, cement failure, increased running drillpipe rate, effect of high temperature, high drilling fluid viscosity, large rig pump out and high pump pressure.	Dynamic Bayesian network	Literature/expert view was used in obtaining prior probabilities.	The risk of lost circulation was considered for the following phase or scenarios: “not circulating”, “tripping in” and circulating. The circulating scenario had the highest occurrence probability of lost circulation. The factors "reasonable drilling fluid density" and "availability of the components of MPD system" were the most significant contributors to lost circulation for the circulating scenario.
(Far & Hosseini, 2017)	MW, depth, pump pressure and flow rate of pump.	ANN	Data were obtained from 3 wells. 70% of the data were for training, 15% of the data were used for validation and 15% of the data were used for testing.	Lost circulation increases with increase in pump pressure as well as MW but decreases by increase in pump flow rate due to hole cleaning and cutting transportation.
(Wu et al., 2019)	Mud flow out paddle, SPP, total volume of pit and mud flow in rate.	Bayesian inference algorithm	The data used for validating the algorithm had data point interval of 5 seconds and the data time ranged from 20:20:07 to 22:34:57.	Combining dynamic-threshold-based diagnosis and Bayesian estimation can improve performance.

(Sabah et al., 2019)	Depth, northing, easting, hole size, weight on bit (WOB), pump rate, pump pressure, bit revolution per minute (RPM), viscosity, shear stresses at shear rates of 600 and 300 RPM, gel strength, drilling time, drilling meterage, solid percent obtained from retort test, formation type, pore pressure, drilling mud pressure and formation fracture pressure.	Decision tree, ANN, ANFIS	1900 data points from 61 wells were used. 70% of the data points were used for training and 30% were used for testing.	Decision tree performed best. Although ANN and ANFIS had better results for smaller number of variables, the decision tree became more efficient as the number of variables increased.
(Ezeakacha & Salehi, 2019)	Cedar fiber concentration and fracture width.	linear regression	Experiments were performed to obtain fluid loss for different cedar fiber concentration and fracture width, thereby resulting in 9 data points.	Analysis of variance showed that changes in cedar fiber concentration and vertical fracture width affect dynamic fluid loss. Experimental results also showed that fracture orientation and positioning also influences dynamic fluid loss.
(Abbas et al., 2019)	Lithology, MW, flow rate, ROP, circulating pressure, inclination, solids content, fluid loss, RPM, WOB, YP, PV, marsh funnel viscosity, 10-second gel strength, 10-minute gel strength, azimuth, measured depth and hole size.	ANN, support vector machines (SVM)	The total data point/cases of 1120 was used. The data was split at a ratio of 3:1 for training and testing.	SVM performed slightly better than ANN. Based on the significances of the input parameters, lithology, MW, flow rate, rate of penetration, circulating pressure, inclination and solids content were most significant.

Table 2.6 Machine learning and pipe sticking

Article source	Input parameters	Machine learning algorithm	Data size used	findings
(Hempkins et al., 1987)	Measured depth, true vertical depth (TVD), average drill gas, connection gas, maximum drill gas, trip gas, review angle, MW, PV, YP, 10-seconds gel strength (GS), 10-minutes GS, water loss, filtrate pH, filtrate chloride, filtrate calcium, oil percent, water percent, hole size, flow rate, drill-collar OD, bit depth, bottom hole assembly (BHA) length, drill collar length, drag, time stuck, torque and percent solids.	Multivariate discriminant analysis	131 cases for sticking in addition to other non sticking cases were used for training and 47 cases from 35 wells where used for testing.	Multivariate discriminant analysis was used to differentiate between mechanically stuck, differentially stuck and non-stuck wells with high success rate of 81 %.
(Shadizadeh et al., 2010)	pH (alkalinity of mud), GF (geometric factor which is a function of the dimension of the hole, BHA and drill collar), YP, PV, 10-minute GS, chloride content, RPM, ROP, differential pressure and annular velocity.	ANN	The dataset used for dynamic condition had 155 non stuck cases and 40 stuck cases. The dataset used for static condition had 156 non stuck cases and 75 stuck cases. The data was split into training, validation and testing according to the ratio 80%, 10% and 10% for both dynamic and static conditions.	For dynamic conditions, the following parameters were more significant: differential pressure, pH, GF, RPM, ROP and PV. For static conditions, differential pressure, GF, pH, YP, PV and GS were more significant. The algorithm was able to achieve overall prediction accuracy greater than 90%.

(Alireza et al., 2011)	Type of formation, top of the formation in meter, dip of the well at the desired point, depth which well was deviated, size of last casing or liner run in well, depth of last casing or liner in the well, bit size, MW, Marsh funnel viscosity, PV, YP, initial GS and 10 minutes GS.	ANN	The total data had 266 data points with 166 data points corresponding to stuck pipe occurrence and 100 data point having no problem. About 70% of input data were used for training, 15% for testing and 15% for validation.	R value for training is 0.95, for testing is 0.76 and for validation is 0.82.
(Jahanbakhshi et al., 2012)	Differential pressure, hole depth, mud filtrate viscosity, fluid loss, solid content, PV, YP, initial GS, 10 minutes GS, BHA, still-pipe time and hole-size.	SVM, ANN	The total data points/samples is 214. 70% was used for training and 30% was used for testing.	SVM yielded better results than ANN.

Table 2.7 Machine learning, pore pressure, ECD and BHCP

Article source	Output	Input parameters	Machine learning algorithm	Data size used	findings
(Chao et al., 2015)	Pore pressure	Seismic wave impedance and overburden pressure data.	ANN	Data from well IE which spans from about 700 m to 2700 m was used for training. Testing was done using data of well WZ with three data points of pore pressure obtained from formation pressure testing.	Prediction accuracies above 90% were achieved.
(Ahmed et al., 2019a)	Pore pressure	WOB, RPM, ROP, MW, bulk density, porosity and compressional time.	ANN	245 data points were used. 70% of the data were used for training and 30% for testing.	An empirical equation from the neural network for pore pressure prediction was presented. This offers the benefit of not requiring prior pressure trend for pore pressure prediction.
(Ahmed et al., 2019b)	Pore pressure	WOB, RPM, ROP, MW, bulk density, porosity and compressional time.	SVM, ANN, RBF, fuzzy logic &	245 data points were used. 70% of the data were used for training and 30% for testing.	Test with all algorithm had the coefficient of determination (R^2) greater than 0.99 and average absolute percentage error (AAPE) values less than 0.4%.

(Vega et al., 2016)	ECD	Time, depth, hole depth, TVD, standpipe pressure, internal pressure, flow, torque, annular pressure, inclination, RPM, WOB, temperature and block position.	ANN	5000 data points were used for training and 5000 data points were used for validation.	The goal was to have an effective control system for smart monitoring and regulation of the ECD. The neural network based controller offered faster disturbance rejection than classic feedback controller.
(Abdelgawad et al., 2019)	ECD	ROP, MW and drillpipe pressure (DPP).	ANN and ANFIS	2376 data points were used in the work. Approximately 70% data points were used for training and 30% data points were used for testing.	Analysis was performed to observe the correlation between the input parameters and ECD. The DPP and MW had very high correlation coefficient with ECD, unlike the ROP. Testing achieved correlation coefficient (R) greater than 0.99 for both training and testing of both algorithms.
(Alkinani et al., 2019b)	ECD	MW, YP, PV, RPM, flow rate, WOB and nozzles total flow area.	ANN	Over 100,000 data points from over 2000 wells were used in the work. 70% of the data points were used for training, 15% for verification and 15% for testing.	The resulting equation of the neural network and the equations which can be used in normalizing the inputs and de-normalizing the output were presented. With these equations, a desired ECD can be designed.
(Ashena & Moghadasi, 2011)	BHCP	Measured depth, TVD, drillpipe injected gas, liquid flow rates, casing pressure, surface temperature and liquid density at surface.	ANN	160 data points were used for the work. 110 data points were used for training and 50 data points were used for validation.	ANN optimized by ant colony and ANN optimized by genetic algorithm performed better than that of conventional back propagation ANN.
(Tamim et al., 2019)	low hydrostatic head	Several factors including abnormal pore pressure, mud loss, improper density, swabbing, gas cut mud where used for the fault-tree analysis.	Bayesian network	Probability values were obtained from expert judgements and past incidents.	Abnormal pore pressure and swabbing were found to be the major contributors for low hydrostatic head.

2.7.1 Summary of review on machine learning for detecting/predicting kick, fracture, lost circulation, stuck pipe, pore pressure, ECD and BHCP

In summary, the following could be observed for kick.

(i) The tripping out phase/state has the highest probability of experiencing kick among the different drilling state.

(ii) Several parameters have been exploited for kick detection which includes, pressure parameters (e.g. standpipe pressure, casing pressure), mud weight, flow rate, conductivity, rate of penetration, temperature and depth.

(iii) ANN is very common and the optimization algorithm used for the ANN plays a critical role in performance.

(iv) The change in trend of input parameters can be exploited for kick detection.

In summary, the following could be observed for fracture.

(i) Parameters with capability to capture strength/density of materials such as seismic data, sonic log and density log are very important.

(ii) ANN is very common and exhibited the capability to perform better than existing correlation equations.

(iii) The performance of machine learning can be enhanced with appropriate de-noising and fuzzy logic implementation.

In summary, the following could be observed for lost circulation.

(i) The circulating phase has the highest probability of experiencing lost circulation among the different drilling state/phase/scenario.

(ii) The following parameters significantly influence lost circulation: pressure data (e.g. standpipe, casing, pump pressure), drilling fluid density, flow rate, lithology and rate of penetration. Also, the concentration of lost circulation materials and nature of well fracture are important for evaluating fluid loss.

(iii) ANN was more popularly used. Other machine learning algorithms such as SVM and random forest had better performance.

(iv) The performance of machine learning can be enhanced by considering trend/dynamic nature of data.

In summary, the following could be observed for stuck pipe.

(i) The following parameters, differential pressure, alkalinity of mud, geometric factor, rotary speed, rate of penetration, plastic viscosity, yield point and gel strength are important for stuck pipe prediction.

(ii) ANN is more popular, but SVM performed better than ANN, based on the literature surveyed.

(iii) It is possible for machine learning to distinguish between mechanical and differential sticking.

In summary, the following could be observed for pore pressure, ECD, BHCP.

(i) For pore pressure, drilling parameters such as weight on bit, rotary speed, penetration rate, mud density (mud weight) and parameters with capability to capture strength/density of materials such as seismic data, sonic log, density log and porosity log were utilized. A close look at the drilling parameters shows that they form the bulk of input parameters required to compute corrected d-exponent.

(ii) Drill pipe pressure and mud weight have very good correlation with equivalent circulating density. Other drilling conditions/activities such as swabbing can increase pressure head loss. An empirical equation which utilizes several parameters was also developed for computing equivalent circulating density.

(iii) ANN was commonly used and the optimization algorithm used for the ANN plays a critical role in performance.

2.8 Literature review on the deep learning and drilling parameters

In this section, we aim to observe how deep learning have been utilized with drilling parameters.

Table 2.8 Applications of deep learning with drilling parameters

Article source	Input parameters and dimension used. (It should be noted that all input window size per output instance are one dimensional unless otherwise stated)	Output parameter	Machine learning algorithm	Data size	Findings/some additional notes
(Zha & Pham, 2018)	The input parameters are: surface data consists of high frequency torque, tension, RPM, WOB and triaxial acceleration data measured at 100 Hz. Downhole channels include torque, RPM and acceleration measured at 40 Hz but resampled to 100 Hz. The window of input considered for each output is 60 seconds.	Drilling vibration	One dimensional CNN and RNN	1400 samples from multiple wells were used for training and validation with the ratio of 75 to 25.	With training data enhancement by augmentation, accuracy for the validation set improved from 96% to 99% and precision for the validation set improved from 73% to 97%.
(Ben et al., 2020)	Two input, “Rotary RPM” and rotary torque were used. Moving window of 20 seconds was employed to obtain sets of data, i.e. current drilling state depends on the last 20 seconds of data. The data had a frequency of 1 data point per second.	rig state	CNN and U-Net architecture	Over 30 million timestamped data point from 40 wells in U.S. onshore basins.	The input data sets were treated as image like data but with the following similarities/difference. (1) While image has three channels for red, green and blue, the input data of this work uses two channels for rotary RPM and torque. (2) While image is in 2 dimension, the data in this work is in 1 dimension. Test result had f1 score ≥ 0.99 and accuracy of over 99%.

(Kaneko et al., 2018)	The input parameters are: surface measured WOB and crown block position. An input sequence window of 20 seconds was used for each output.	Downhole WOB	RNN	Synthetic data from simulations were used. The number of training data set is 10,800. The number of test data set is 3000.	The algorithm was applied to data obtained different simulated models: linear, non-linear and various sea wave conditions. Visual observation of test results shows good performance for linear and non linear models, although some types of nonlinear implementation had lesser performance. The algorithm also performed well for different sea wave conditions as observed by R^2 values exceeding 0.9 in most cases.
(He et al., 2019)	The input parameters are: ROP, RPM, rotational torque and drilling force. Equivalent two-dimensional structures of gray values (pixel image format) were obtained from the input drilling parameter data. This was to present input data in suitable form for training with CNN. The input data for each case is a 2D array of 76 by 76 dimension.	Cohesion, internal friction angle, and Unconfined compressive strength.	CNN	30 cases were used for training and 6 cases were used for testing.	Results from a case study had error of performance less than 10% in comparison to results obtained from standard laboratory. The mean error for UCS obtained using the CNN algorithm was lower than that obtained using the Mohr–Coulomb criterion, based on data from the case study.
(Fjetland et al., 2019)	The input parameters are: flow rate in, flow rate out, SPP, choke pressure, choke opening, bit pressure and bit depth. Data was sampled at 1 Hz.	kick detection and influx size estimation	LSTM-RNN	5781 random drilling cases with 10 minutes of data where generated with OpenLab from NORCE. 29% of the simulations contained influx, 5% contained mud loss and 0.6% experienced simultaneous loss. 70% of data was used for training, 10% for validation and 20% for testing.	Algorithm achieves early kick detection with delay generally less than 4 seconds, except for very small influx values for which the algorithm failed to detect influx.

(Y. Yu et al., 2019)	The input parameters are: RPM, block position, hook load, flow rate, surface WOB, slip status, surface torque and SPP. Each input case/sample has a time window of 256 seconds.	Rig states e.g. rotary drilling, tripping out, pumping etc.	Combining one dimensional CNN and LSTM layers	115,200 samples were used for training and 42880 samples were used for testing.	97.5% accuracy was reached for rig state prediction on test data.
(Chen et al., 2019)	The input parameters are: RPM and accelerations in two lateral axes. Three channels with window size of 128 were used.	BHA whirling/non-whirling state	One dimensional CNN	17080 simulated datasets were used and 68% of the dataset were labeled as whirling cases. 80% of the dataset were used for training and 20% for testing.	Applying the algorithm achieved accuracy of 97% for testing.
(Alkinani et al., 2019c)	The input parameters are: WOB, RPM, MW, PV, YP, flow rate, total flow area and unconfined compressive strength. A delay line (window) of 4 was used.	ROP	RNN	30,000 drilling data points from more than 2000 wells were collected from various sources worldwide. 70% of the data were used for training, 15% of the data were used for validation and 15% of the data were used for testing.	R^2 for training, validation and test data all exceeded 0.92.

(Chen et al., 2020)	The input parameter used is drilling string vibration at a sample frequency of 20Hz. Butterworth high pass filter and short-time Fourier transform were applied to pre-process the drill string vibration. The pre-processed signal was expressed in 2D image formats for machine learning.	Lithology	CNN, MobileNet and ResNet.	The total data comprises of 1410 sets. 70% of the data were used for training, 20% were used for development and 10% for testing.	Macro-precision (average precision) of 0.90 and macro-recall (average recall) of 0.893 were achieved for the test data.
(Kanfar et al., 2020)	The input parameters: Depth, ROP, WOB, flow rate, and mechanical specific energy were used for predicting porosity and density. The input parameters: ROP, WOB, torque and mechanical specific energy were used for predicting compressional sonic. Each input case or instance has 50 time steps and each time steps has 5 channels. The five channels are for five input drilling parameters.	Porosity, density and compressional sonic	Inception-based CNN combined with a TCN	The total data used had 16680 cases. 5% of the data were used for testing.	Although correlation was low for test data (between 0.4 and 0.6), the figure showing test results indicated that the trends in the data were captured by the algorithm applied.
(Gupta et al., 2020)	The input parameters comprise of Hook load, RPM, depth, torque, flow and gamma ray. The sampling interval used for the parameter was 3s. For each output instance, a window of 1000 s was used for the input signals.	Output parameters are a subset of input parameters.	Recurrent Auto Encoders	Time series data obtained from thousands of wells were used for training. Test was done to spot error in several cases of time series data.	The algorithm was able to identify missing data points as well as outliers and sensor drift in data.

Table 2.8 shows deep learning algorithms been used with drilling parameters for drilling rig state determination (e.g. tripping), drilling event identification (e.g. kick), lithology identification, generating logging/other drilling parameters and detecting occurrence of abnormality in data.

2.9 Gaps in the use of machine learning in drilling events as observed in literature survey

Several issues were observed which were chiefly based on training data as well as the nature of deep learning algorithms. These are:

1. For most cases, each article used data different from that used in other articles. This can likely translate to machine learning algorithms which may only be successful for the field data they trained with.

2. Nearly all data set used for machine learning implementation had data points/instances less than a million, with many of the dataset having instances less than a thousand. This suggest possible room for improvement based on results shown in Fig. 2.2.

3. Nearly all data set used for training are not publicly available.

4. Deep learning algorithms used were not primarily designed or customized for drilling activities. For example, CNN and its variants were chiefly customized for image detection, and LSTM-RNN was developed to learn complex sequential relationship such as those in audio/language modeling or detection task (Chung et al., 2014; Hochreiter & Schmidhuber, 1997).

Based on these observed limitations, publicly accessible database with data from different oilfield is needed. Incentives for establishing publicly available and updatable drilling dataset include:

1. Transfer learning becomes easy to achieve.

2. More researchers in drilling engineering can easily conduct research with machine learning as the huddle of obtaining data would have been reduced

3. Competition for best performing machine learning algorithms becomes feasible. This would enhance development of machine learning algorithms well customized and suited for drilling problems.

2.10 Conclusions

Results of bibliometric analysis in the area of supervised machine learning in hazard related events during drilling clearly indicate a growing trend in the use of machine learning. The results of a review of the literature on supervised machine learning for hazardous drilling events -- kick, fracture, lost circulation, and stuck pipe -- are reported. In addition, some studies in the application of supervised machine learning on pore pressure, equivalent circulation density and bottom hole circulating pressure are also discussed because of the close relationship between these quantities and hazardous drilling events. A review of deep learning and drilling parameters is also presented. In addition to these, gaps in machine learning usage and the means by which such gaps can be mitigated is also provided. The conclusions based on the bibliometric analysis and reviews are presented here.

The following are the conclusions based on the bibliometric analysis detailed in this chapter:

(i) Artificial neural network is the most popular among all supervised machine learning tools considered.

(ii) Bibliometric analysis shows deep learning, random forest and support vector machine algorithms are gaining momentum in terms of recent usage by researchers.

(iii) Deep learning is recommended when a large amount of data is available, while random forest and support vector machine are recommended for non-large scale classification problems.

(iv) Increasing availability of different computing platforms has likely aided research in the use of machine learning especially, deep learning.

The following are the conclusions based on a detailed review on kick, fracture, lost circulation, stuck pipe, pore pressure, equivalent circulation density and bottom hole circulating pressure.

(i) ANN appears to be the dominant algorithm in popularity. The performance of ANN can be affected by the optimization algorithm used. Fuzzy logic can be applied with ANN to improve performance.

(ii) In other to aid machine learning, proper de-noising is recommended. Also, the trend/dynamic nature of data can be exploited for machine learning.

(iii) The input parameters used by authors in predicting/detecting kick, fracture, lost circulation, stuck pipe, pore pressure, equivalent circulating density have been presented.

(iv) The probability of the occurrence of a hazardous event is influenced by the drilling state/phase, e.g. the risk of kick is highest at the tripping out phase most likely due to high speed of tripping out.

(v) In specific cases, SVM performed better than ANN. However, it worthy to note that for big data, deep learning which is an offshoot of ANN is expected to perform better.

The following are the conclusions based on review on deep learning of drilling parameters:

(i) With developments in deep learning algorithms and more availability of computing platforms, it is expected that the capabilities of deep learning will be exploited to get better results from big data than from conventional/traditional machine learning.

(ii) Researchers are now beginning to use deep learning on drilling parameters for lithology identification, drilling rig state determination, drilling event identification, generating logging/other drilling parameters and detecting abnormality in data (data pre-processing).

(iii) CNN and RNN (including its variants e.g. LSTM-RNN) appear to be the most commonly used deep learning algorithms.

The following are the gaps in the use of machine learning:

(i) There is an absence of publicly accessible global database of drilling data. Typically, researchers perform machine learning based on the data from a particular oil field . The results of such analysis may not be generalizable.

(ii) Almost all datasets used had data points or instances fewer than a million, with many of the datasets having instances less than a thousand. Therefore, a lot of these datasets may not fully benefit from the capabilities of deep learning.

(iii) Most datasets used are not publicly available.

(iv) Most of the deep learning algorithms used were not primarily customized for drilling activity.

Based on these observed gaps, publicly accessible database with data from different oilfield is recommended as this will help researchers and industrial users to obtain more generalizable results, perform transfer learning to compensate for a new field with limited data and also encourage development of machine learning algorithms in line with the unique needs of drilling data/task.

2.11 Acronyms

ANFIS	adaptive neuro-fuzzy inference system
ANN	artificial neural network
ANOVA	analysis of variance
BHA	bottom hole assembly
BHCP	bottom hole circulating pressure
CNN	convolutional neural network
ECD	equivalent circulating density
FIS	fuzzy inference system
GS	gel strength
MPD	managed pressure drilling
MRA	multiple regression analysis
MW	mud weight

PV	plastic viscosity
RNN	recurrent neural network
ROP	rate of penetration
LSTM	long short-term memory
SPP	stand pipe pressure
SVM	support vector machines
SVR	support vector regression
RBF	radial basis function
RPM	revolutions per minute
TCN	temporal convolutional network
TVD	true vertical depth
WOB	weight on bit
YP	yield point

2.12 References

Abbas, A. K., Al-haideri, N. A., & Bashikh, A. A. (2019). Implementing artificial neural networks and support vector machines to predict lost circulation. Egyptian Journal of Petroleum, 28(4), 339–347.

Abdelgawad, K. Z., Elzenary, M., Elkatatny, S., Mahmoud, M., Abdulraheem, A., & Patil, S. (2019). New approach to evaluate the equivalent circulating density (ECD) using artificial intelligence techniques. Journal of Petroleum Exploration and Production Technology, 9(2), 1569–1578.

Abdideh, M. (2016). Estimation of the fracture density in reservoir rock using regression analysis of the petrophysical data. Geodesy and Cartography, 42(3), 85–91.

Abimbola, M., Khan, F., & Khakzad, N. (2014). Dynamic safety risk analysis of offshore drilling. Journal of Loss Prevention in the Process Industries, 30, 74–85.

Abimbola, M., Khan, F., Khakzad, N., & Butt, S. (2015). Safety and risk analysis of managed pressure drilling operation using Bayesian network. Safety Science, 76, 133–144.

Adedigba, S. A., Oloruntobi, O., Khan, F., & Butt, S. (2018). Data-driven dynamic risk analysis of offshore drilling operations. Journal of Petroleum Science and Engineering, 165, 444–452.

Agwu, O. E., Akpabio, J. U., Alabi, S. B., & Dosunmu, A. (2018). Artificial intelligence techniques and their applications in drilling fluid engineering: A review. Journal of Petroleum Science and Engineering, 167, 300–315.

Ahmed, A., Elkatatny, S., Ali, A., & Abdulraheem, A. (2019). Comparative analysis of artificial intelligence techniques for formation pressure prediction while drilling. Arabian Journal of Geosciences, 12(18), 592. https://doi.org/10.1007/s12517-019-4800-7

Ahmed, A., Elkatatny, S., Ali, A., Mahmoud, M., & Abdulraheem, A. (2019). New model for pore pressure prediction while drilling using artificial neural networks. Arabian Journal for Science and Engineering, 44(6), 6079–6088.

Alireza, M., Mohammad, N., & Ahmad, A. (2011). Reducing consumed energy while drilling an oil well through a deep rig time analysis. Advances in Petroleum Exploration and Development, 1(1), 22–31.

Alkinani, H H, Al-Hameedi, A. T., Dunn-Norman, S., Flori, R. E., Al-Alwani, M. A., & Mutar, R. A. (2019). Dynamic Neural Network Model to Predict the Rate of Penetration Prior to Drilling. 53rd US Rock Mechanics/Geomechanics Symposium.

Alkinani, Husam H, Al-Hameedi, A. T. T., Dunn-Norman, S., Flori, R. E., Alsaba, M. T., & Amer, A. S. (2019). Applications of artificial neural networks in the petroleum industry: A review. SPE Middle East Oil and Gas Show and Conference.

Alkinani, Husam H, Al-Hameedi, A. T. T., Dunn-Norman, S., & Lian, D. (2019). Application of artificial neural networks in the drilling processes: Can equivalent circulation density be estimated prior to drilling? Egyptian Journal of Petroleum.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. Electronics, 8(3), 292.

Alyafeai, Z., & Ghouti, L. (2020). A fully-automated deep learning pipeline for cervical cancer classification. Expert Systems with Applications, 141, 112951.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26–38.

Ashena, R., & Moghadasi, J. (2011). Bottom hole pressure estimation using evolved neural networks by real coded ant colony optimization and genetic algorithm. Journal of Petroleum Science and Engineering, 77(3–4), 375–385.

Bello, O., Holzmann, J., Yaqoob, T., & Teodoriu, C. (2015). Application of artificial intelligence methods in drilling system design and operations: a review of the state of the art. Journal of Artificial Intelligence and Soft Computing Research, 5(2), 121–139.

Ben, Y., Han, W., James, C., & Cao, D. (2020). Building a General and Sustainable Machine Learning Solution in a Real-Time Drilling System. IADC/SPE International Drilling Conference and Exhibition.

Brahma, J., & Sircar, A. (2018). Design of safe well on the top of Atharamura anticline, Tripura, India, on the basis of predicted pore pressure from seismic velocity data. Journal of Petroleum Exploration and Production Technology, 8(4), 1209–1224.

Bravo, C. E., Saputelli, L., Rivas, F., Pérez, A. G., Nickolaou, M., Zangl, G., De Guzmán, N., Mohaghegh, S. D., & Nunez, G. (2014). State of the art of artificial intelligence and predictive analytics in the E&P industry: a technology survey. Spe Journal, 19(04), 547–563.

Chao, W. U., Jianhua, L. I. U., ZHANG, D., Xiaofeng, C., & Weijie, Z. (2015). A prediction of borehole stability while drilling preliminary prospecting wells based on seismic impedance. Petroleum Exploration and Development, 42(3), 427–433.

Chen, G., Chen, M., Hong, G., Lu, Y., Zhou, B., & Gao, Y. (2020). A New Method of Lithology Classification Based on Convolutional Neural Network Algorithm by Utilizing Drilling String Vibration Data. Energies, 13(4), 888.

Chen, W., Yu, Y., Shen, Y., Zhang, Z., & Vesselinov, V. (2019). Automatic Drilling Dynamics Interpretation Using Deep Learning. SPE Annual Technical Conference and Exhibition.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv Preprint ArXiv:1412.3555.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. IEEE Signal Processing Magazine, 35(1), 53–65.

Ditzler, G., Polikar, R., & Rosen, G. (2015). Multi-layer and recursive neural networks for metagenomic classification. IEEE Transactions on Nanobioscience, 14(6), 608–616.

Du, Y., Yao, C., Huo, S., & Liu, J. (2017). A new item-based deep network structure using a restricted Boltzmann machine for collaborative filtering. Frontiers of Information Technology & Electronic Engineering, 18(5), 658–666.

Elsevier. (2020). About Scopus . In https://www.elsevier.com/en-in/solutions/scopus .

Ezeakacha, C. P., & Salehi, S. (2019). Experimental and statistical investigation of drilling fluid loss in porous media: Part 2 (Fractures). Journal of Natural Gas Science and Engineering, 65, 257–266.

Fang, J., Zhou, F., & Tang, Z. (2017). Discrete fracture network modelling in a naturally fractured carbonate reservoir in the Jingbei oilfield, China. Energies, 10(2), 183.

Far, P. B., & Hosseini, P. (2017). Estimation of lost circulation amount occurs during under balanced drilling using drilling data and neural network. Egyptian Journal of Petroleum, 26(3), 627–634.

Feng, S., Zhou, H., & Dong, H. (2019). Using deep neural network with small dataset to predict material defects. Materials & Design, 162, 300–310.

Feng, Y., Arlanoglu, C., Podnos, E., Becker, E., & Gray, K. E. (2015). Finite-element studies of hoop-stress enhancement for wellbore strengthening. SPE Drilling & Completion, 30(01), 38–51.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? The Journal of Machine Learning Research, 15(1), 3133–3181.

Fjetland, A. K., Zhou, J., Abeyrathna, D., & Gravdal, J. E. (2019). Kick detection and influx size estimation during offshore drilling operations using deep learning. 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2321–2326.

Freeman, D., & Chio, C. (2018). Machine learning and Security. O'Reilly Media: Newton, MA, USA.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in Neural Information Processing Systems, 2672–2680.

Gupta, S., Chatar, C., & R Celaya, J. (2020). Recurrent Auto Encoders for Automatic Sensor Validation; Tomorrows Data with Yesterday's Sensors. IADC/SPE International Drilling Conference and Exhibition.

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). Neural network design. PWS Publishing Co.

Hatcher, W. G., & Yu, W. (2018). A survey of deep learning: Platforms, applications and emerging research trends. IEEE Access, 6, 24411–24432.

Haykin, S. (2009). Neural networks and learning machines,–3rd ed., Copyright by Pearson Education. Inc., Upper Saddle River, New Jersey, 7458.

He, M., Zhang, Z., Ren, J., Huan, J., Li, G., Chen, Y., & Li, N. (2019). Deep convolutional neural network for fast determination of the rock strength parameters using drilling data. International Journal of Rock Mechanics and Mining Sciences, 123, 104084.

Hempkins, W. B., Kingsborough, R. H., Lohec, W. E., & Nini, C. J. (1987). Multivariate statistical analysis of stuck drillpipe situations. SPE Drilling Engineering, 2(03), 237–244.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Jahanbakhshi, R., Keshavarzi, R., Aliyari Shoorehdeli, M., & Emamzadeh, A. (2012). Intelligent prediction of differential pipe sticking by support vector machine compared with conventional artificial neural networks: An example of iranian offshore oil fields. SPE Drilling & Completion, 27(04), 586–595.

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

Kaneko, T., Wada, R., Ozaki, M., & Inoue, T. (2018). Combining Physics-Based and Data-Driven Models for Estimation of WOB During Ultra-Deep Ocean Drilling. ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering.

Kanfar, R., Shaikh, O., Yousefzadeh, M., & Mukerji, T. (2020). Real-Time Well Log Prediction From Drilling Data Using Deep Learning. ArXiv Preprint ArXiv:2001.10156.

Khakzad, N., Khan, F., & Amyotte, P. (2013). Quantitative risk analysis of offshore drilling operations: A Bayesian approach. Safety Science, 57, 108–117.

Khoshnaw, F., Jaf, P., & Farkha, S. (2014). Pore, abnormal formation and fracture pressure prediction. WIT Transactions on Ecology and the Environment, 186, 579–593.

Kumar, A., Kim, J., Lyndon, D., Fulham, M., & Feng, D. (2017). An ensemble of fine-tuned convolutional neural networks for medical image classification. IEEE Journal of Biomedical and Health Informatics, 21(1), 31–40.

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

Li, X., Ji, S., Han, M., Ji, J., Ren, Z., Liu, Y., & Wu, C. (2019). Adversarial examples versus cloud-based detectors: A black-box empirical study. IEEE Transactions on Dependable and Secure Computing.

Liang, H., Yongqiang, T., Xiang, L., & Yangyang, L. (2014). RESEARCH ON DRILLING KICK AND LOSS MONITORING METHOD BASED ON BAYESIAN CLASSIFICATION. Pakistan Journal of Statistics, 30(6).

Liang, H., Zou, J., & Liang, W. (2019). An early intelligent diagnosis model for drilling overflow based on GA–BP algorithm. Cluster Computing, 22(5), 10649–10668.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234, 11–26.

Ma, X., Lin, Y., Nie, Z., & Ma, H. (2020). Structural damage identification based on unsupervised feature-extraction via Variational Auto-encoder. Measurement, 107811.

Malallah, A., & Nashawi, I. S. (2005). Estimating the fracture gradient coefficient using neural networks for a field in the Middle East. Journal of Petroleum Science and Engineering, 49(3–4), 193–211.

Mamudu, A., Khan, F., Zendehboudi, S., & Adedigba, S. (2020). Dynamic risk assessment of reservoir production using data-driven probabilistic approach. Journal of Petroleum Science and Engineering, 184, 106486.

Mason, C. F. (2019). Policy brief—regulating offshore oil and gas exploration: Insights from the deepwater horizon experience in the Gulf of Mexico. Review of Environmental Economics and Policy, 13(1), 149–154.

MC.AI. (2019). Difference between ML and deep learning with respect to Splitting of the dataset into. In https://mc.ai/difference-between-ml-and-deep-learning-with-respect-to-splitting-of-the-dataset-into/.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2018). Deep learning for IoT big data and streaming analytics: A survey. IEEE Communications Surveys & Tutorials, 20(4), 2923–2960.

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2017). Machine learning: algorithms and applications. Crc Press.

Noshi, C. I., & Schubert, J. J. (2018). The role of machine learning in drilling operations; a review. SPE/AAPG Eastern Regional Meeting.

Osarogiagbon, A. U., Udeze, C. C., & Imorame, I. J. (2015). Modeling petrophysical property variations in reservoir sand bodies using artificial neural network and object based techniques. Society of Petroleum Engineers - SPE Nigeria Annual International Conference and Exhibition, NAICE 2015.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 51(5), 1–36.

PSAC. (2018). Industry Overview. In https://www.psac.ca/business/industry-overview/.

Pu, C., Suprem, A., Lima, R. A., Musaev, A., Wang, D., Irani, D., Webb, S., & Ferreira, J. E. (2020). Beyond Artificial Reality: Finding and Monitoring Live Events from Social Sensors. ACM Transactions on Internet Technology (TOIT), 20(1), 1–21.

Roy, D. G., Singh, T. N., & Kodikara, J. (2018). Predicting mode-I fracture toughness of rocks using soft computing and multiple regression. Measurement, 126, 231–241.

Russell, S. J., & Norvig, P. (2010). Artificial Intelligence-A Modern Approach, Third International Edition. Pearson Education London.

Sabah, M., Talebkeikhah, M., Agin, F., Talebkeikhah, F., & Hasheminasab, E. (2019). Application of decision tree, artificial neural networks, and adaptive neuro-fuzzy inference system on predicting lost circulation: A case study from Marun oil field. Journal of Petroleum Science and Engineering, 177, 236–249.

Salakhutdinov, R., & Hinton, G. (2012). An efficient learning procedure for deep Boltzmann machines. Neural Computation, 24(8), 1967–2006.

Shadizadeh, S. R., Karimi, F., & Zoveidavianpoor, M. (2010). Drilling stuck pipe prediction in iranian oil fields: an artificial neural network approach. Iranian Journal of Chemical Engineering, 7(4), 29–41.

Shaier, S. (2020). An intro to machine learning algorithms. In https://towardsdatascience.com/ml-algorithms-one-sd-%CF%83-74bcb28fafb6 .

Sharan, R. V, & Moir, T. J. (2017). Robust acoustic event classification using deep neural networks. Information Sciences, 396, 24–32.

Socher, R., Lin, C. C., Manning, C., & Ng, A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. Proceedings of the 28th International Conference on Machine Learning (ICML-11), 129–136.

Soroush, H., Rasouli, V., & Tokhmechi, B. (2010). A data processing algorithm proposed for identification of breakout zones in tight formations: A case study in Barnett gas shale. Journal of Petroleum Science and Engineering, 74(3–4), 154–162.

Sun, J., Wang, X., Xiong, N., & Shao, J. (2018). Learning sparse representation with variational auto-encoder for anomaly detection. IEEE Access, 6, 33353–33361.

Sun, X., Sun, B., Zhang, S., Wang, Z., Gao, Y., & Li, H. (2018). A new pattern recognition model for gas kick diagnosis in deepwater drilling. Journal of Petroleum Science and Engineering, 167, 418–425.

Sundermeyer, M., Ney, H., & Schlüter, R. (2015). From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(3), 517–529.

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295–2329.

Tamim, N., Laboureur, D. M., Hasan, A. R., & Mannan, M. S. (2019). Developing leading indicators-based decision support algorithms and probabilistic models using Bayesian network to predict kicks while drilling. Process Safety and Environmental Protection, 121, 239–246.

Tang, A., Tam, R., Cadrin-Chênevert, A., Guest, W., Chong, J., Barfett, J., Chepelev, L., Cairns, R., Mitchell, J. R., & Cicero, M. D. (2018). Canadian Association of Radiologists white paper on artificial intelligence in radiology. Canadian Association of Radiologists Journal, 69(2), 120–135.

Tokhmchi, B., Memarian, H., & Rezaee, M. R. (2010). Estimation of the fracture density in fractured zones using petrophysical logs. Journal of Petroleum Science and Engineering, 72(1–2), 206–213.

Toreifi, H., & Rostami, H. (2014). New method for prediction and solving the problem of drilling fluid loss using modular neural network and particle swarm optimization algorithm. Journal of Petroleum Exploration and Production Technology, 4(4), 371–379.

Vega, M. P., Vieira, F. B., Fernandes, L. D., Freitas, M. G., Russano, E., & Martins, A. L. (2016). Smart monitoring and decision making for regulating annulus bottom hole pressure while drilling oil wells. Brazilian Journal of Chemical Engineering, 33(4), 969–983.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. Journal of Big Data, 3(1), 9.

Wikipedia. (2020). List of datasets for machine-learning research. In https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research .

Wu, S., Zhang, L., Fan, J., Zheng, W., & Zhou, Y. (2019). Real-time risk analysis method for diagnosis and warning of offshore downhole drilling incident. Journal of Loss Prevention in the Process Industries, 62, 103933.

Wu, S., Zhang, L., Zheng, W., Liu, Y., & Lundteigen, M. A. (2016). A DBN-based risk assessment model for prediction and diagnosis of offshore drilling incidents. Journal of Natural Gas Science and Engineering, 34, 139–158.

Xie, H., Shanmugam, A. K., & Issa, R. R. A. (2018). Big Data Analysis for Monitoring of Kick Formation in Complex Underwater Drilling Projects. Journal of Computing in Civil Engineering, 32(5), 4018030.

Yin, H., Si, M., & Wang, H. (2014). THE WARNING MODEL OF THE EARLY KICK BASED ON BP NEURAL NETWORK. Pakistan Journal of Statistics, 30(6).

Yu, Y., Liu, Q., Chambon, S., & Hamzah, M. (2019). Using Deep Kalman Filter to Predict Drilling Time Series. International Petroleum Technology Conference.

Zha, Y., & Pham, S. (2018). Monitoring downhole drilling vibrations using surface data through deep learning. In SEG Technical Program Expanded Abstracts 2018 (pp. 2101–2105). Society of Exploration Geophysicists.

Zhang, J., & Yin, S.-X. (2017). Fracture gradient prediction: an overview and an improved method. Petroleum Science, 14(4), 720–730.

Zhang, L., Wu, S., Zheng, W., & Fan, J. (2018). A dynamic and quantitative risk assessment method with uncertainties for offshore managed pressure drilling phases. Safety Science, 104, 39–54.

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. Information Fusion, 42, 146–157.

Chapter 3

3. A new methodology for kick detection during petroleum drilling using long short-term memory recurrent neural network

Preamble

This chapter addresses an objective of this dissertation as outlined in Section 1.3 which is exploiting how relevant features of drilling parameters can be utilized by deep learning algorithm for kick detection. The methodology presented in this section aims to improve early kick detection without false alarm which is vital for drilling safety, as presented in Section 1.1. and Section 1.6.

I (Augustine Uhunoma Osarogiagbon) have contributed to Conceptualization, Methodology, Formal Analysis, Software, Investigation, Writing - Original Draft, and Writing - Review & Editing of this work, while Somadina Muojeke's contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing; Dr. Ramachandran Venkatesan contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing, Supervision, and Project Administration; Dr. Faisal Khan contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing, Supervision, and Project Administration; and Dr. Paul Gillard contributed to Writing - Review & Editing, and Supervision. A version of this chapter has been published in the Journal of Process Safety and Environmental Protection, Volume 142, October 2020, Pages 126-137, https://doi.org/10.1016/j.psep.2020.05.046.

Abstract

Kick is a downhole phenomenon which can lead to blowout, and so early detection is important. In addition to early detection, the need to prevent false alarm is also useful in order to minimize wastage of operation time. A major challenge in ensuring early detection is that it increases the chances of false alarm. While several data-driven approaches have been used in the past, there is also ongoing research on the use of derived indicators such as d-exponent for kick detection. This chapter presents a data-driven approach which uses d-exponent and standpipe pressure for kick detection. The data-driven approach presented in this chapter serves as a complementary methodology to other stand-alone kick detection equations, and uses d-exponent and standpipe pressure as inputs.

This chapter proposes a methodology which uses the long short-term memory recurrent neural network (LSTM-RNN) to capture temporal relationships between time series data comprising of d-exponent data and standpipe pressure data with the aim of increasing the chances of achieving early kick detection without false alarm. The methodology involves obtaining the slope trend of the d-exponent data and the peak reduction in the standpipe pressure data for training the LSTM-RNN for kick detection. Field data is used for training and testing. Early detection is achieved without false alarm.

Keywords: Drilling; kick detection; machine learning; long short-term memory (LSTM); recurrent neural network (RNN); time series data

3.1 Introduction

Machine learning is an interesting field which involves finding relationship amongst data. There is a lot of interest in machine learning because of its success in challenging domains such as speech recognition, medical imaging, etc. (Alom et al., 2019). The level of success achieved in solving a task through the use of machine learning depends on the training data as well as the type of machine learning algorithm used. For example, while convolution neural network (CNN) is designed to take advantage of spatial relationship in two-dimensional data, recurrent neural network (RNN) is designed to explore temporal relationships in data (Alom et al., 2019).

During drilling for exploration/production of hydro-carbon, kick (i.e. influx of hydrocarbon into the well bore) can occur. This can be very dangerous especially if it is a gas influx, as this can lead to blowout when the gas influx is not properly controlled. Therefore, occurrence of kick is detected by locating sensors to monitor drilling parameters. Some parameters typically monitored for kick detection are torque, rate of penetration (ROP), weight on bit (WOB), flow differential and stand pipe pressure (SPP). Kick is typically indicated by sudden increase in torque, increase of outflow over inflow, decrease in SPP, increase in ROP and decrease in WOB. Equation 1 describes the d-exponent also known as the normalized rate of penetration. The d-exponent can be used as a means by which kick occurrence can be identified; this was demonstrated in the article by Mao *et al*.,(Y. Mao & Zhang, 2019) and by Tang *et al*., (Tang et al., 2019).

$$\text{d-exponent} = \frac{\log(\frac{ROP}{60N})}{\log(\frac{12WOB}{10^6 D_b})}, \tag{3.1}$$

where N refers to rotary speed and D_b refers to bit size.

Several researchers have utilized machine learning and a combination of different sensors for kick detection. This can be seen in Table 3.1.

Table 3.1 Listing of selected techniques for kick detection using machine learning that have been reported since 2001

Article source	Input parameters	Methodology
(Hargreaves et al., 2001)	Inflow and outflow	Bayesian classifier
(Nybo et al., 2008a)	Pump rate and mud density	Echo state network and physical model
(Nybo et al., 2008b)	Pump rate	Auto-regressive integrated moving average algorithm and echo state network
(Kamyab et al., 2010)	Active pit totalizer, suction pit, pump pressure (many parameters were tested but just these three detected kick)	Focused time-delay neural network
(Haibo et. al., 2014)	Casing pressure and SPP	Bayesian classifier
(Yin et al., 2014)	SPP, drilling time, cell volume, outlet flow rate, outlet conductivity, outlet volume density, outlet temperature, total hydrocarbon gas volume, C1 component content	Backpropagation neural network
(Pournazari et al., 2015)	Pit volume, SPP, flow-out and flow-in	Naïve Bayes, decision tree, random forest
(Liang et al., 2019)	Casing pressure and SPP	Genetic algorithm accelerated backpropagation neural network
(Alouhali et al., 2018)	Pressure gauges, flow meters, hook load, ROP, torque, pump rate, and WOB	Decision tree, k-nearest neighbor, sequential minimal optimization algorithm, artificial neural network and Bayesian network
(Xie et al., 2018)	ROP, mass per unit volume of drilling fluid (“Schlumberger definition”), mud weight (MW) going into the well, MW going out of the well, MW of circulating fluid and depth of well	Genetic wavelet neural network
(Fjetland et al., 2019)	Flow rate in, flow rate out, SPP, choke pressure, choke opening, bit pressure and bit depth	LSTM-RNN

In addition to the articles on machine learning approach for kick detection presented in Table 3.1, a systematic approach for kick detection was presented in the article by Vajargah & van Oort, (2015). The systematic approach shows how kick occurrence can be detected by observing flow in, flow out, pit gain, pump pressure and annular pressure profile.

Sensor data used for kick detection in several cases are time series in nature. When temporal relationship exist in sensor time series data, the event indicated at a time (kick or no kick) could be described as a function of the sensor reading both at that particular time, as well as sensor readings at earlier times in order to improve the accuracy of kick detection. For example, temporal relationship can be observed in Fig. 3 of the article by Hargreaves et al., (2001). This figure shows that flow out ramps up over a period of time in response to kick. We can observe that using the flow out value at an instant in time is not sufficient to indicate kick occurrence. Instead, several consecutive values of flow out will be needed in order to observe the trend or pattern of flow out over time for accurate kick detection.

Several dynamic neural network algorithms such as focused time delay neural network, classical recurrent neural network, recursive neural network can be utilised in learning temporal relationships in data. When it comes to learning long term temporal relationships in data, the long short-term memory recurrent neural network (LSTM-RNN) offers the advantage of overcoming the gradient vanishing problem which makes it difficult for other dynamic neural network algorithms to learn long term dependencies. The LSTM-RNN achieves this by having an architecture made up of memory cell, input gate, forget gate and output gate (Graves, 2012).

The objective of our work is to develop a methodology which can help achieve automatic, early and accurate detection of kick onset using d-exponent and standpipe pressure data. While effort has been made to develop equations, which can be used to detect kick as a function of d-exponent, standpipe pressure and flow measurement data, a lot of uncertainties are still encountered. For example in the article by Tang *et al.*, (Tang et al., 2019), kick indicating parameters (namely, flow-in rate, flow-out rate, SPP, WOB and ROP) were used in two equations. These equations are: (i) drilling parameter group (DPG) which is the same as d-exponent; (ii) flow parameter group (FPG) which utilises flow-in rate, flow-out rate, compressibility of the drilling mud (C_m) and SPP as shown by Equation 3.2.

$$FPG = flow\ out - \frac{flow\ in}{C_m \times (SPP - P_r)}, \tag{3.2}$$

where P_r represents reference pressure which is 14.7 psig at ground surface.

Also, in the article by Mao & Zhang, (2019), DPG, FPG and pit volume gain were used for kick detection. The methodology used in Tang et al., (2019) was also applied in Mao & Zhang, (2019).

Although FPG, DPG and pit volume gain can be utilised together for kick detection, some challenges such as: (i) the ratio by which the probability of kick occurrence using DPG, FPG and pit volume gain are combined for early kick detection with minimal chances of false alarm, (ii) the tolerance value of DPG, FPG and pit volume gain adopted for accurate kick detection, and (iii) the threshold value of kick-risk index (KRI) to indicate kick show areas for further research (Tang et al., 2019). The benefit of machine learning is that with sufficient training data, the machine learning algorithm can capture the relationship between input parameters/equations such as the ratio by which the inputs are combined for kick detection. In this chapter, data-driven approach using LSTM-RNN is implemented to utilize the relationship between SPP and d-exponent in order to achieve early detection of kick without false alarm. The industrial significance of the proposed methodology is that it can improve the chances of obtaining automatic, accurate, and early kick detection with sufficient training SPP and d-exponent data for different scenarios of kick and no-kick periods during drilling.

This chapter is structured as follows. Section 3.2 describes the proposed kick detection methodology. Section 3.3 describes the data used for verification of the methodology. Section 3.4 reports the result obtained by using the proposed methodology on the data described in Section 3.3. Concluding remarks are provided in Section 3.5.

3.2 METHODOLOGY

The flowchart of the proposed methodology is presented in Fig. 3.1. Detailed description of the methodology is presented in subsequent subsections.

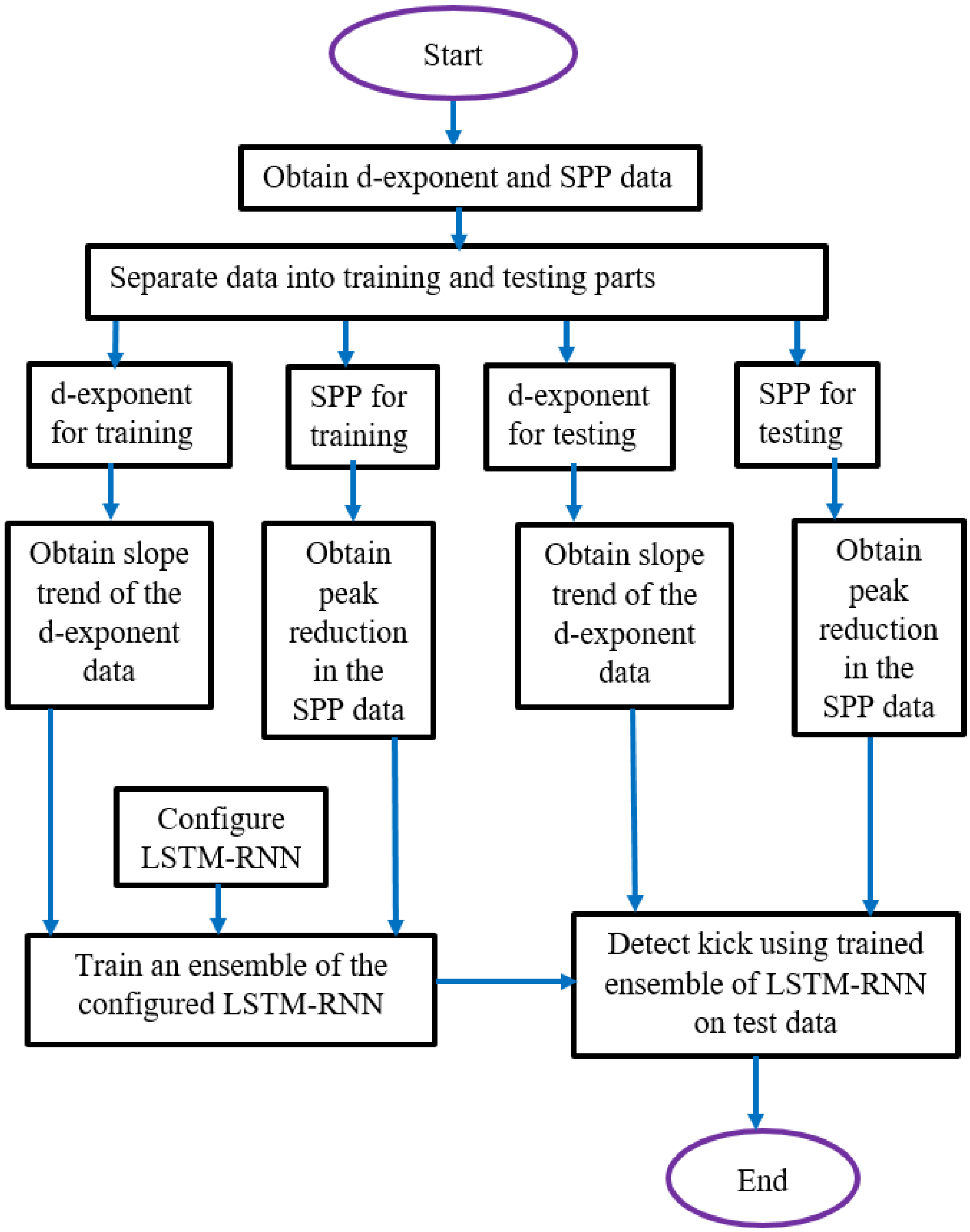

Fig. 3.1 Proposed data-driven methodology for kick detection

3.2.1 Obtaining peak reduction in SPP data

This process is made up of the following: (i) filter out noise, (ii) estimate change in trend of data, (iii) trap surgical changes in data, and (iv) normalize using minimum value (maximum absolute value).

3.2.1.1 Filtering out noise

A low pass filter can be used to filter out noise from the SPP data. A low pass Kaiser window finite impulse response type 1 filter with a discrete time cut-off frequency of 0.01π rad, stop band attenuation of 50 dB, and filter length of 51 is used. A simple moving average filter can also be used to achieve the same goal.

3.2.1.2 Estimating change in trend of data

Considering that the occurrence of kick causes reduction in SPP, the aim of this section is to quantify reduction in SPP at a point in time as a function of all previous SPP data obtained during drilling. Equation (3.3) shows how to obtain the change in trend data $SPPd_t$ at time t using the current SPP data SPP_t at time t and all previous SPP data.

$$SPPd_t = SPP_t - \frac{1}{t-1} \sum_{i=1}^{t-1} SPP_i. \tag{3.3}$$

3.2.1.3 Trapping surgical changes in data

For situations where reduction in SPP due to kick occurs in a transient manner, emphasis should be placed on the peak reduction in SPP in comparison to the other values of SPP which occurs after the onset of kick. The MATLAB code for implementing this is:

```
for i = 2:length(SPPd)
   if (SPPd(i) > SPPd(i − 1) )
      SPPd(i)  = SPPd(i − 1);
   end
end
```

$SPPd$ represents the change in trend of SPP for a case (a time series drilling scenario) using Equation 3.3.

3.2.1.4 Normalize data using minimum value (maximum absolute value)

Once the modified $SPPd$ is obtained for all training cases using the algorithm in section 3.2.1.3, the modified $SPPd$ values for all training cases are normalized by dividing them by the their minimum value. For example, if there are only two training cases A and B such that the modified $SPPd$ for training case A is 0, -2, -6, -7, -9 and the modified $SPPd$ for training case B is 0, -1, -2, -3; then the minimum $SPPd$ of all training cases will be -9. Therefore, the normalised $SPPd$ for case A would be 0, 2/9, 6/9, 7/9, 1 and that for case B would be 0, 1/9, 2/9, 3/9. The minimum value of all modified $SPPd$ training data is also used to normalize the modified $SPPd$ data for the testing case. This is because one does not know what the minimum modified $SPPd$ data for the testing case will be during field implementation.

3.2.2 Obtaining slope trend of d-exponent data

The process is made up of three parts which are (i) filter out noise, (ii) slope extraction by the use of a sliding window, and (iii) normalize data using mean and standard deviation of each case.

3.2.2.1 Filter out noise

Similar to in Section 3.2.1.1, a low pass filter for suppressing noise is also applied.

3.2.2.2 Slope extraction by the use of a sliding window

The slope of the d-exponent data with respect to time gives some information about kick occurrence during drilling (Tang et al., 2019). Therefore, the slope of the d-exponent data is used for kick detection analysis. The slope of the d-exponent data at every corresponding point in time is obtained using a sliding window. The size of the sliding window should be large enough to suppress random variations in the slope of the d-exponent data.

3.2.2.3 Normalize data using mean and standard deviation of each case

The d-exponents of each case to be used for training is normalised using their respective mean and standard deviation. For example, assume that case 2, case 3 and case 4 are used for training and case 1 is used for testing. If one represents the extracted slope of the d-exponent data of case 3 as $SD3$, the mean of $SD3$ as μ_{c3} and the standard deviation of $SD3$ as s_{c3}, then Equation 3.4 gives the normalised slope of the d-exponent data.

$$NSD3_t = (SD3_t - \mu_{c3})/s_{c3}. \tag{3.4}$$

$SD3_t$ represents the slope of the d-exponent data for case 3 at time t, where $NSD3_t$ represents the corresponding normalized value. In the same manner, the slopes of the d-exponent data of the other training cases are also normalized. For the test case, authors assume that they do not know what its final mean and standard deviation would be. In order to normalise the testing data, authors concatenated the extracted slope of the d-exponent of all the training cases and obtain the mean and standard deviation of the concatenated data. The mean and standard deviation are then used to obtain the normalised slope of the d-exponent of the test data.

3.2.3 Machine learning implementation

Although the LSTM-RNN is the main machine learning algorithm used in this work, the simple artificial neural network (ANN) which is also referred to as fully-connected neural network in some literature e.g. (D. Zhang et al., 2018) will also be tested. This can provide a justification for preferring the more sophisticated LSTM-RNN. This is similar to the approach utilised in the article by Zhang *et al.,* (Zhang et al., 2018), where simple ANNs and LSTM-RNNs were used to estimate missing log information, and the LSTM-RNN performed better than the simple ANN implementation. This section is made up of four parts which are:

(i) introduction to simple ANN, (ii) introduction to LSTM-RNN, (iii) configuration of the LSTM-RNN and simple ANN, and (iv) training of an ensemble of the LSTM-RNNs and an ensemble of the simple ANNs.

3.2.3.1 Simple ANN

A simple ANN is typically made up of the input layer, one or more simple hidden layers and an output layer. Fig. 3.2 describes the simple ANN used in this work.

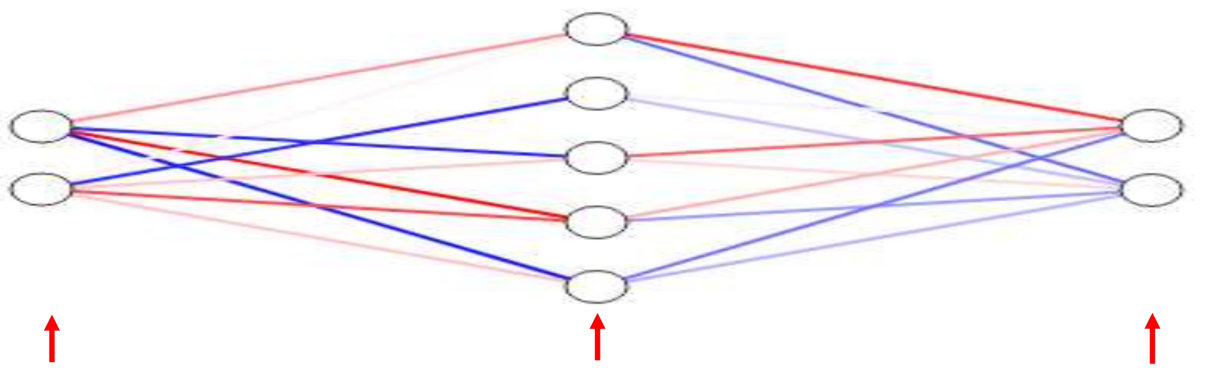

Fig. 3.2 A simple ANN (diagram drawn with the aid of the website (Lenail, 2019))

The input layer is given two selected inputs: slope trend of d-exponent data and peak reduction in SPP data. Fig. 3.2 shows the hidden layer with 5 nodes. Actual implementation could require much larger number of nodes in this layer in order to achieve effective training. The output layer is equipped with softmax function, and has two nodes because the task involves classifying the output as either kick or no-kick, with one of the nodes responsible for evaluating the probability of having a kick event while the other node is responsible for evaluating the probability of having no-kick event. Therefore, during detection, the output node with the higher probability dictates if the event is a kick or no-kick. Data reaching the nodes of both hidden and output layers is obtained by multiplying the data from the preceding layers with weight values shown by the link lines in the figure. These data reaching a node in these layers are then summed up and passed through an activation function to get the corresponding output of that node. The link lines are shown in different shades of colours in the figure in order to indicate variability in weight values which typically occur amongst the weights of a neural network implementation. For example red colours represent positive weights values, blue colours represent negative weight values, and the lightness of the colour indicates the magnitude of the weight. Each node also has a trainable bias input value. More information on ANNs can be found in Hagan et al., (1996).

3.2.3.2 LSTM-RNN

The main difference between the LSTM-RNN implementation and the simple ANN implementation in this work is that the hidden layer of the simple ANN implementation is replaced by an LSTM layer for the LSTM-RNN implementation. Before presenting the hidden layer (LSTM layer) of a LSTM_RNN implementation, let us first consider the hidden layer of a simple RNN implementation. This difference between the simple ANN and simple RNN can be observed by considering one of the nodes in the hidden layer of Fig 3.2. Fig. 3.3 compares a node in the hidden layer for simple ANN implementation shown in Fig. 3.3a and for a simple RNN implementation shown in Fig. 3.3b.

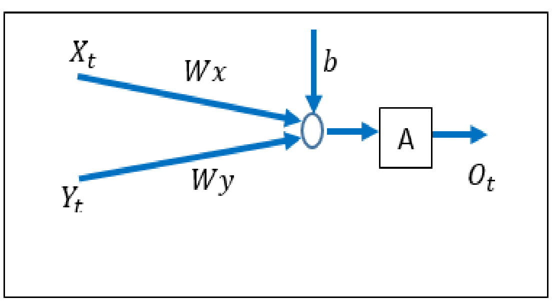

Fig. 3.3a

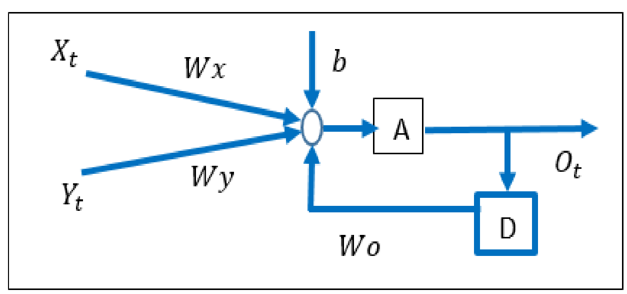

Fig. 3.3b

Fig. 3.3 A hidden layer node in simple ANN and RNN a. ANN b. RNN

In Fig. 3.3, Wx, Wy and Wo are neural network weights, b represents bias, X_t and Y_t represents inputs, O_t represents output, A represents activation function and D is a time delay function.

For the simple ANN implementation, output (O_t) at time t is simply a function of the two inputs (X_t , Y_t) at time t as shown by Equation 3.5. For the simple RNN, the output at time t is not only a function of the inputs at time t, but also a function of the output at time $t-1$, as shown by Equation 3.6. The output at time $t-1$ is achieved by passing the output through a time delay function, D. In equations (3.5) and (3.6), $f\{z\}$ represents the application of an activation function such as hyperbolic tangent to z.

$$O_t = f\{(X_t \times Wx) + (Y_t \times Wy) + b\}, \tag{3.5}$$

$$O_t = f\{(X_t \times Wx) + (Y_t \times Wy) + b + (O_{t-1} \times Wo)\}. \tag{3.6}$$

From Equation (3.6), it can be observed that O_{t-1} is also a function of X_{t-1}, Y_{t-1} and O_{t-2}. This shows that the output from the node in RNN implementation can capture both the present input values and all past input values to the node. This gives the simple RNN an advantage over the simple ANN implementation when temporal relationship occurs between the output and input data at different time steps. Although the simple RNN is designed to capture temporal relationships in data, the problem of vanishing gradient limits this capability when a series data is very long. The article by Hochreiter et al., (Hochreiter & Schmidhuber, 1997), gives a detailed explanation of how the LSTM implementation overcomes this challenge. A simple diagrammatic description of an LSTM unit working with 1 input is shown by Fig. 3.4.

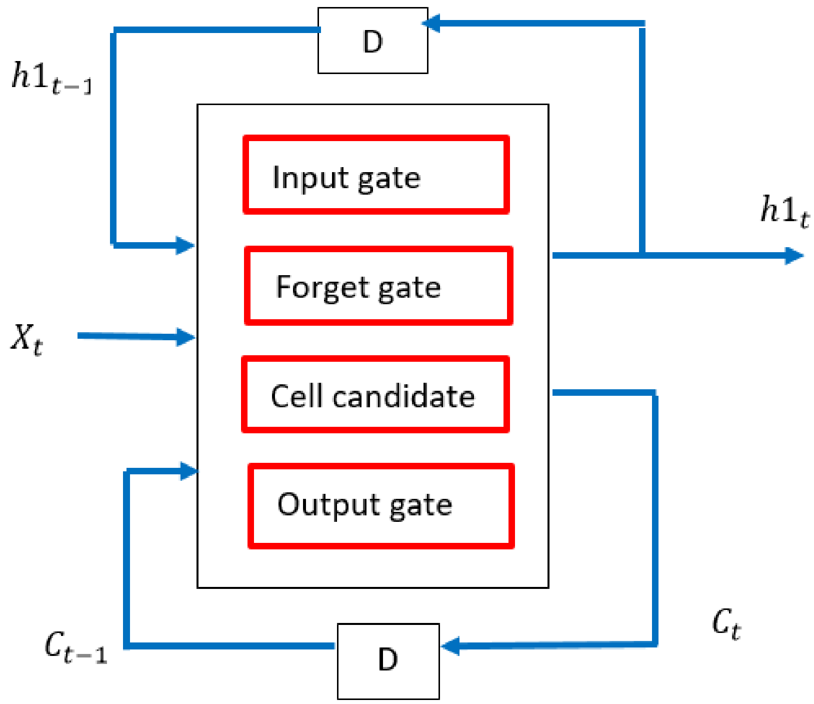

Fig. 3.4   LSTM unit

For the LSTM shown in Fig. 3.4, the output O_t at time t is a function of the output gate and memory cell value C_t at time t. The memory cell value at time t is a function of the input gate, forget gate, cell gate/candidate and previous memory cell value C_{t-1}. Each of the four gates operates as a function of the current input value X_t and the previous output value O_{t-1}. Equations (3.7) to (3.10) summarise this (Hochreiter & Schmidhuber, 1997; D. Zhang et al., 2018).

$$Gi_t = F_s\{(X_t \times wi) + (O_{t-1} \times vi) + bi\}, \tag{3.7}$$

$$Gf_t = F_s\{(X_t \times wf) + (O_{t-1} \times vf) + bf\}, \tag{3.8}$$

$$Gc_t = F_h\{(X_t \times wc) + (O_{t-1} \times vc) + bc\}, \tag{3.9}$$

$$Go_t = F_s\{(X_t \times wo) + (O_{t-1} \times vo) + bo\}, \quad (3.10)$$

where Gi_t refers to control gate, Gf_t refers to forget gate, Gc_t refers to cell candidate or gate, Go_t refers to output gate. wi, wf, wc, wo, vi, vf, vc and vo are weights associated with the respective gates, while bi, bf, bc and bo are the corresponding bias values for their respective gates. It should be noted that F_s and F_h are sigmoid (logistic) and hyperbolic tangent activation functions, as shown by Equations (3.11) and (3.12).

$$F_s\{z\} = \frac{1}{1 + e^{-z}}, \quad (3.11)$$

$$F_h\{z\} = \tanh(z). \quad (3.12)$$

Equation (3.13) shows how the memory cell value is updated as a function of its previous value, the forget gate, the input gate and the cell gate.

$$C_t = (Gf_t \times C_{t-1}) + (Gc_t \times Gi_t). \quad (3.13)$$

The output from the LSTM unit is shown in Equation (3.14) as a function of output gate and memory cell value.

$$O_t = Go_t \times F_h\{C_t\}. \quad (3.14)$$

With the introduction of the LSTM, improvement has been achieved in comparison to traditional RNN in a number of applications. An example of this is language modeling by Sundermeyer *et al*., (Sundermeyer et al., 2015).

3.2.3.3 Configuration of the LSTM-RNN and simple ANN

A large number of hidden layer weights can help the neural network capture more information (Hagan et al., 1996). This observation motivates the use of a large number of LSTM units for LSTM-RNN implementation and a large number of hidden layer nodes for simple ANN implementation. However, more weights will result in the requirement of more training time or computing resources as well as the possibility of poor training due to overfitting (Hagan et al., 1996). In this chapter, the number of LSTM hidden layer units and the number of hidden layer nodes for simple ANN were chosen to sufficiently capture the learnable features of the data. Table 3.2 shows the configuration of the neural networks implemented.

Table 3.2 Configuration of the LSTM-RNN and simple ANN

	LSTM-RNN	Simple ANN
Input layer	2 inputs for sequence data	2 inputs for sequence data
Hidden layer	50 LSTM units	1000 nodes with hyperbolic tangent activation function
Output layer	2 nodes with softmax activation function	2 nodes with softmax activation function

3.2.3.4 Training an ensemble of LSTM-RNNs and an ensemble of simple ANNs

Several factors, such as the optimization algorithm used, number of training epochs, regularization method etc., can influence the learning performance as well as training time of the neural network implemented (Alom et al., 2019). The Adam optimization algorithm was used for training, although other optimization algorithms such as RMSProp could also be successfully used for training, Adam was chosen simply because it combines the advantages of RMSProp (good for handling cases with on-line settings) and AdaGrad (good for handling sparse gradients) (Kingma & Ba, 2014). In order to prevent error due to exploding gradient, the gradient errors could be clipped at an absolute maximum value of 15, based on a case study in the PhD dissertation by Mikolov (Mikolov, 2012). Because the task is to perform binary classification, rather than regression, the optimization algorithm aimed at minimizing cross entropy error rather than root mean square error of the training data (Bishop, 2006). Table 3.3 summarises the approach used for training the configured LSTM-RNN and simple ANN.

Table 3.3 Training the LSTM-RNN and simple ANN algorithms

Training options	Method/Value used	Reason
Optimization algorithm	Adam	A recent and robust method (Alom et al., 2019; Kingma & Ba, 2014).
Regularization method	L2 Regularization	Based on the articles (Alom et al., 2019; Kingma & Ba, 2014)
Factor for L2 Regularization	0.0004	Based on the article (Alom et al., 2019).
Gradient threshold method	L^2 norm	Based on the articles (Kingma & Ba, 2014; Pascanu et al., 2013).
Gradient threshold value	15	Inspired by the work (Mikolov, 2012).
Maximum Epochs	50	Sufficient for learning the test data.
Initial learning rate	0.001	Based on the article (Kingma & Ba, 2014).
β_1	0.9	Based on the article (Kingma & Ba, 2014).
β_2	0.999	Based on the article (Kingma & Ba, 2014).
ϵ	0.00000001	Based on the article (Kingma & Ba, 2014).
Training type	Offline training	

Weights were initialized using Gaussian distribution with a mean value of 0 and a standard deviation value of 0.01. The bias values were initialised to 0. Variations can occur in terms of detection time for the same RNN-LSTM configuration and training data when training is done at different point in time. This is due to the random initialization of the parameters of the network. The mode of learning/training implemented for ANN and LSTM-RNN of this work is supervised learning.

An ensemble of LSTM-RNNs was used because random errors from several classifiers can be suppressed by averaging (Anifowose et al., 2017). Based on this, a committee (or an ensemble) of thirty independent LSTM-RNNs of the same configuration were used for training; and detection was done by considering the mode output (majority decision) of the committee of LSTM-RNNs at each point in time. Similarly, an ensemble of thirty independent simple ANNs with the same configuration was also used for training and detection. MATLAB R2018b and R2019a were used for the simulations.

3.3 Data used for verifying methodology

The data used for this study were obtained from the article by Tang *et al.*, (Tang et al., 2019). The smoothed DPG (d-exponent) data and SPP data of Fig. 6, Fig. 8, Fig. 10 and Fig. 12 (representing cases 1, 2, 3 and 4,) in Tang et al., (2019) were obtained. This was done by reverse engineering the figures using *WebPlotDigitizer* and *UN-SCAN-IT* software. These software have been used in some articles for data extraction and analysis (Mani et al., 2018; Phattanawasin et al., 2016). The figures obtained through reverse engineering resemble their original counterparts. While it can be argued that the data obtained through reverse engineering will have some errors, the ability of an algorithm to learn and perform detection despite the imperfection in the data obtained could be seen as a plus for the algorithm; this is comparable to having a machine learning method that is developed with regularization capabilities to enhance learning with noisy data (Foresee & Hagan, 1997). The data obtained was processed to have a time axis resolution of 1 second. Fig. 3.5 to Fig. 3.12 shows the SPP and d-exponent data obtained for case 1 to 4.

Fig. 3.5 SPP data for case 1

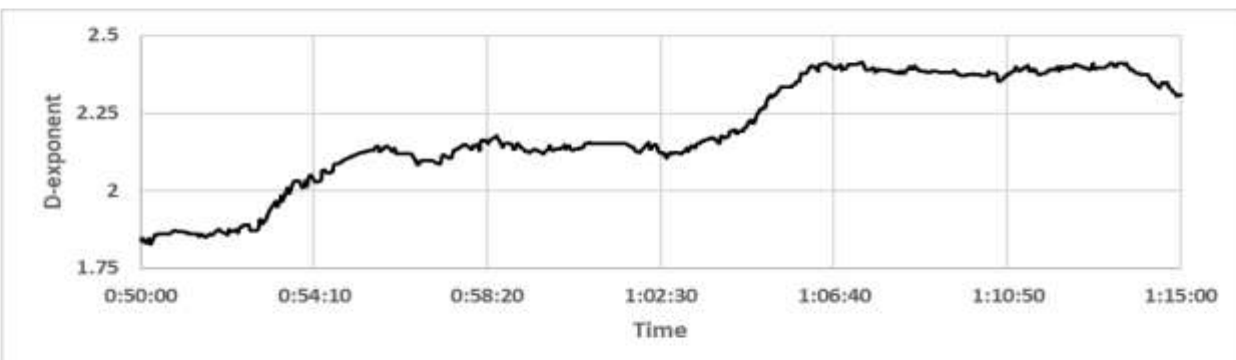

Fig. 3.6 D-exponent for case 1

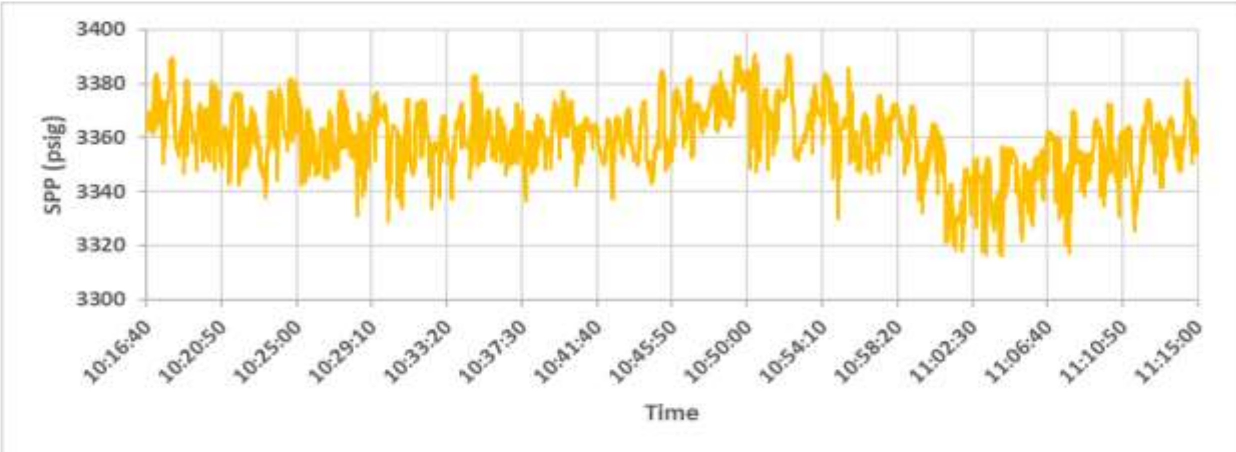

Fig. 3.7 SPP data for case 2

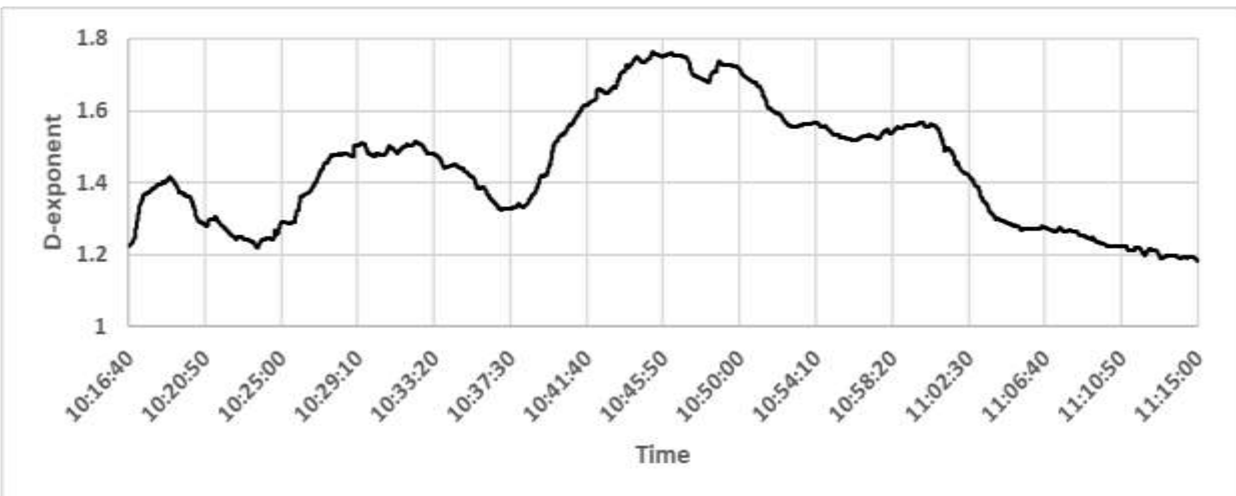

Fig. 3.8 D-exponent for case 2

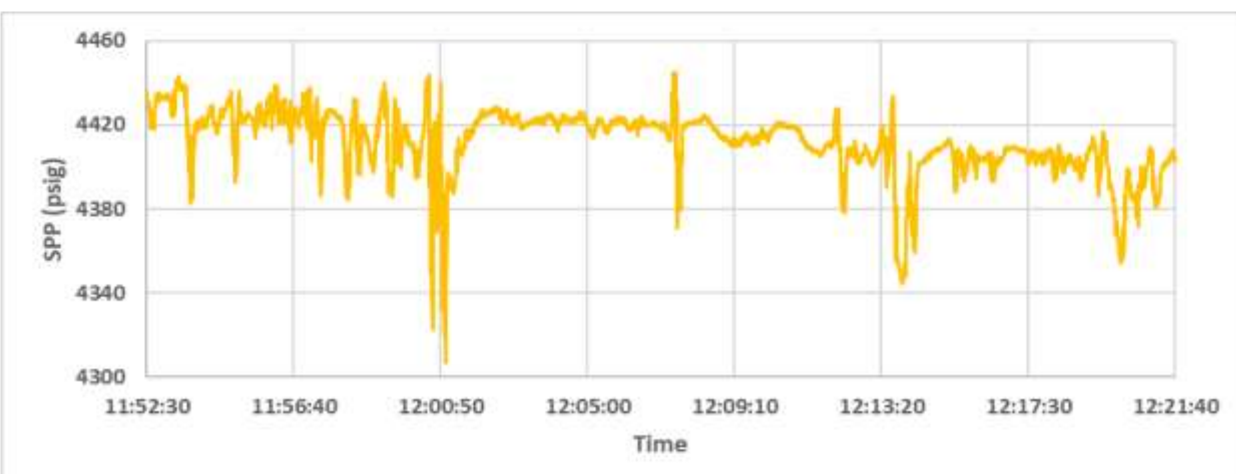

Fig. 3.9 SPP data for case 3

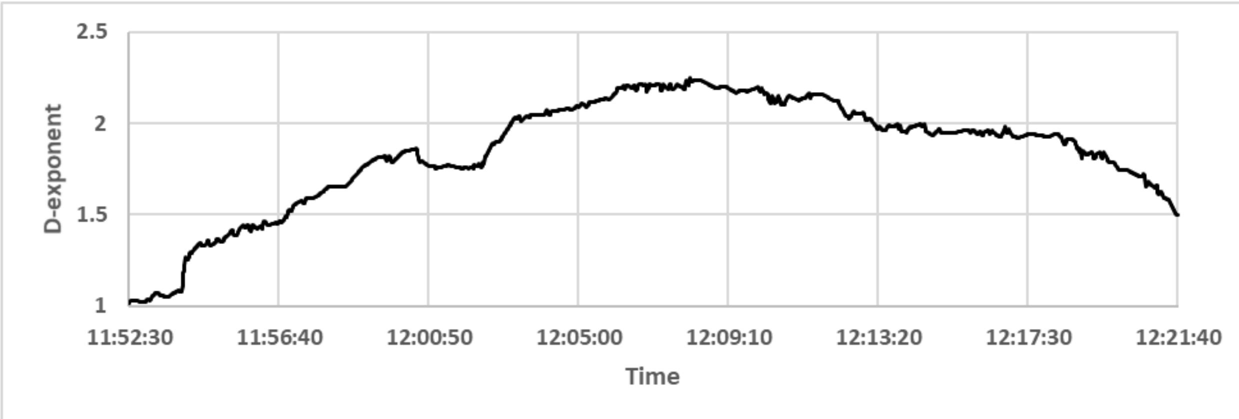

Fig. 3.10 D-exponent for case 3

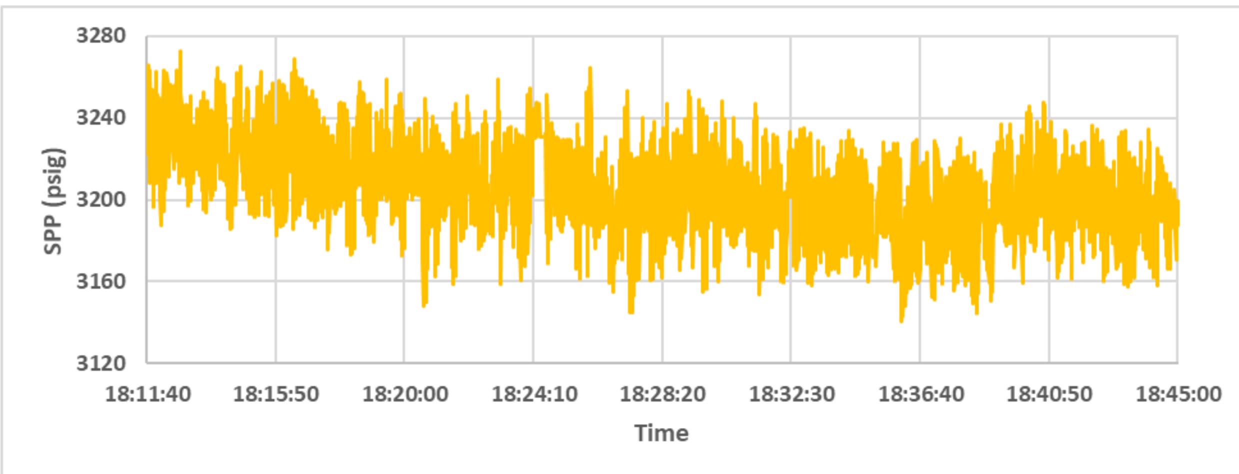

Fig. 3.11 SPP data for case 4

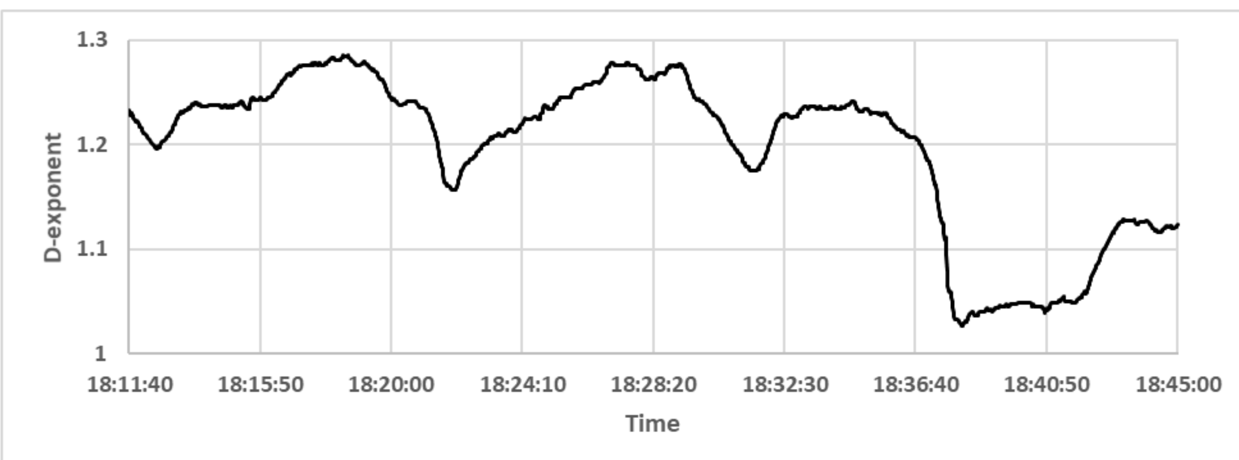

Fig. 3.12 D-exponent for case 4

3.3.1 Estimated time of kick occurrence

The task is defined as a binary classification problem, and so the output data event is described as either kick or no-kick. Because the ANN and LSTM-RNN to be implemented will undergo supervised learning, the output/target data will be specified as a kick or no-kick event at every point in time for which we have SPP and d-exponent data. The no-kick event occurs from start time (based on the drilling data) till the time just before kick onset, while the kick event occurs from the point of kick onset till the end of data collected; this can be seen in the figures in Section 3.4.

Table 2 in the article Tang et al., (2019) shows the estimated kick onset time, which is obtained by observing the trend in the DPG (d-exponent) data. This kick onset time given in Table 2 of Tang et al., (2019) has a resolution of 1 min. Because the SPP and d-exponent data obtained through reverse engineering has a resolution of 1 second, the d-exponent data is further scrutinized to obtain the time in seconds when the kick is most likely to have occurred. This is done by: (i) considering the d-exponent data points between -30 seconds and +30 seconds of the kick onset time given by Table 2 of Tang et al., (2019), (ii) obtaining the time with the maximum d-exponent value within the region of consideration, and (iii) choosing the next time value after the time with the maximum d-exponent value as the kick onset time. This is because the kick onset is estimated as the time when the d-exponent data begins to deviate from the previously noticed increasing trend (Tang et al., 2019). Table 3.4 gives the kick onset time from Tang et al., (2019) and the higher resolution kick onset time obtained by further observing the d-exponent data.

Table 3.4 Actual kick onset time from the article Tang et al., (2019) and the higher resolution actual kick onset time obtained by close scrutiny of the d-exponent data

Case	Kick onset time from Tang et al., (2019) in hours and minutes	Kick onset time by further scrutiny of the d-exponent data in hours, minutes and seconds
1	01:06	01:06:29
2	10:52	10:51:31
3	12:07	12:06:41
4	18:37	18:36:41

3.3.2 An example of obtaining peak reduction in SPP data

The figures showing peak reduction in SPP data assuming case 1 is to be used for testing and cases 2, 3 and 4 used for training are provided as supplementary data in Appendix A (Section 3.6). Fig. 3.13 clearly shows relatively high values of the peak reduction in the SPP data after kick onset time (01:06:29, as shown in Table 3.4 for case 1).

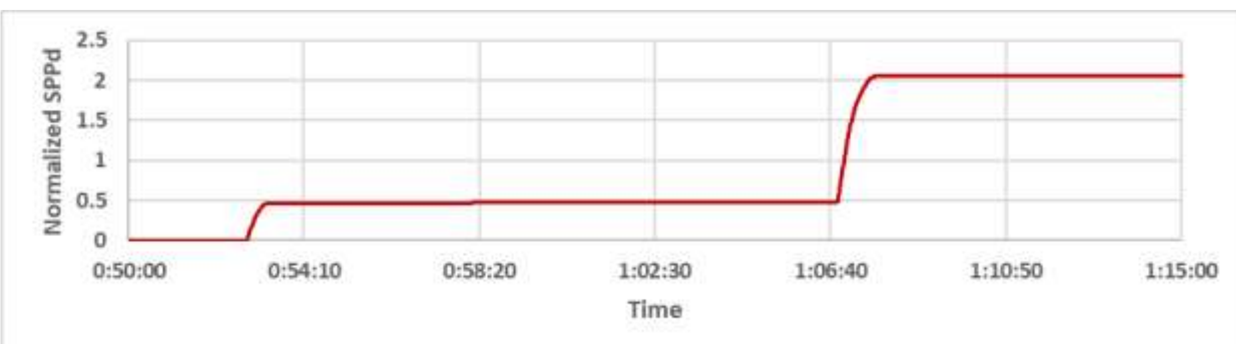

Fig. 3.13 Peak reduction in SPP data for case 1. In this scenario, case 2, 3 and 4 are for training while case 1 is for testing

3.3.3 An example of obtaining the slope trend of the d-exponent data

The figures showing the slope trend of the d-exponent data assuming case 1 is to be used for testing and cases 2, 3 and 4 used for training are provided as supplementary data in Appendix B (Section 3.7). It should be noted that the d-exponent data obtained has noisy variations already suppressed. This was done by using a median filter with a kernel size of 51 (Tang et al., 2019). Thus, the step described in section 3.2.2.1 (Filtering) is skipped. The methodology implemented in Tang et al., (2019) utilized long sliding window of 3 minutes, 5 minutes and 1 minute; this gives an indication that window sizes within this range can be utilized for the d-exponent data obtained. Therefore, sliding window sizes of 2 minutes, 3 minutes and 4 minutes are used for obtaining the slope of the d-exponent data. Fig. 3.14 shows that higher slope values of d-exponent data are likely to be in the region of no kick.

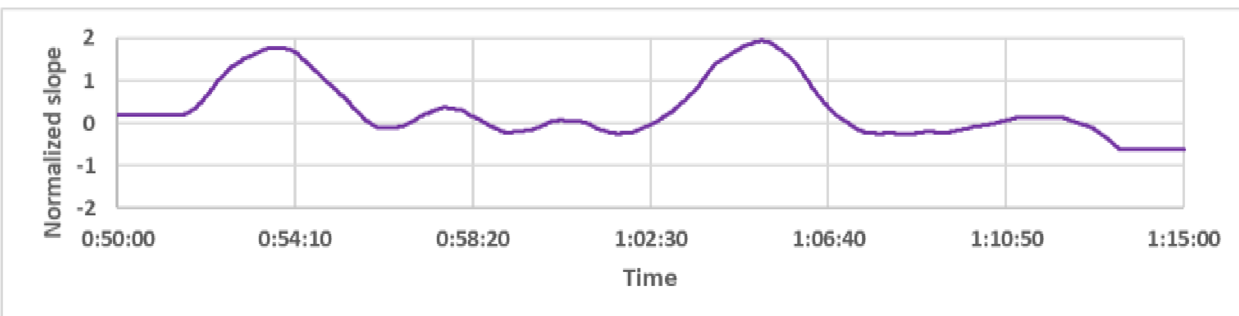

Fig. 3.14 Slope trend of d-exponent data for case 1 (using sliding window of 3 minutes). In this scenario, cases 2, 3 and 4 are for training while case 1 is for testing

3.4 Results and discussion

Considering that data corresponding to four cases are available, four categories of training and testing will be done. Category 1 involves using all data points/samples of case 2, case 3 and case 4 to train while all data points/samples of case 1 is used for testing. Category 2 involves training with all data points/samples of cases 1, 3 and 4 while all data points/samples of case 2 is used for testing. Category 3 involves training with all data points/samples of cases 1, 2 and 4 while all data points/samples of case 3 is used for testing. Category 4 involves training with all data points/samples of cases 1, 2 and 3, while all data points/samples of case 4 is used for testing. In addition, the final part of this section (section 3.4.5) compares the result obtained with the proposed methodology to that of published methodology. All figures of section 3.4 are a plot of kick/no-kick event vs time. In the vertical axis of the figures of section 3.4, the value 1 represents kick and 0 represents no-kick event.

3.4.1 Category 1 testing

Fig. 9 and Fig. 10 provide a graphical illustration of the performance of the methodology. Fig. 3.15 represents the actual events that are of interest to detect and Fig. 3.16 shows detection with the LSTM-RNN implementation and sliding window size of 4 minutes for obtaining the slope of the d-exponent. Fig. 3.16 shows that the algorithm can detect the kick event to a good degree after a short delay.

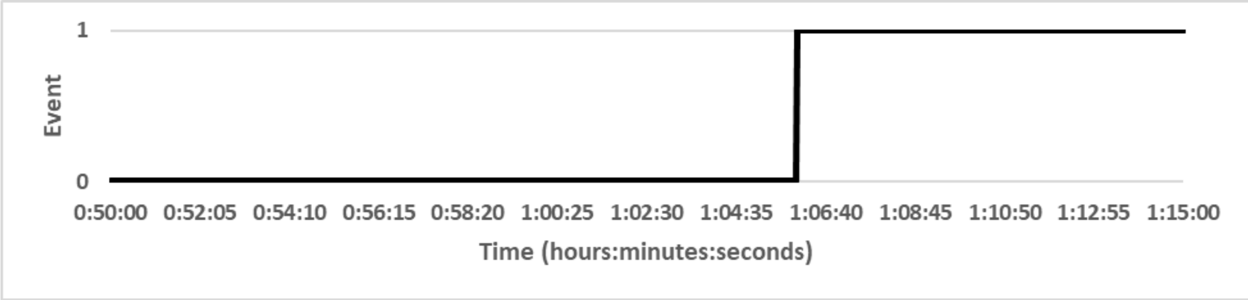

Fig. 3.15 Actual kick or no-kick event for case 1

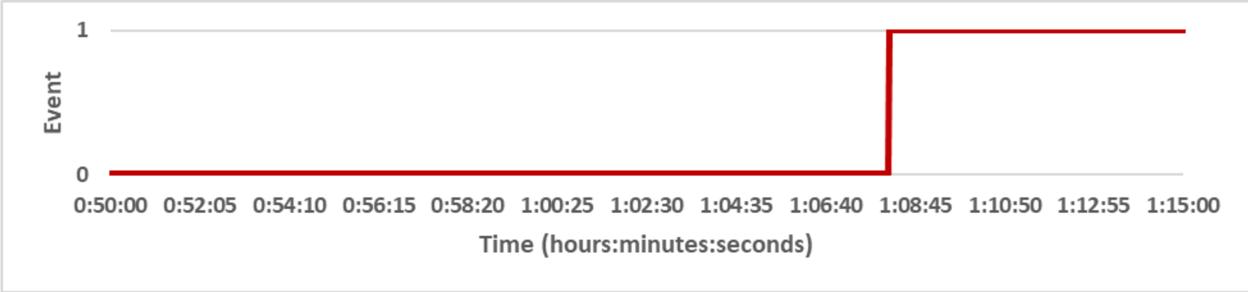

Fig. 3.16 Kick onset detection for case 1 using LSTM-RNN with peak reduction in standpipe pressure data and slope trend of d-exponent data (using sliding window size of 4 minutes) as input

Table 3.5 Kick onset time for category 1 testing

	Actual kick	Algorithm / window size for computing slope of d-exponent			
		LSTM-RNN / 2 minutes	LSTM-RNN / 3 minutes	LSTM-RNN / 4 minutes	Simple ANN / 3 minutes
Onset time	01:06:29	01:08:05	01:08:09	01:08:09	01:06:53
Delay in detection relative to kick occurrence time		00:01:36	00:01:40	00:01:40	00:00:24

Table 3.5 shows that the LSTM-RNN and simple ANN detected kick for the d-exponent sliding window sizes considered without false alarm. The results also show that detection time was very similar for the different sliding window sizes considered for the LSTM-RNN methodology.

3.4.2 Category 2 testing

Fig. 3.17 shows the actual kick and no-kick events and Fig. 3.18 shows detection with the LSTM-RNN implementation and sliding window size of 4 minutes for obtaining the slope of the d-exponent. Fig. 3.18 shows that the methodology proposed can detect the events to a good degree with insignificant delay for the sliding window size of 4 minutes.

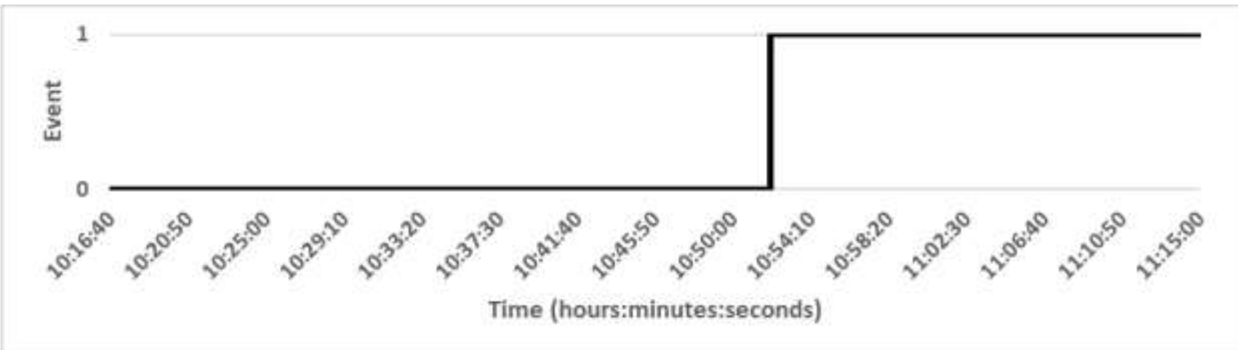

Fig. 3.17 Actual kick or no-kick event for case 2

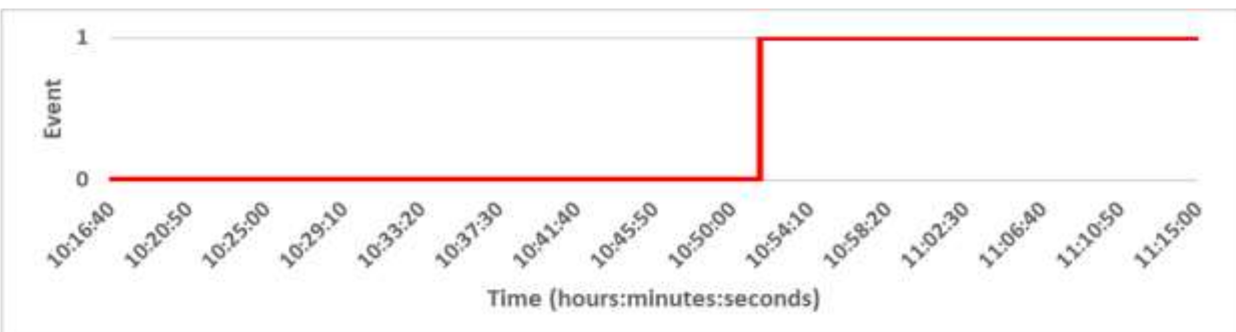

Fig. 3.18 Kick onset detection for case 2 using LSTM-RNN with peak reduction in standpipe pressure data and slope trend of d-exponent data (using sliding window size of 4 minutes) as input

Table 3.6 Kick onset time for category 2

	Actual kick	Algorithm / window size for computing slope of d-exponent			
		LSTM-RNN / 2 minutes	LSTM-RNN / 3 minutes	LSTM-RNN / 4 minutes	Simple ANN / 3 minutes
Onset time	10:51:31	11:01:52	10:52:42	10:51:32	11:01:20
Delay in detection relative to kick occurrence time		0:10:21	0:01:11	0:00:01	0:09:49

Table 3.6 shows that the LSTM-RNN and simple ANN detect kick for the d-exponent sliding window sizes considered without false alarm. Although the use of different sliding window sizes still resulted in successful detection, there is a significant variation between detection time achieved between the 2 minutes sliding window and the 4 minutes sliding window. A possible reason for this is that while the d-exponent shows a decreasing trend at around 10:52 AM, the SPP data shows decreasing trend at around 11:02 AM (Tang et al., 2019).

3.4.3 Category 3 testing

Fig. 3.19 shows the actual kick and no-kick events and Fig. 3.20 shows detection with the LSTM-RNN implementation and sliding window size of 4 minutes for obtaining the slope of the d-exponent. Fig. 3.20 also shows that the algorithm can detect the events to a good degree although with some delay. Table 3.7 shows that the LSTM-RNN and simple ANN detected kick for d-exponent sliding window sizes considered without false alarm.

Fig. 3.19 Actual kick or no-kick event for case 3

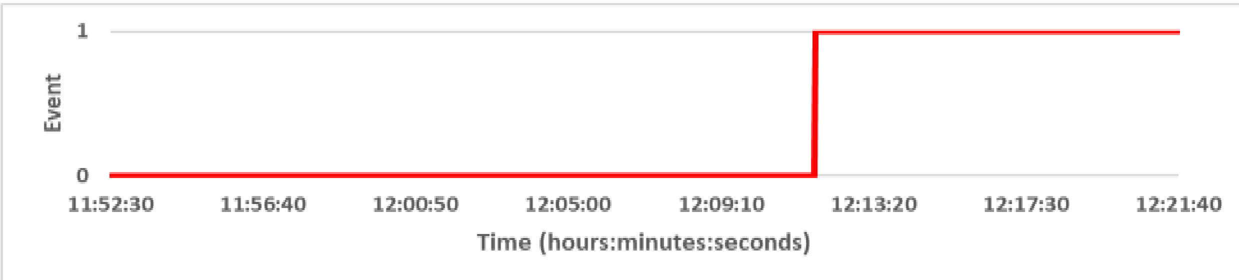

Fig. 3.20 Kick onset detection for case 3 using LSTM-RNN with peak reduction in standpipe pressure data and slope trend of d-exponent data (using sliding window size of 4 minutes) as input

Table 3.7 Kick onset time for category 3

	Actual kick	Algorithm / window size for computing slope of d-exponent			
		LSTM-RNN / 2 minutes	LSTM-RNN / 3 minutes	LSTM-RNN / 4 minutes	Simple ANN / 3 minutes
Onset time	12:06:41	12:12:20	12:12:16	12:11:45	12:07:58
Delay in detection relative to kick occurrence time		00:05:39	00:05:35	00:05:04	00:01:17

3.4.4 Category 4 testing

Fig. 3.21 shows the actual kick and no-kick events and Fig. 3.22 shows detection with the LSTM-RNN implementation and sliding window size of 4 minutes for obtaining the slope of the d-exponent. Fig. 3.23 shows multiple false alarm when simple ANN is used as opposed to the nearly perfect detection with LSTM-RNN shown in Fig. 3.22. Table 3.8 also shows that the LSTM-RNN detected kick for the sliding window sizes considered without false alarm. Unfortunately, the simple ANN had multiple false alarms. Based on this, several configurations were tested for simple ANN implementation. Also, different window sizes for extracting slope of d-exponent data were also tried, but all resulted in false alarm. These results are summarized in Table 3.9.

Fig. 3.21 Actua l kick or no-kick event for case 4

Fig. 3.22 Kick onset detection for case 4 using LSTM-RNN with peak reduction in standpipe pressure data and slope trend of d-exponent data (using sliding window size of 4 minutes) as input

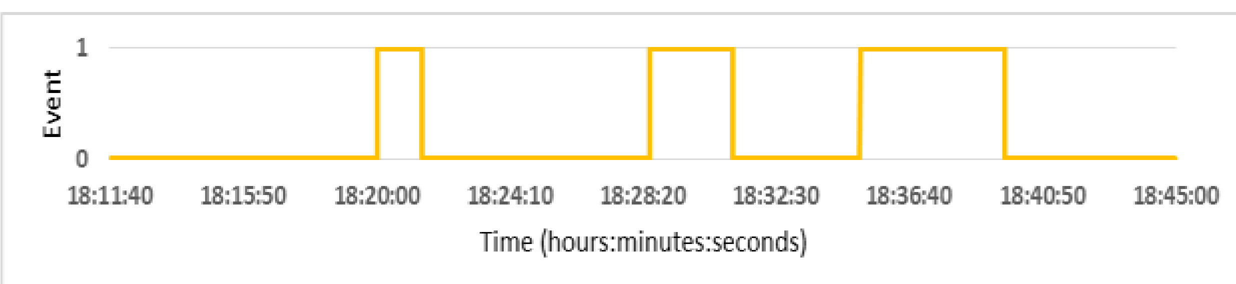

Fig. 3.23 Kick onset detection for case 4 using simple ANN with peak reduction in standpipe pressure data and slope trend of d-exponent data (using window size of 4 minutes) as input.

Table 3.8 Kick onset time for category 4

	Actual kick	Algorithm / window size for computing slope of d-exponent			
		LSTM-RNN / 2 minutes	LSTM-RNN / 3 minutes	LSTM-RNN / 4 minutes	Simple ANN / 3 minutes
Time	18:36:41	18:37:18	18:37:02	18:36:42	False alarm
Delay in detection relative to kick occurrence time		00:00:37	00:00:21	00:00:01	

Table 3.9 Summary of different trials for case 4 using different simple ANN configurations

Number of nodes in first hidden layer	Number of additional hidden layers with 1000 nodes	False alarm(s) occurred?	Time of first false alarm occurrence	Sliding window size for d-exponent slope extraction
100	None	Yes	18:20:36	3 minutes
1000	None	Yes	18:20:33	3 minutes
5000	None	Yes	18:28:24	3 minutes
10,000	None	Yes	18:28:37	3 minutes
1000	1	Yes	18:28:36	3 minutes
1000	3	Yes	18:28:25	3 minutes
1000	None	Yes	18:20:58	2 minutes
1000	None	Yes	18:20:02	4 minutes

3.4.5 Validation of the developed methodology

Table 3.10 compares the results obtained using the proposed methodology (using LSTM-RNN and a sliding window length of 3 minutes) with those in Table 2 of Reference Tang et al., (2019) (also using 3-minute long sliding window). It should be noted that the aim of the comparison is not to claim superiority of methodology but to show that the methodology presented here can complement the methodology presented in Mao & Zhang, (2019) and Tang et al., (2019). The methodology of Mao & Zhang, (2019) and Tang et al., (2019) can be used without having a database for training, whereas the methodology presented here is a data driven approach.

Table 3.10 Comparison of results: proposed methodology vs Tang et al. (Tang et al., 2019)

Case Number / Category	Kick occurrence time	Methodology of Tang et al., (2019) (Time only available in hours and minutes)	Proposed methodology using LSTM-RNN
1	01:06:29	01:10	01:08:09
2	10:51:31	11:05	10:52:42
3	12:06:41	12:11	12:12:16
4	18:36:41	18:41	18:37:02

Although the delay in kick detection for case 3 using our proposed methodology was between 5 and 6 minutes, the reservoir influx was observed at 12:21 (equivalent to a delay of about 14 minutes) was 11 bbl or 1.75 m^3 (Tang et al., 2019). Hence, the reservoir fluid influx for case 3 using our proposed methodology was less than 2 m^3, and this is still safe based on the guidelines in the article by Yin et al., (2019).

The performance of the proposed LSTM-RNN methodology depends on the quantity of training data. More training data will offer the LSTM-RNN a better opportunity to capture the relationship between input parameters required to detect kick. Although it may be suggested that only SPP should be considered as the input in order to achieve faster kick detection, test with only SPP input (neglecting d-exponent data) resulted in false alarm in 2 of the 4 cases. This shows the benefit of considering multiple independent input sources.

One of the extra requirements of using a data driven approach is the requirement of training time and software/hardware to perform training. The training time depends on several factors such as computational strength (e.g. number of processing cores/units), size of data etc. For example, the training time achieved by using Nvidia GeForce GTX 1050 through MATLAB R2018b is shown in Table 3.11.

Table 3.11 Training time for the different categories

Case Number / Category	Average training time for an LSTM-RNN in seconds
1	8
2	5
3	8
4	8

The training time values in Table 3.11 need to be multiplied by 30 to get total training time, this is because we utilized an ensemble of thirty identical LSTM-RNNs for a category. This results in about 240 seconds (4 minutes) for each of three categories of training and 150 seconds for one category. While this training time may not be too high, the value could increase when more training data is available.

Although the work presented here utilized data driven approach for d-exponent and SPP, further work could involve utilizing data driven approach for d-exponent and FPG, considering that FPG utilizes both SPP and flow measurements as shown by Equation 3.2. Although the data in Tang et al., (2019) may be used for this, the flow paddle malfunctioned in one of the cases thereby limiting the available cases from 4 to 3. One of the draw backs of data driven approach is that it depends on the availability of data. However, once sufficient training data is available, the data driven approach provides a valuable means of complementing direct equation approach. The Article Mao & Zhang, (2019) indicates the availability of data from 15 wells with 24,546 drilling hours of 23 kick events. The data driven methodology proposed in this work can be used for the data in Mao & Zhang, (2019). This is because Mao & Zhang, (2019) contains more drilling time data and kick events in comparison to Tang et al., (2019) and the performance of data driven methodology especially with the use of deep learning is likely to improve with more data as seen in Figure 7 of Alom et al., (2019) and Figure 2 of Tang et al., (2018).

3.5 Conclusions

We have proposed a methodology for the successful use of data-driven approach for kick detection by using an ensemble of LSTM-RNNs. The aim of using LSTM-RNN is to overcome the challenges of learning how to utilize d-exponent and stand pipe pressure data for early kick detection without false alarm.

The use of an ensemble of LSTM-RNNs is able to achieve early kick detection without miss and false alarm in all the four drilling cases considered here, whereas the ensemble of simple ANNs is able to do the same only for three out of the four cases. It could be argued that the performance of the simple ANN may improve with a sufficient amount of training data and or certain network configuration/training options.

Comparing the result of the methodology presented here with the result achieved in Tang et al., (2019) shows that the ensemble LSTM-RNN methodology being introduced here can be used to complement the methodology used in Tang et al., (2019) and Mao & Zhang, (2019) with sufficient training data.

It is recommended that continual training and testing be conducted by acquiring more data in order to improve the confidence in the proposed method prior to field implementation. This is generally the requirement for any machine learning based method, as performance depends on the data used in training it. Therefore, it is recommended that petroleum drilling companies set up accessible large database of sensor data and the corresponding events in order to facilitate the implementation of the methodology proposed here for kick detection during drilling.

3.6 Appendix A. Peak reduction in SPP data assuming case 1 is to be used for testing and cases 2, 3 and 4 used for training

Fig. 3.24 Peak reduction in SPP data for case 1

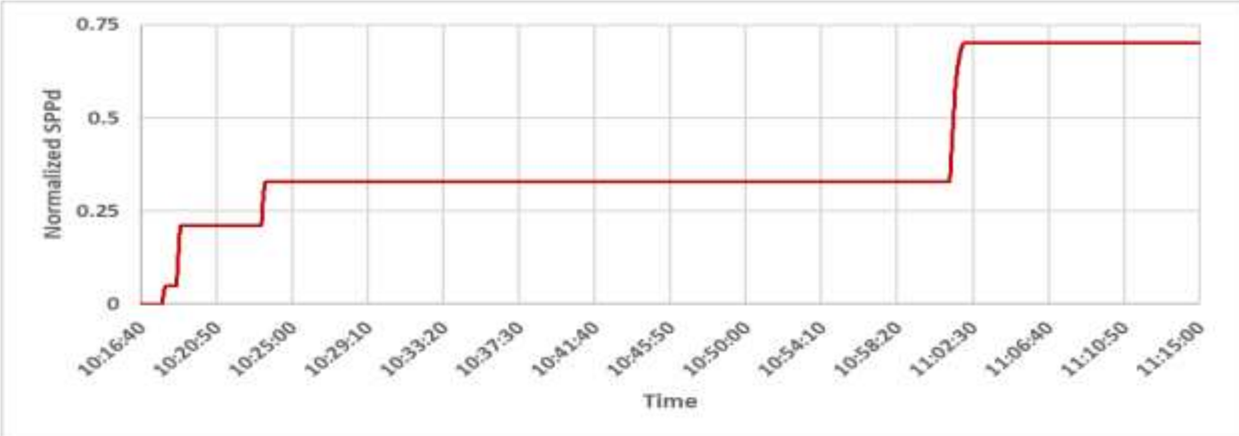

Fig. 3.25 Peak reduction in SPP data for case 2

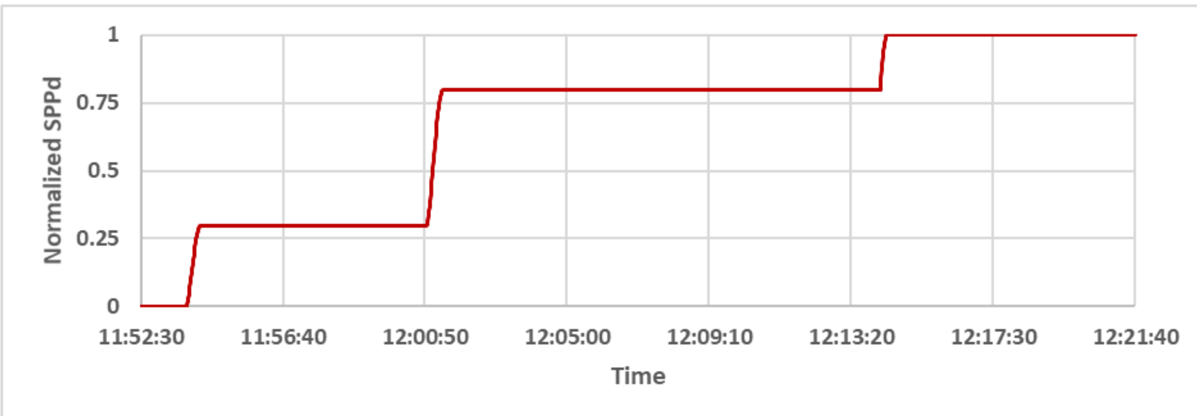

Fig. 3.26 Peak reduction in SPP data for case 3

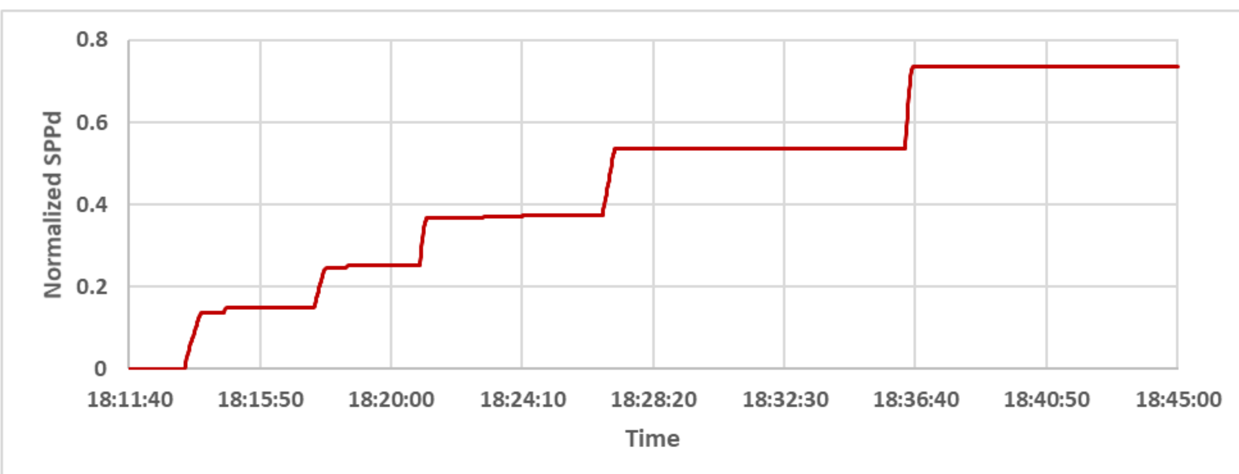

Fig. 3.27 Peak reduction in SPP data for case 4

3.7 Appendix B. Slope trend of the d-exponent data assuming case 1 is to be used for testing and cases 2, 3 and 4 used for training

Fig. 3.28 Slope trend of d-exponent data for case 1 (using sliding window of 3 minutes)

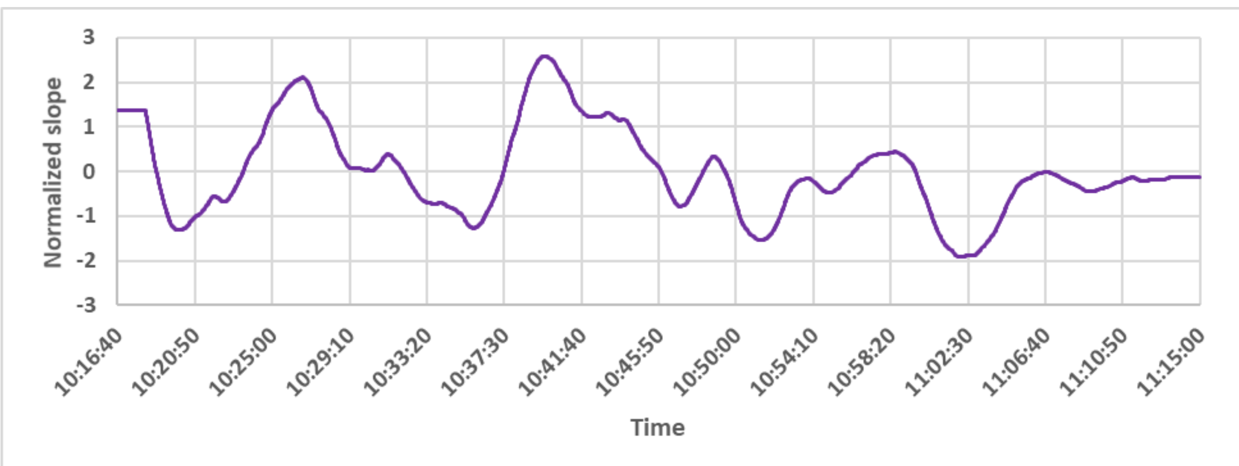

Fig. 3.29 Slope trend of d-exponent data for case 2 (using sliding window of 3 minutes)

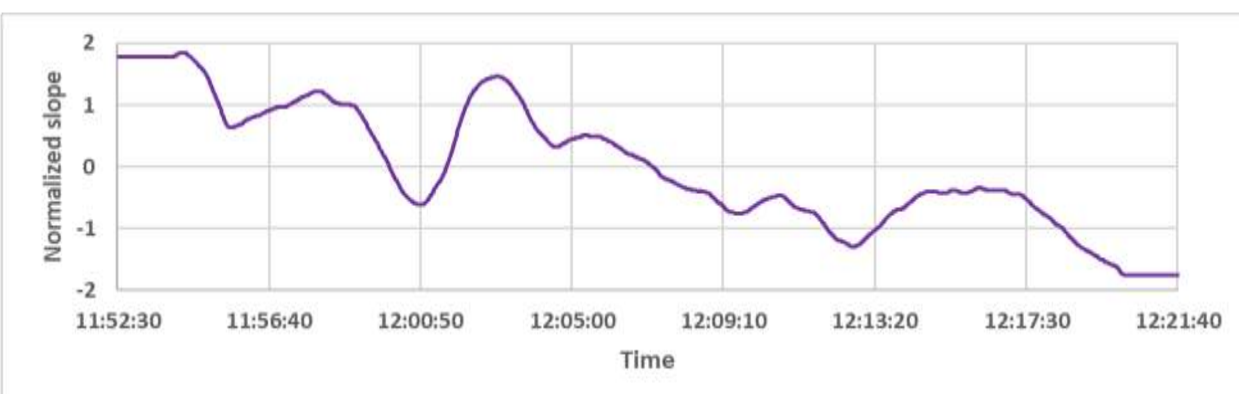

Fig. 3.30 Slope trend of d-exponent data for case 3 (using sliding window of 3 minutes)

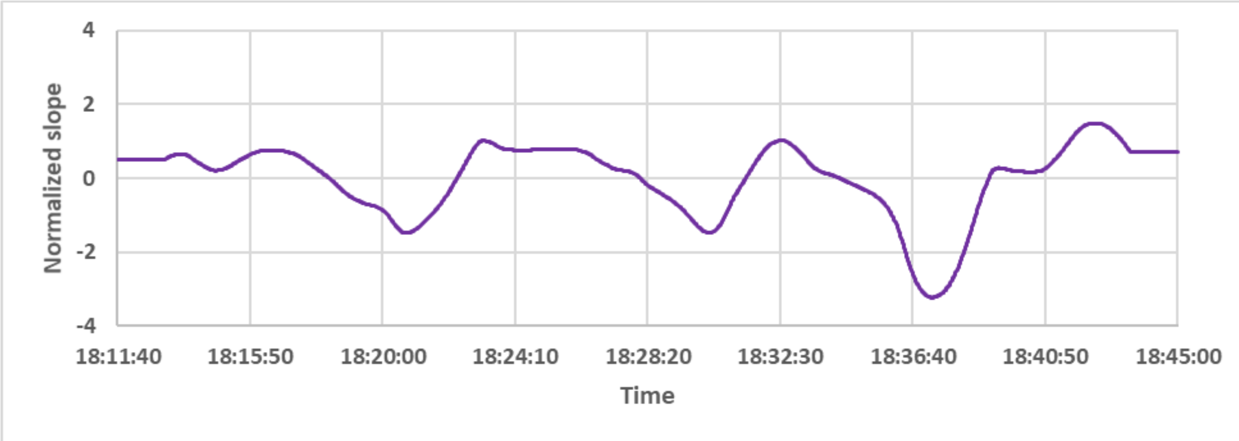

Fig. 3.31 Slope trend of d-exponent data for case 4 (using sliding window of 3 minutes)

3.8 Nomenclature

SPP_t	Standpipe pressure data at time t
$SPPd$	Change in trend of a complete time series data
$SPPd_t$	Change in trend of a standpipe pressure data at time t
SD	Slope of a complete time series d-exponent data
SDx	Slope of a complete time series d-exponent data of a particular case named x
μ_{cx}	Mean of SDx
s_{cx}	Standard deviation of SDx
$NSD3_t$	Normalised SDx

3.9 Acronyms

DPG	Drilling parameter group
ANN	Artificial neural network
FPG	Flow parameter group
LSTM	Long short-term memory
RNN	Recurrent neural network
SPP	Standpipe pressure
WOB	Weight on bit
ROP	Rate of penetration

3.10 References

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. Electronics, 8(3), 292.

Alouhali, R., Aljubran, M., Gharbi, S., & Al-yami, A. (2018). Drilling through data: Automated kick detection using data mining. SPE International Heavy Oil Conference and Exhibition.

Anifowose, F. A., Labadin, J., & Abdulraheem, A. (2017). Ensemble machine learning: An untapped modeling paradigm for petroleum reservoir characterization. Journal of Petroleum Science and Engineering, 151, 480–487.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning . Springer, New York.

Fjetland, A. K., Zhou, J., Abeyrathna, D., & Gravdal, J. E. (2019). Kick detection and influx size estimation during offshore drilling operations using deep learning. 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2321–2326.

Foresee, F. D., & Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. Proceedings of International Conference on Neural Networks (ICNN'97), 3, 1930–1935.

Graves, A. (2012). Supervised Sequence Labelling wth Recurrent Neural Networks, Vol. 385. Springer, Heidelberg.

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). Neural network design. PWS Publishing Co.

Haibo, L; Tang, Y; Li, X; Luo, Y. (2014). RESEARCH ON DRILLING KICK AND LOSS MONITORING METHOD BASED ON BAYESIAN CLASSIFICATION. Pakistan Journal of Statistics, 30(6).

Hargreaves, D., Jardine, S., & Jeffryes, B. (2001). Early kick detection for deepwater drilling: New probabilistic methods applied in the field. SPE Annual Technical Conference and Exhibition.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Kamyab, M., Shadizadeh, S. R., Jazayeri-rad, H., & Dinarvand, N. (2010). Early kick detection using real time data analysis with dynamic neural network: A case study in iranian oil fields. Nigeria Annual International Conference and Exhibition.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. ArXiv Preprint ArXiv:1412.6980.

Lenail, A. (2019). Publication-ready NN-architecture schematics . In alexlenail.me/NN-SVG/ .

Liang, H., Zou, J., & Liang, W. (2019). An early intelligent diagnosis model for drilling overflow based on GA–BP algorithm. Cluster Computing, 22(5), 10649–10668.

Mani, S., Sharma, S., & Kaur, D. (2018). Web plot digitizer software: can it be used to measure neck posture in clinical practice. Asian J Pharm Clin Res, 11, 2–3.

Mao, Y., & Zhang, P. (2019). An automated kick alarm system based on statistical analysis of real-time drilling data. Abu Dhabi International Petroleum Exhibition & Conference.

Mikolov, T. (2012). Statistical language models based on neural networks. Ph.D. Thesis, Brno University of Technology.

Nybo, R., Bjorkevoll, K. S., & Rommetveit, R. (2008). Spotting a false alarm. Integrating experience and real-time analysis with artificial intelligence. Intelligent Energy Conference and Exhibition.

Nybo, R., Bjorkevoll, K. S., Rommetveit, R., Skalle, P., & Herbert, M. C. (2008). Improved And Robust Drilling Simulators Using Past Real-Time Measurements And Artificial Intelligence. Europec/EAGE Conference and Exhibition.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. International Conference on Machine Learning, 1310–1318.

Phattanawasin, P., Burana-Osot, J., Sotanaphun, U., & Kumsum, A. (2016). Stability-indicating TLC—image analysis method for determination of andrographolide in bulk drug and Andrographis paniculata formulations. Acta Chromatographica, 28(4), 525–540.

Pournazari, P., Ashok, P., van Oort, E., Unrau, S., & Lai, S. (2015). Enhanced kick detection with low-cost rig sensors through automated pattern recognition and real-time sensor calibration. SPE Middle East Intelligent Oil and Gas Conference and Exhibition.

Sundermeyer, M., Ney, H., & Schlüter, R. (2015). From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(3), 517–529.

Tang, A., Tam, R., Cadrin-Chênevert, A., Guest, W., Chong, J., Barfett, J., Chepelev, L., Cairns, R., Mitchell, J. R., & Cicero, M. D. (2018). Canadian Association of Radiologists white paper on artificial intelligence in radiology. Canadian Association of Radiologists Journal, 69(2), 120–135.

Tang, H., Zhang, S., Zhang, F., & Venugopal, S. (2019). Time series data analysis for automatic flow influx detection during drilling. Journal of Petroleum Science and Engineering, 172, 1103–1111.

Vajargah, A. K., & van Oort, E. (2015). Early kick detection and well control decision-making for managed pressure drilling automation. Journal of Natural Gas Science and Engineering, 27, 354–366.

Xie, H., Shanmugam, A. K., & Issa, R. R. A. (2018). Big Data Analysis for Monitoring of Kick Formation in Complex Underwater Drilling Projects. Journal of Computing in Civil Engineering, 32(5), 4018030.

Yin, H., Si, M., & Wang, H. (2014). THE WARNING MODEL OF THE EARLY KICK BASED ON BP NEURAL NETWORK. Pakistan Journal of Statistics, 30(6).

Yin, H., Si, M., Li, Q., Zhang, J., & Dai, L. (2019). Kick Risk Forecasting and Evaluating During Drilling Based on Autoregressive Integrated Moving Average Model. Energies, 12(18), 3540.

Zhang, D., Chen, Y., & Meng, J. (2018). Synthetic well logs generation via Recurrent Neural Networks. Petroleum Exploration and Development, 45(4), 629–639.

Chapter 4

4. Combining porosity and resistivity logs for pore pressure prediction

Preamble

This chapter addresses an objective of this dissertation as outlined in Section 1.3, which is exploiting interelationship between parameters for pore pressure prediction. The methodology presented in this chapter aims to enhance the accuracy of combining sonic porosity and resistivity data for pore pressure prediction. Accurate pore pressure prediction is vital for drilling safety, as presented in Section 1.1. and Section 1.6.

I (Augustine Uhunoma Osarogiagbon) have contributed to Conceptualization, Methodology, Formal Analysis, Software, Investigation, Writing - Original Draft, and Writing - Review & Editing of this work, while Dr. Olalere Oloruntobi's contributed to Methodology, Formal Analysis, Writing - Review & Editing; Dr. Faisal Khan contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing, Supervision, and Project Administration; Dr. Ramachandran Venkatesan contributed to Methodology, Formal Analysis, Writing - Review & Editing, Supervision, Project Administration; and Dr. Paul Gillard contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing, and Supervision. A version of this chapter has been submitted in the Journal of Petroleum Science and Engineering.

Abstract

Pore pressure prediction represents an important safety aspect of drilling engineering. Accurate pore pressure prediction is required for appropriate mud weight usage. Kick can occur when mud weight is lower than pore pressure gradient and this can result in disastrous events such as blowout when the kick is not properly controlled. Likewise, too high mud density can fracture the reservoir which can lead to several problems. Thus, the need to research on means of improving accurate pore pressure prediction during drilling is in order.

In this article, two methodologies are presented. One of the methodologies is developed to utilize resistivity data for pore pressure prediction, and the other methodology is developed if resistivity and porosity data are available for pore pressure prediction. Several methodologies already exist for pore pressure prediction with resistivity data. Therefore, the methodology presented in this article is compared to other resistivity-based methodologies in order to observe their pore pressure prediction capabilities. Field data is used for testing prediction performance in terms of mean absolute percentage error, root mean square error and Pearson product moment correlation coefficient. Results of the test show that the methodology developed in this article performed best.

Different logging/measurement parameters are used for pore pressure prediction e.g. resistivity log, sonic velocity, corrected d-exponent, etc. One way to improve accuracy of pore pressure prediction is utilizing pore pressure prediction from different logging/measurement parameters. For the other methodology which utilizes resistivity and porosity for pore pressure prediction, the methodology is proposed to utilize the change in Archie's cementation exponent. This is because the effect of cementation on pore pressure prediction could become significant at greater depths. Testing with field data showed that this methodology also performs better than simply averaging

pore pressure prediction from resistivity and porosity logs using other conventional equations. In addition to the methodology developed for combining porosity and resistivity log for pore pressure prediction, machine learning can also be utilized. Results obtained using artificial neural network indicate better performance in comparison to simply averaging predictions from conventional means of using resistivity and porosity.

Keywords: Pore pressure; resistivity; porosity; overburden stress; drilling, kick; effective stress coefficient; vertical effective stress; cementation, artificial neural network, machine learning.

4.1 Introduction

The importance of accurately predicting pore pressure during drilling cannot be over-emphasized. This is because mud weight should be higher than pore pressure gradient to prevent kick (except for special cases e.g., underbalanced drilling), and mud weight should be lower than formation fracture gradient to prevent formation fracture (except when fracture is clearly intended). Uncontrolled kick can lead to disastrous consequences such as blowout; likewise, well fracture not properly managed can lead to loss circulation, kick, well collapse, etc. (Abimbola et al., 2015; Sadiq & Nashawi, 2000). Therefore, accurate pore pressure prediction is important in designing a drilling mud weight (density) window of operation in order to prevent the use of too high or too low drilling mud weight (Brahma & Sircar, 2018; Feng et al., 2015; Oloruntobi et al., 2020; Osarogiagbon et al., 2021).

Several methodologies developed for pore pressure prediction refer to the relationship between overburden stress, vertical effective stress and pore pressure as a reference point. These equations often use sensing parameters such as resistivity, sonic velocity, corrected drilling exponent, porosity, hydro mechanical specific energy, etc. (Eaton, 1975; Oloruntobi, 2019; Zhang, 2011). Despite successes achieved with the use of these parameters, limitations in accuracies can occur due to the nature of the reservoir. For example, resistivity, which is used for pore pressure prediction, is also sensitive to porosity, salinity, etc. (Saleh et al., 2013). Likewise, porosity changes can also be influenced by chemical processes such as cementation and dissolution, thereby making it more challenging to predict pore pressure based on porosity changes (Swarbrick, 2001). Due to numerous challenges of achieving accurate pore pressure prediction, the use of multiple approaches to enhance accuracy is recommended.

This work introduces a novel approach which combines resistivity and porosity measurements for pore pressure prediction, by exploring possible interrelationship between porosity and resistivity. In addition, a novel means of utilizing resistivity for pore pressure prediction is also presented. The industrial significance of this research work is that it offers a new set of methodology with the potential to improve accuracy of pore pressure prediction in comparison to methodologies in use. This chapter is structured as follows. Section 4.2 provides a general survey on pore pressure prediction models, Section 4.3 presents the methodology of this work, Section 4.4 describes the data used for testing the methodology presented, Section 4.5 describes results obtained, and Section 4.6 presents conclusion.

4.2 Review of pore pressure prediction models

Based on Terzaghi's soil bearing capacity theory, it has been shown that the overburden stress at a point is supported by the rock matrix and the fluid in the rock matrix, as shown by Equation (4.1) (Eaton, 1975; Liu et al., 2018).

$$S_v = S_e + P, \tag{4.1}$$

where P refers to pore pressure, S_v refers to overburden stress and S_e refers to vertical effective stress. Trapped fluid in the reservoir can therefore experience overpressure in comparison to the normal hydrostatic pressure due to the overburden stress. The overburden stress is a function of bulk density (Equation (4.2)), and bulk density can be obtained from log data.

$$S_v = 0.433 \int_o^Z \rho dZ, \tag{4.2}$$

where Z represents depth and ρ represents formation bulk density (Eaton, 1975; Liu et al., 2018; Oloruntobi et al., 2018; Oloruntobi & Butt, 2019). The properties of a rock formation, such as resistivity, sonic/seismic velocity and corrected d-exponent, normally follow an increasing trend with depth. However, the presence of an over pressured fluid within the rock can cause resistivity, sonic velocity and corrected d-exponent values to differ from the expected values following the normal trend. Equation (4.1) forms the foundation by which many pore pressure methodologies (e.g. by Eaton) were developed.

In the article by Eaton (1975), equations by which pore pressure can be estimated as a function of sonic velocity or resistivity or corrected d-exponent data were presented as described by Equation (4.3).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{x_o}{x_n}\right)^k, \tag{4.3}$$

where $\frac{P}{Z}$ represent the predicted pore pressure gradient, $\frac{P_n}{Z}$ represents the normally expected hydrostatic pressure gradient and $\frac{S_v}{Z}$ represents overburden stress gradient. $\frac{x_o}{x_n}$ refers to the ratio of the observed data to the normally expected data for resistivity parameter or sonic/seismic velocity parameter or corrected d-exponent parameter. The Eaton's coefficient k can take a value of 1.2 for pore pressure prediction using resistivity data or corrected d-exponent data, and k can take a value of 3 for sonic/seismic velocity data. For pore pressure measurement based on sonic/seismic velocity where transit time is used as an indication of velocity, Equation (4.3) is modified to yield Equation (4.4).

$$\frac{P}{Z} = \frac{S_v}{Z} - (\frac{S_v}{Z} - \frac{P_n}{Z})(\frac{\Delta t_n}{\Delta t_o})^3, \tag{4.4}$$

where Δt_n is the normally expected transit time at that depth, Δt_o is the observed transit time at the depth.

Pore pressure can also be predicted using log of resistivity. Foster & Whalen (1965) presented an equation which shows how pore pressure can be obtained from normal shale resistivity R_n and observed shale resistivity R_o as shown in Equation (4.5) (Foster & Whalen, 1965; Oloruntobi, 2019).

$$\frac{P}{Z} = \frac{P_n}{Z} + \frac{0.535}{\mathrm{Z} \times \log(S)} \log(\frac{R_n}{R_o}). \tag{4.5}$$

In Equation (4.5), $\log(S)$ represents the slope of the formation factor vs depth plot.

Bowers pointed out that outside under compaction, fluid expansion can also result in overpressure (Bowers, 1995). The significance of overpressure due to fluid expansion is that it can cause the velocity-stress relationship in the rock to deviate from the expected trend (virgin curve) and follow an unloading curve. Based on this, he presented equations for evaluating the vertical effective stress as a function of sonic/acoustic velocity with and without the effect of fluid expansion.

$$V = 5000 + A({S_e}^B). \tag{4.6}$$

Equation (4.6) shows velocity (V) as a function of vertical effective stress S_e without the effect of fluid expansion. A and B are parameters that can be obtained by fitting data.

$$V = 5000 + A[S_{eMax}(S_e/S_{eMax})^{1/U}]^B. \tag{4.7}$$

Equation (4.7) shows velocity (V) as a function of vertical effective stress S_e with the effect of fluid expansion causing the velocity-stress relationship to assume the unloading curve. A and B are the same for Equation (4.6). U is a parameter which describes how plastic the sediment has become. For example, when U is infinity, the deformation experienced by the rock due to the fluid expansion is irreversible, and a value of 1 shows no permanent deformation.

S_{eMax} can be obtained as a function of v_{max} as given by Equation (4.8).

$$S_{eMax} = \left(\frac{V_{Max}-5000}{A}\right)^{1/B}, \tag{4.8}$$

where S_{eMax} and V_{Max} represent vertical effective stress and velocity, respectively, at the onset of unloading from the plot of velocity versus vertical effective stress curve. The methodology by Bowers therefore presents a more accurate means of pore pressure prediction in comparison to the methodology by Eaton when fluid expansion more significantly contributes towards over pressure in comparison to under compaction. However, the Eaton's approach can also work very well if the exponent factor k in Equation (4.3) is properly chosen. For example, in the article by Bowers (1995), for regions were overpressure was mainly due to fluid expansion rather than under compaction, setting the value of k to 5 instead of the traditional value of 3 in the Eaton's model for pore pressure prediction using velocity, resulted in good prediction.

Several other models have been developed for pore pressure prediction including:

i. Miller's method (Zhang, 2011, 2013)

$$P = S_v - \frac{1}{\lambda}\ln(\frac{V_m - V_{ml}}{V_m - V_p}), \tag{4.9}$$

where V_m is the sonic velocity in the shale matrix, V_{ml} is the sonic velocity at ground surface, V_p is the compressional velocity at our depth of interest, λ is an empirically obtained parameter which defines the rate of velocity increase with effective stress. For the case where the rock follows the "unloading curve", Equation (4.10) is used.

$$P = S_v - \frac{1}{\lambda}\ln[a_m\left(1 - \frac{V_p - V_{Max}}{V_m - V_{ml}}\right)], \tag{4.10}$$

where V_{Max} represents the velocity at the onset of unloading, a_m is normally 1.8 and it can be obtained as shown: $a_m = \frac{V_p}{V_{Max}}$.

ii. Tau model (Zhang, 2011, 2013)

$$P = S_v - A_s(\frac{C - \Delta t}{\Delta t - D})^{B_s}, \tag{4.11}$$

where Δt is the compressional transit time, A_s and B_s are fitting parameters from data. C and D are normally $200\ \mu s/ft$ and $50\ \mu s/ft$ respectively.

iii. Model by Liu et al. (L. Liu et al., 2018)

$$P = a({V_t}^b - V^b), \tag{4.12}$$

where V_t is a theoretical velocity given by Equation (4.13).

$$V_t = V_0 + s * Z, \tag{4.13}$$

where Z represents depth, V_0 and s represents constants. V is the measured seismic/sonic velocity. a and b are constants which relates effective stress S_e to velocity V as shown by Equation (4.14). b could be assumed to be equal to 3.

$$S_e = aV^b. \tag{4.14}$$

iv. Compressibility method (Azadpour et al., 2015)

$$P = \left(\frac{(1-\varphi)C_p S_e}{(1-\varphi)C_p - \varphi C_P}\right)^{\gamma}, \tag{4.15}$$

where φ represents porosity, C_p represents pore compressibility, S_e is the vertical effective pressure determined from the difference between overburden pressure and hydrostatic pressure, and γ is a constant which could range from 0.9 to 1. The pore compressibility was also expressed as a function of porosity as shown in Equation (4.16).

$$C_p = \frac{10^{-6}}{0.444+0.131\ln(\varphi)}\ psi^{-1}. \tag{4.16}$$

In the article by Azadpour et al., (2015), comparison was made between Eaton, Bowers and compressibility methods for pore pressure prediction using a gas field in the Persian Gulf basin. This comparison revealed that the Eaton and compressibility method performed better than the Bowers method. The Eaton method with exponent coefficient (k in Equation (3)) of 0.5 performed best.

v. Multi regression model (Deng et al., 2017)

Empirical equations were presented to predict pore pressure for shallow soft clay formation as shown:

$$V_m = A_1\rho_m + A_2\sigma + A_3 v_{sh}, \tag{4.17}$$

$$V_f = B_1\rho_f + B_2\rho_p + B_3 T, \tag{4.18}$$

$$V_p = (V_m)^{(1-\varphi)} \times (V_f)^{\varphi}, \tag{4.19}$$

where V_m represents p-wave velocity for the rock skeleton, V_f represents p-wave velocity for the formation fluid, V_p represents the p-wave velocity at a given position, φ represents porosity, v_{sh} represents shale content, T represents temperature, ρ_m represents density of framework of rock, ρ_f represents the density of fluid and ρ_p represents equivalent density of formation pore pressure. $A_1, A_2,\ A_3,\ B_1,\ B_2$ and B_3 represent coefficients to be determined from regression on the given data.

vi. Pore pressure from porosity (Zhang, 2011; Zhang, 2019)

In the article by Zhang (2011), pore pressure was modeled as a function of porosity, and porosity in turn was modelled as a function of sonic velocity. Porosity normally decreases as a function of depth as shown by Equation (4.20). Therefore, an abnormality in the porosity trend is an indication of under compaction.

$$\varphi_n = \varphi_g e^{-cZ}, \tag{4.20}$$

where φ_g is the porosity at mudline, φ_n is the normal porosity trend, φ_o is the actual shale porosity at depth below mudline Z, and c is compaction constant. Equation (4.21) relates porosity to pore pressure.

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{\varphi g}{\varphi_o}\right)}{cZ}\right). \tag{4.21}$$

The parameters in Equation (4.21) have the same definitions as those of Equations 4.3 and 4.20. Several researches have shown that Equation (4.1) may not be accurate for reservoirs located at great depth or regions with high cementation effect (Dassanayake et al., 2015; Mao et al., 2018). Equation (4.22) represents a more robust model for developing pore pressure prediction equations in comparison to Equation (4.1) (Amiri et al., 2019; Dassanayake et al., 2015; Mao et al., 2018; Sayers et al., 2002; Zhang, 2013).

$$S_v = S_e + \alpha P. \tag{4.22}$$

In Equation (4.22), α refers to effective stress coefficient; all other parameters in Equation (4.22) have the same definitions as those in Equation (4.1). A review of several possible means of evaluating α can be found in (Mao et al., 2018). Equation (4.23) represents a theoretical means of evaluating α for isotropic rocks (Biot, 1941; Njiekak & Schmitt, 2019; Nur & Byerlee, 1971; Shen et al., 2017).

$$\alpha = 1 - \frac{C_s}{C_v}, \tag{4.23}$$

where C_s is the compressibility of the solid material, C_v is the compressibility of the total volume. In the work by Mao et al. (2018), it was shown that the effective stress coefficient can take values in the range of φ to 1, where φ refers to porosity. The effective stress coefficient value of 1 models rock without cementation and effective stress coefficient value of φ models rock with perfect cementation (Mao et al., 2018; Zhang, 2013). Considering that effective stress coefficient can vary based on reservoir properties, several empirical equations have also been developed to capture the mathematical relationship between effective stress coefficient and porosity. These empirical equations were developed for different lithologies, porosity values, extent of consolidation etc. A summary of some of these equations can be found in Amiri et al. (2019). Sarker & Batzle (2008) recommended depth varying values of effective stress coefficient to be used for pore pressure prediction as opposed to varying Eaton's coefficient (k in Equation (4.3)), because effective stress coefficient is a rock property. They recommended that depth varying effective stress coefficient could be obtained by back calculating from calibration wells where pressure data have already been obtained.

We have not come across any methodology which explores the interrelationship between resistivity and porosity for pore pressure prediction. The review on pore pressure prediction has shown that there remains a lot to explore on developing pore pressure prediction models. For example, even though correlations have been developed between porosity and effective stress coefficient, Frempong & Butt (2006) showed that effective stress coefficient can be influenced by porosity, pore geometry, pore pressure and confining pressure. In addition, other factors such as clay content/lithology (Luo et al., 2015), fracture (Xu et al., 2006) and cementation (Alam et al., 2012) can influence effective stress coefficient. The significance of cementation on pore pressure prediction has encouraged us to develop a porosity-resistivity pore pressure relationship that can capture Archie's cementation exponent, because Archie's cementation exponent is influenced by degree of cementation (Glover, 2009).

4.3 Methodology for pore pressure prediction

The main outcome of this chapter is to present two new equations for pore pressure prediction, which are Equation (4.24) and Equation (4.25).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{log_e\left(\frac{\varphi_g}{\varphi_n}\right)} \times \frac{log_e\left(\frac{R_{sf}}{R_n}\right)}{log_e\left(\frac{R_{sf}}{R_o}\right)}\right)^{\frac{1}{2}}, \tag{4.24}$$

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{R_{sf}}{R_n}\right)}{log_e\left(\frac{R_{sf}}{R_o}\right)}\right). \tag{4.25}$$

In Equation (4.24) and Equation (4.25), R_n and R_o have the same meaning as those of Equation (4.5) and R_{sf} represents the resistivity scaling factor which can be determined empirically in a similar manner to Eaton's coefficient k of Equation 3. All other parameters of Equation (4.24) and Equation (4.25) have the same definitions as those of Equation (4.20) and Equation (4.21). Equation (4.24) can be used if resistivity and sonic porosity data are available, and Equation (4.25) can be used if only resistivity data is available for pore pressure prediction. Equation (4.24) can be referred to as cementation-exponent approach while Equation (4.25) can be referred to the resistivity component of the cementation-exponent approach. Equation (4.24) is referred to as cementation-exponent approach because it is the same as Equation (4.26) (see section 4.9).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\left(\frac{1}{m_o} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_o})} \right) \div \left(\frac{1}{m_n} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_n})} \right) \right)^{\frac{1}{2}}, \tag{4.26}$$

where m_n is the normal Archie's cementation exponent obtained from the normal porosity and normal resistivity trend models. m_o is the observed Archie's cementation exponent obtained from actual porosity and resistivity measurements. Hence in Equation (4.26), pore pressure prediction at a particular depth depends on the deviation of actual cementation exponent at that depth location from the normally expected cementation exponent at the same depth location based on the rock compaction trend. The details on how Equation (4.24) and Equation (4.25) are derived are presented in section 4.3.1. In addition to Equation (4.24), artificial neural network also offers an additional means of exploring the interrelationship between resistivity log and porosity log for pore pressure prediction. More details on how artificial neural network is utilized in this work can be found in Section 4.3.2. The flowchart of the methodology is presented in Fig. 4.1.

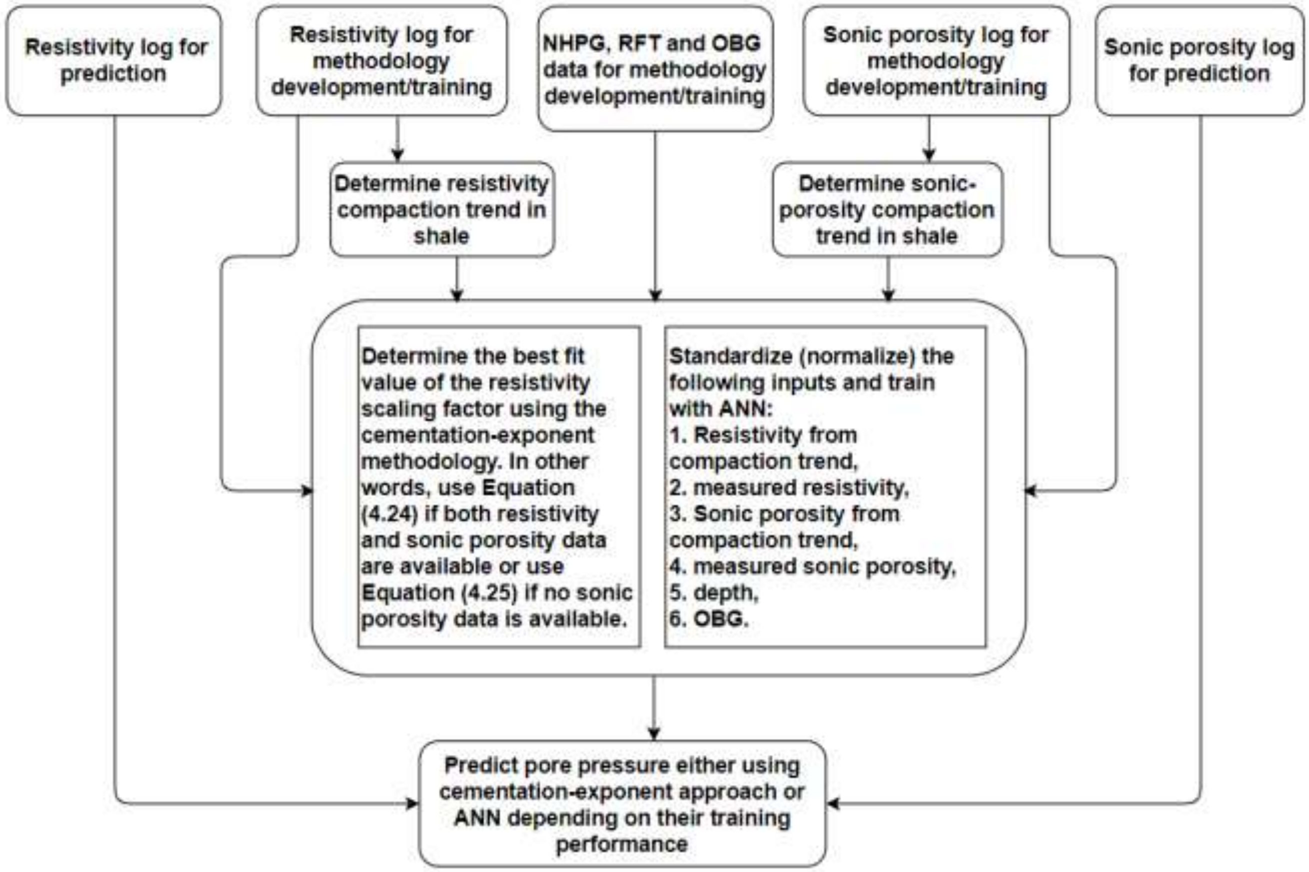

Fig. 4.1 Flowchart showing use of developed methodology for pore pressure prediction

4.3.1 Developing the cementation-exponent approach

Generally, depth compaction trends exist in rocks (Magara, 1980). For example, porosity generally reduces with depth, and Equation (4.20) is an example of such a porosity-depth relationship. Likewise, for a given rock type (e.g. shale) completely saturated by a single fluid type (e.g. salt water), resistivity will also have a trend of increasing with depth (Zhang, 2011; Zhao et al., 2018). The major reason for this is that shale has higher resistivity than salt water, and porosity (which represents the fraction of volume that salt water can occupy in a rock) reduces with depth. Archie's equation relates resistivity to porosity, as well as how rock cementation can influence this relationship. Equation (4.27) shows how Archie's cementation exponent m is related to resistivity of brine saturated rock formation R_o, rock formation water resistivity R_w and tortuosity factor or structural parameter δ which can be occasionally assumed as 1 (Archie, 1942; Saner et al., 1996; Sethi, 1979) . φ_o refers to porosity in Equation (4.27). Although Archie's equation was initially adopted for sandstone/carbonates formations (Worthington, 1993), it has also been used for shale formation (Yu & Aguilera, 2011; Zhang, 2019).

$$\frac{R_o}{R_w} = \frac{\delta}{\varphi_o^m}. \tag{4.27}$$

Equation (4.21) provides a means by which logarithm of porosity is used for pore pressure prediction and Equation (4.5) also presents a means by which the logarithm of resistivity reading can be used for pore pressure prediction. Therefore, Equation (4.21) is modified to accommodate for the effect of variation in rock formation resistivity measurements.

Equation (4.5) can be expressed as Equation (4.28) with the meaning of K_1 shown in Equation (4.29).

$$\frac{P}{Z} = \frac{P_n}{Z} + k_1 \log(\frac{1}{R_o}) - k_1 \log(\frac{1}{R_n}), \tag{4.28}$$

$$K_1 = \frac{0.535}{Z \times \log(S)}. \tag{4.29}$$

Equation (4.28) shows that increase in $\log(\frac{1}{R_o})$ leads to increase in pore pressure but increase in $\log(\frac{1}{R_n})$ leads to decrease in pore pressure. This can be used in enhancing Equation (4.21) by including $\log(\frac{1}{R_o})$ and $\log(\frac{1}{R_n})$ which yields Equation (4.30).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{cZ} \times \frac{log_e(\frac{1}{R_n})}{log_e(\frac{1}{R_o})}\right). \tag{4.30}$$

Equation (4.30) can be made more general by introducing parameters R_{sf} and *D*, thereby yielding Equation (4.31).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)(\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_0}\right)}{cZ}\right)\left(\frac{log_e\left(\frac{R_{sf}}{R_n}\right)}{log_e\left(\frac{R_{sf}}{R_o}\right)}\right))^D. \tag{4.31}$$

The purpose of introducing parameter R_{sf} is to include a tortuosity factor and formation water resistivity because we aim to transform Equation (4.31) to account for Archie's cementation exponent of Equation (4.27). The parameter R_{sf} can be termed resistivity tuning parameter or resistivity scaling factor because it can adjust the effect of resistivity reading on pore pressure prediction. For example, when $D = 1$ and R_{sf} tends to infinity or zero, Equation (4.31) tends to Equation (4.21). Considering that Equation (4.31) involves multiplying both porosity indicators and resistivity indicators for pore pressure prediction, the constant *D* which can be termed gain equalization factor serves as a means of tempering the effect. *D* can be given a value of 1/2 in order to perform a square root operation.

Considering that $cZ = log_e(\frac{\varphi_g}{\varphi_g\ e^{-cZ}})$; and from Equation (4.20), $\varphi_n = \varphi_g e^{-cZ}$, Equation (4.32) therefore shows a broader description of cZ.

$$cZ = log_e(\frac{\varphi_g}{\varphi_n}). \tag{4.32}$$

Equation (4.32) can be substituted into Equation (4.21) and Equation (4.31) to yield Equation (4.33) and Equation (4.34) respectively.

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{log_e(\frac{\varphi_g}{\varphi_n})}\right), \tag{4.33}$$

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{log_e(\frac{\varphi_g}{\varphi_n})} \times \frac{log_e(\frac{R_{sf}}{R_n})}{log_e(\frac{R_{sf}}{R_o})}\right)^{\frac{1}{2}}. \tag{4.34}$$

Table 4.1 Comparing Equation (4.33) and Equation (4.34)

Equation (4.33) (predicting pore pressure from porosity).	Equation (4.34) (obtained by enhancing Equation (4.21) in order to consider both porosity and resistivity).
$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{log_e(\frac{\varphi_g}{\varphi_n})}\right)$	$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{log_e(\frac{\varphi_g}{\varphi_n})} \times \frac{log_e(\frac{R_{sf}}{R_n})}{log_e(\frac{R_{sf}}{R_o})}\right)^{\frac{1}{2}}$

Table 4.1 shows the similarity between Equation (4.33) and Equation (4.34) in terms of using porosity and resistivity deviations from normal trends for pore pressure prediction. For example, φ_g and R_{sf} represent constant values, $log_e\left(\frac{\varphi_g}{\varphi_o}\right)$ of porosity corresponds to $log_e(\frac{R_{sf}}{R_o})$ of resistivity, likewise $log_e\left(\frac{\varphi_g}{\varphi_n}\right)$ of porosity corresponds to $log_e(\frac{R_{sf}}{R_n})$ of resistivity. For rock formation sections, where m, R_w and δ are constant, $log_e(R_o)$ has a linear relationship with $log_e(\varphi_o)$ as shown by Equation (4.35), which can be derived From Equation (4.27). Also, both φ_n and R_n varies exponentially with depth (Zhang, 2011).

$$log_e(R_o) = -mlog_e(\varphi_o) + log_e(\delta R_w). \tag{4.35}$$

Equation (4.34) can be expressed as Equation (4.36) by substituting in m of Equation (4.27) as shown in section 4.9.

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{Z}\right)\left(\left(\frac{1}{m_o} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_o})}\right) \div \left(\frac{1}{m_n} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_n})}\right)\right)^{\frac{1}{2}}. \tag{4.36}$$

4.3.2 Training and testing with artificial neural network

Artificial neural network (ANN) is the most popular machine learning approach used in petroleum drilling related studies (Osarogiagbon et al., 2021). More information on how the ANN operates can be found in (Alom et al., 2019; Hagan et al., 1996; Osarogiagbon et al., 2020). The simple ANN will be utilized to observe how machine learning can be of benefit in exploring the relationship between resistivity and porosity for pore pressure prediction. The data of each of the six input parameters; R_n, R_o, φ_n, φ_o, Z, $\frac{S_v}{Z}$ and the output parameter (RFT) used for training are standardized (normalized) by subtracting their respective mean before dividing by their respective standard deviation. For example, Equation (4.37) shows how depth is standardized before feeding into the ANN.

$$Z_{std\_train} = (Z_{train} - \mu_Z)/\sigma_Z, \tag{4.37}$$

where Z_{train} represents the part of Z that will be used for training, μ_Z represents the mean of Z_{train}, σ_Z represents the standard deviation of $Z_{train,}$ and Z_{std_train} represents the standardized depth that will be directly fed into ANN for training.

After training the ANN, the data of the input parameters (R_n, R_o, φ_n, φ_o, Z, $\frac{S_v}{Z}$) for testing are also standardized by using the mean and standard deviation of the training data of there respective counterpart before feeding them into the trained ANN. For example, Equation (4.38) shows how depth is standardized before feeding into the ANN for testing.

$$Z_{std\_test} = (Z_{test} - \mu_Z)/\sigma_Z, \tag{4.38}$$

where Z_{test} represents the part of Z that will be used for testing and Z_{std_test} represents the standardized depth that will be directly fed into ANN for testing. All other parameters in Equation (4.38) have the same definitions as those in Equation (4.37).

After feeding the trained ANN with the standardized test data of the input parameters, the output of the ANN (RFT_{std_test}) is re-adjusted by removing the effect of standardization using Equation (4.39).

$$RFT_{predicted} = (\sigma_{RFT} \times RFT_{std\_test}) + \mu_{RFT}, \tag{4.39}$$

where μ_{RFT} and σ_{RFT} represents the mean and standard deviation of the part of the RFT data used for training, and $RFT_{predicted}$ represents the predicted pore pressure gradient by the ANN implementation.

The configuration used in training the ANN is presented in Table 4.2

Table 4.2 Summary of ANN configuration

s/n	Parameter	Value
1	Number of hidden layers	1 or 2.
2	Number of hidden layer nodes considered	5, 10, 20, 30. (the number of nodes with best training result will be selected)
3	Hidden layer activation	Hyperbolic tangent
4	Output layer activation function	Linear
5	Optimization algorithm	Gauss-Newton approximation to Bayesian regularization (Foresee & Hagan, 1997)
6	Maximum training iteration	100

The training performance of the ANN configurations used in pore pressure prediction can be found in Section 4:12.

4.4 Data used for testing the model

The data for this work is the same as the data illustrated in Figures 6 and 11 of the paper by Zhang (2011).

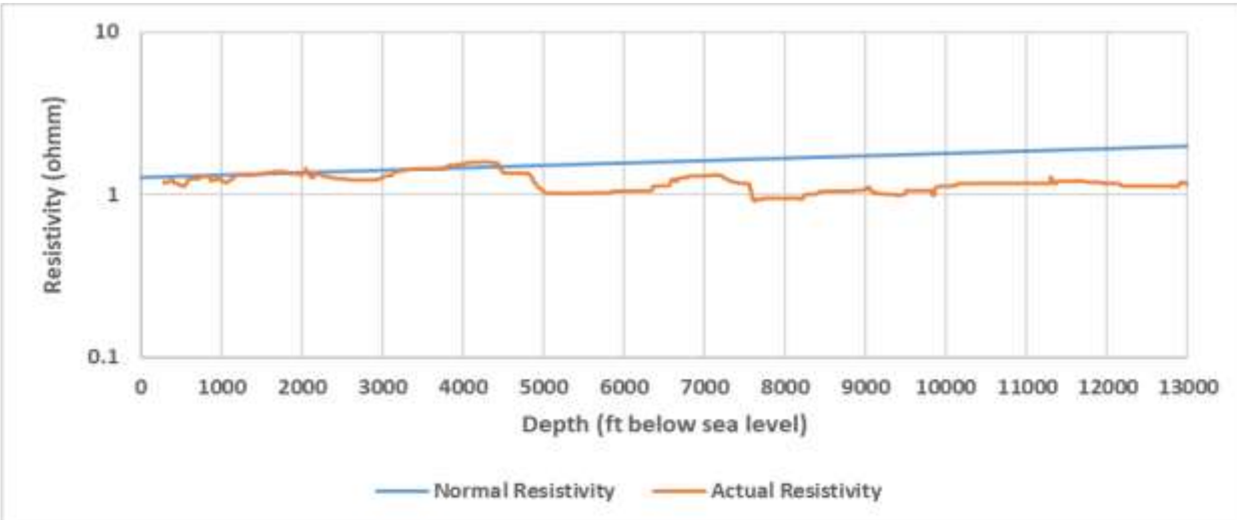

Fig. 4.2 Resistivity data (Zhang, 2011; Zhang, 2019)

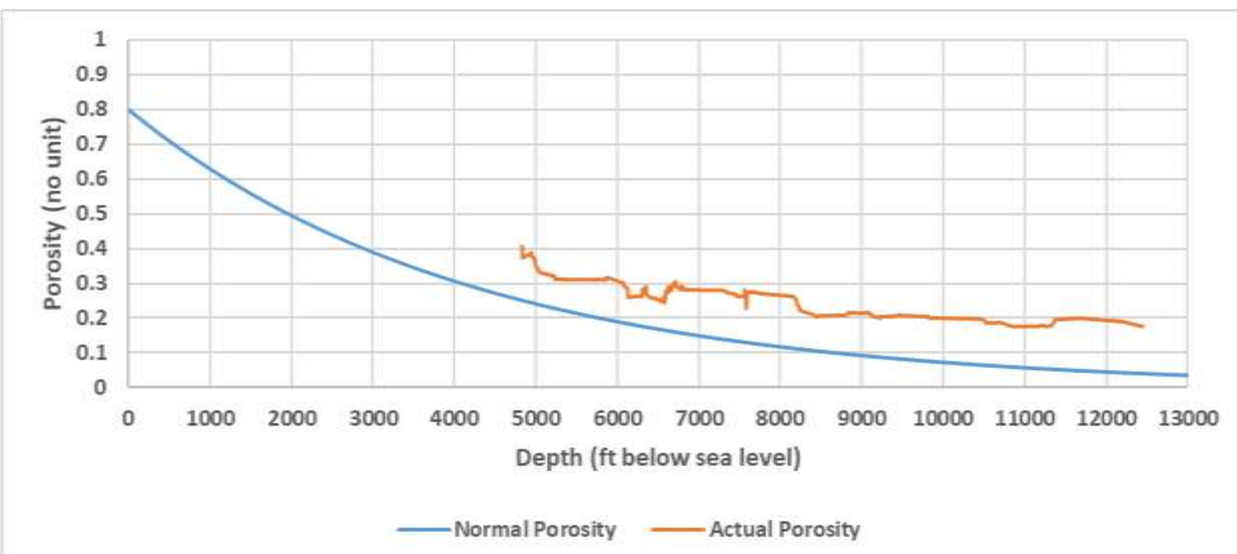

Fig. 4.3 Porosity data (Zhang, 2011; Zhang, 2019)

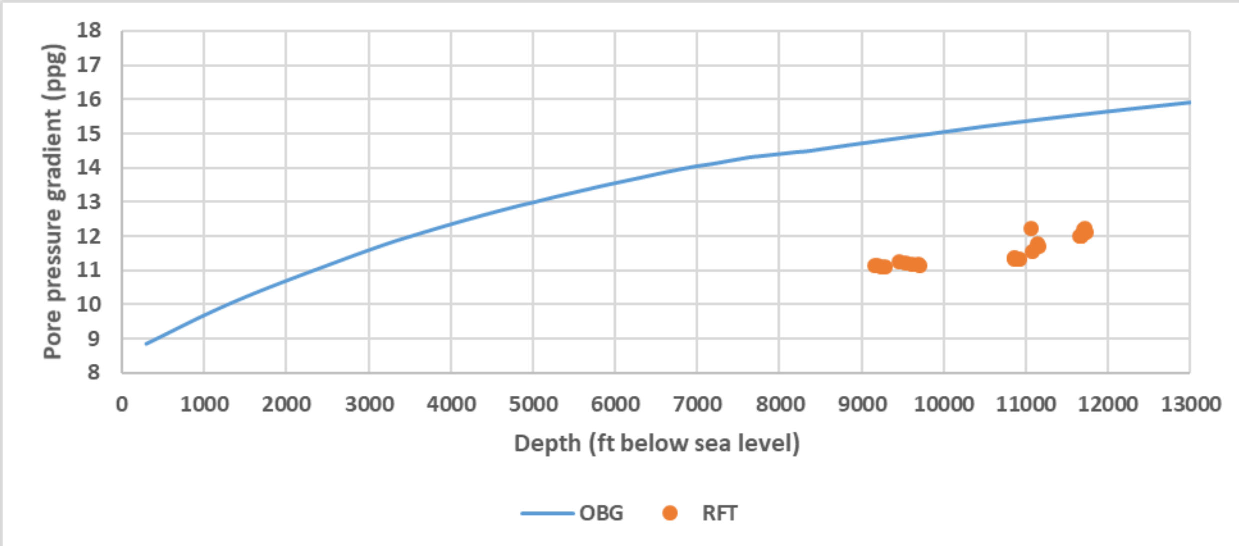

Fig. 4.4 Overburden gradient (OBG) and measured pore pressure gradient (RFT) (Zhang, 2011; Zhang, 2019)

Figures 4.2, 4.3 and 4.4 illustrate the data that was used in testing the cementation-exponent approach presented in this chapter. The value of the normal hydrostatic pressure gradient (NHPG) used is 8.7 ppg. In order to obtain uniform values of porosity, resistivity and OBG across the rock sections for which pore pressure prediction will be computed (5000 to 12000 ft. below sea level), linear interpolation was employed (Mathwork, 2020a) and the resulting interpolated data had a resolution of 0.5 ft.

4.5 RESULTS AND DISCUSSION

This section has five parts, and the aims of the subsections are:

i. Section 4.5.1 shows that sensitivity of the methodology developed in this article as a function of R_{sf}.

ii. Section 4.5.2 explains the general procedures by which the methodology developed in this article is compared to conventional approach for pore pressure prediction.

iii. Section 4.5.3 demonstrates the capability of the cementation-exponent approach in capturing the relationship between resistivity/porosity and pore pressure by using all RFT data, i.e. all data is used for training, and their performance is observed.

iv. Section 4.5.4 divides the field data into training and testing, explains the methodology developed in this article, and compares conventional approach and ANN in terms of their ability to perform pore pressure prediction on test data after been trained with the training data.

v. Section 4.5.5 provides discussion on the results and the methodologies.

4.5.1 Sensitivity analysis of the cementation-exponent approach

Considering that several possible values of R_{sf} can be used, the results obtained by using different values of R_{sf} will be presented. Fig. 4.5 shows how varying values of R_{sf} can influence performance.

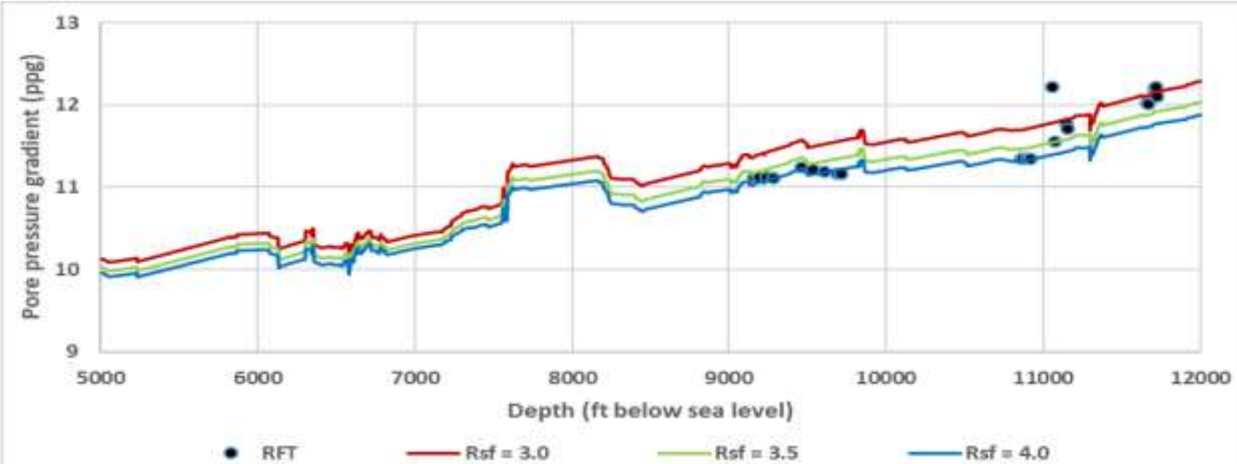

Fig. 4.5 Pore pressure prediction using different values of resistivity scaling factor (Rsf) for the cementation-exponent approach

Fig. 4.5 shows that lower values of R_{sf} (around 3.0) perform better for pore pressure prediction at depth below sea level greater than 11000 ft., whereas higher values of R_{sf} (around 4.0) perform better for pore pressure prediction at depth below sea level lesser than 10000 ft.

4.5.2 Comparing the capability of the methodology developed to perform pore pressure prediction

Comparison will be done based on:

i. The use of only resistivity data,

ii. The use of resistivity and sonic porosity data.

For use of only resistivity data, the approach developed in this chapter will be compared with the method developed by:

i. Eaton as shown by Equation (4.3),

ii. Whelan and Foster as shown by Equation (4.5).

For use of resistivity and sonic porosity, the cementation-exponent approach and ANN will be compared with a conventional means of combining multiple source of the same information (simple averaging) as described by Equation (4.40).

$$\frac{P}{Z}_{(average)} = \frac{1}{2} \times \left(\left(\frac{P}{Z}\ from\ Equation\ (4.3)\ or\ (4.5) \right) + \frac{P}{Z}\ from\ Equation\ (4.21) \right). \quad (4.40)$$

Equation (4.40) is a simple and practical means of combining porosity and resistivity for pore pressure prediction by averaging the pore pressure predicted using the porosity approach by Zhang (2011) and the better option between the resistivity approach by Eaton (1975) and the resistivity approach by Foster & Whalen (1965).

Quantitative comparison of prediction by the methodologies will be done in terms of mean absolute percentage error (MAPE), root mean square error (RMSE) and square of the Pearson product moment correlation coefficient (R square). This will be with reference to the measured pore pressure gradient (RFT data) serving as ground truth data.

Empirical parameters such as k for Eaton's approach, R_{sf} for the cement-exponent approach and $\log(S)$ by Foster & Whalen (1965) strongly influences the performance of their respective methodologies. Hence, each methodology is trained by varying its empirical parameter over a range and the performance of these methodologies as a function of their respective empirical parameter can be evaluated. This is done to obtain the best performance that each methodology can achieve. The performances of each methodology as a function of their respective empirical parameters can be found in Section 4.10 and Section 4.11. The summaries of the performances (minimum MAPE, minimum RMSE and maximum R square) from the figures of Section 4.10 and Section 4.11 are presented in Table 4.4 and Table 4.7.

4.5.3 Observing performance of the cementation-exponent approach with all RFT data available.

In this section, all RFT data (24 data point) will be used for training to determine how robust the methodology can be in terms of capturing the relationship between pore pressure and resistivity/porosity data. This approach is similar to that done in Eaton (1975) in order to observe performance as a function of Eaton's coefficient k. Table 4.3 shows the statistical summary of the data used for this work.

Table 4.3 Summary of all field data used in studying the robust nature of the methodologies

Number of data points: 24	Depth (ft. below sea level)	Normal Resistivity (ohm m)	Actual Resistivity (ohm m)	Normal Porosity	Actual Porosity	OBG (ppg)	RFT (ppg)
Arithmetic mean	10578	1.835	1.132	0.065	0.193	15.221	11.543
Harmonic mean	10492	1.833	1.126	0.062	0.192	15.216	11.529
Standard deviation	941	0.058	0.079	0.015	0.013	0.296	0.406
Minimum Value	9162	1.748	1.000	0.048	0.176	14.768	11.115
Maximum Value	11729	1.907	1.223	0.089	0.209	15.571	12.222
Mode	None	None	1.17707	None	None	None	None
Skewness	-0.310	-0.288	-0.563	0.448	-0.419	-0.366	0.579
Kurtosis	-1.573	-1.574	-1.457	-1.539	-1.796	-1.580	-1.374
R square (with respect to RFT)	0.755	0.761	0.621	0.714	0.031	0.738	1.000

The summary of results obtained using Table 4.3 is presented as Table 4.4. The detailed results from which Table 4.4 was obtained can be found in Section 4.10

Table 4.4 Best performance values obtained using the approach developed in this dissertation in comparison to conventional approaches for all field data (approximated to two decimal places)

	Resistivity data only			Resistivity and Porosity data	
	Cementation-exponent approach	Eaton's approach	Foster and Whelan's approach	Cementation-exponent approach	Conventional approach (Equation (40))
Minimum MAPE (%)	2.26	3.02	6.45	1.28	1.61
Minimum RMSE	0.31	0.42	0.84	0.20	0.24
Maximum R square	0.86	0.85	0.60	0.86	0.86

Table 4.4 shows that the cementation-exponent approach performed best, closely followed by Eaton's approach.

4.5.4 Dividing data into training and testing to demonstrate field application

As shown in Fig. 4.21, the first 75% of the RFT data is used for training and the last 25% is used for testing. This shows a typical field application in which data initially acquired is first used to develop a model; thereafter, the developed model is then used for prediction. Section 2 shows the typical fraction of total data used for training, which varied from around 70% to 90%.

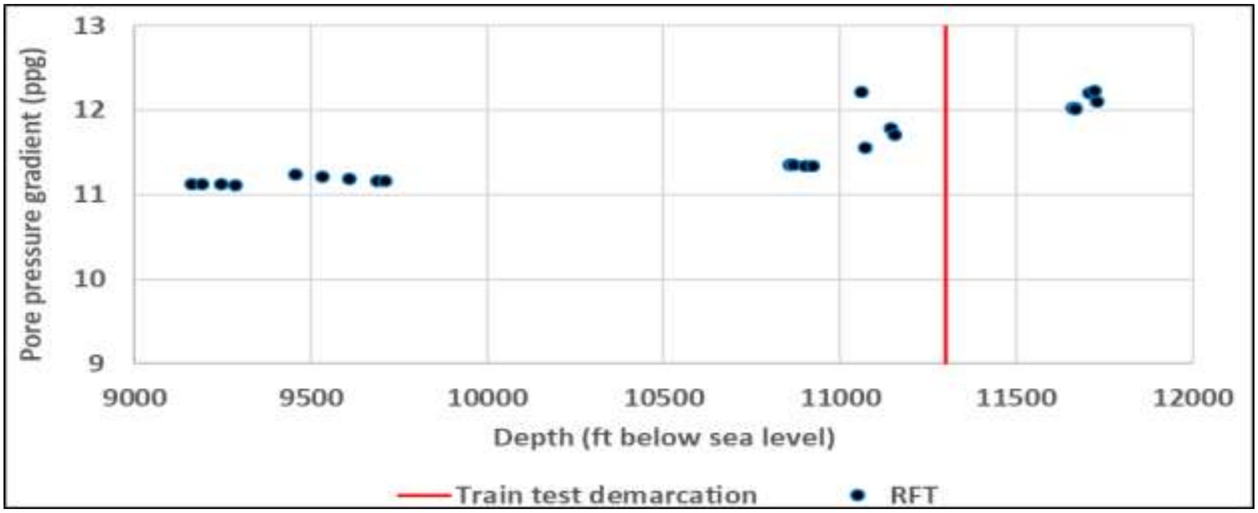

Fig. 4.6 Separating RFT data into training and testing part

The statistical summary of the data used for training and testing are presented in Table 4.5 and Table 4.6 respectively.

Table 4.5 Statistics of data used for training

Number of data points: 18	Depth (ft. below sea level)	Normal Resistivity (ohm m)	Actual Resistivity (ohm m)	Normal Porosity	Actual Porosity	OBG (ppg)	RFT (ppg)
Arithmetic mean	10207	1.812	1.106	0.070	0.191	15.108	11.358
Harmonic mean	10145	1.810	1.101	0.068	0.189	15.104	11.351
Standard deviation	793	0.049	0.074	0.013	0.015	0.256	0.283
Minimum Value	9162	1.748	1.000	0.055	0.176	14.768	11.115
Maximum Value	11154	1.870	1.177	0.089	0.209	15.409	12.216
Mode	None	None	1.17707	None	None	None	None
Skewness	-0.070	-0.062	-0.192	0.124	0.024	-0.082	1.828
Kurtosis	-2.039	-2.044	-1.963	-1.988	-2.220	-2.037	3.485
R square (with respect to RFT)	0.554	0.556	0.463	0.537	0.454	0.547	1.000

Table 4.6 Statistics of data used for testing

Number of data points: 6	Depth (ft. below sea level)	Normal Resistivity (ohm m)	Actual Resistivity (ohm m)	Normal Porosity	Actual Porosity	OBG (ppg)	RFT (ppg)
Arithmetic mean	11692	1.9048	1.2122	0.0484	0.1990	15.5614	12.0975
Harmonic mean	11692	1.9048	1.2121	0.0484	0.1990	15.5614	12.0969
Standard deviation	28	0.0018	0.0101	0.0003	0.0005	0.0072	0.0863
Minimum Value	11660	1.9027	1.2018	0.0479	0.1983	15.5530	12.0180
Maximum Value	11729	1.9072	1.2229	0.0487	0.1995	15.5713	12.2222
Mode	None	None	None	None	None	None	None
Skewness	0.1682	0.1686	0.0024	-0.1653	-0.1682	0.1861	0.5538
Kurtosis	-2.6597	-2.6591	-3.3198	-2.6643	-2.6597	-2.5829	-2.1582
R square (with respect to RFT)	0.6645	0.6643	0.8201	0.6656	0.6645	0.6522	1.0000

The summary of the results obtained using data of Table 4.5 for training and Table 4.6 for testing is shown in Table 4.7. Section 4.11 and Section 4.12 shows how training was done.

Table 4.7 Results of pore pressure prediction with test data after training

		Resistivity data only			Resistivity and Porosity data		
		Cementation exponent approach	Eaton's approach	Foster and Whelan's approach	Cementation-exponent approach	Conventional approach (Equation (40))	Artificial Neural Network
MAPE (%)	Training minimum	1.35	2.15	4.77	0.89	1.09	0.64
	Testing	5.11	7.05	13.22	2.64	3.27	0.85
RMSE	Training minimum	0.24	0.32	0.64	0.20	0.22	0.14
	Testing	0.55	0.76	1.38	0.24	0.33	0.12
R square	Training maximum	0.64	0.64	0.46	0.62	0.62	0.75
	Testing	0.79	0.79	0.82	0.77	0.79	0.87

From Table 4.7, it can be observed that the cementation-exponent approach performed better than conventional approaches due to minimum error (MAPE and RMSE) achieved. However, from the same Table 4.7, it can be observed that ANN performed best.

4.5.5 Discussion

Results in Section 4.5 shows that the approach by Foster & Whalen (1965) performed poorer than the approach by Eaton (1975). This could be one of the reasons why the approach by Eaton is more popular than that by Foster & Whalen. Even though the cementation exponent approach slightly performed better than Eaton's approach (≈ 0.7% for resistivity approach in Table 4.4) when all data was used, splitting the data into training and testing shows that the cementation-exponent approach even performed better by a wider margin ($\approx 2.0\%$ for resistivity approach in Table 4.7).

The cementation-exponent approach shows better performance than conventional approach when only resistivity was considered and when both resistivity and sonic porosity were considered. A valid argument is that the superior performance of the cementation-exponent approach when both resistivity and sonic porosity were considered was chiefly due to the resistivity component of the cementation-exponent approach. Based on this, a quick test was done to determine the performance of simply averaging the resistivity arm of the cementation-exponent approach with the porosity approach by Zhang (2011). This resulted in a slight drop in MAPE performance by ≈ 0.2 % in comparison to the complete cementation-exponent approach developed in this article for combining resistivity and porosity for pore pressure prediction (Equation (4.24)). This slight change in performance may not be sufficient to come up with a valid conclusion. However, the results obtained in this work encourage more test with field data, especially at depths where the effect of cementation is clear. Another reason why testing with different field data is recommended, is to be able to obtain a universal value for resistivity scaling factor. This will make it possible for an estimated pore pressure prediction to be carried out in new fields without drilling data.

Machine learning indeed represents a recommended means of performing prediction when sufficient training data is available. The development of simplified pore pressure equations, such as Equations (4.3),(4.5),(4.21) and (4.24), are recommended because they can be easily used in a field with limited data. For example, in a new field where resistivity data is to be used for pore pressure prediction, a value of 1.2 for Eaton's coefficient can be used when data is limited (as shown in Section 4.2). The value of Eaton's coefficient can then be fine-tuned as more data for the well/field is obtained. The acceptability of the Eaton's coefficient value of 1.2 as a good approximation can be observed in several articles where it has been used in different fields (Nhabanga & Ringrose, 2019; Ugwu & Nwankwo, 2014; Zhang, 2011). In the case of machine learning, researchers are beginning to develop equations from trained data that can be utilized for pore pressure. For example, in Ahmed et al., (2019), an equation for pore pressure prediction was developed by training ANN with the following input: density, porosity, compressional sonic time, weight on bit, drilling rotational speed in revolutions per minute, drilling rate of penetration and mud weight. In the same article by Ahmed et al., (2019), 245 data points from one well were available with 70% used for training and 30% for testing. The MAPE for both training and testing were less than 0.3%. Although the use of ANN has shown to be successful, there is the need for developed ANN equations to be tested with data from a different field in order to observe their robustness.

Some limitations that should be considered with respect to the methodologies presented in this work are:

1. The porosity utilized in this work was derived from sonic log. Other possible means of deriving porosity log include neutron and density log (Kamel & Mabrouk, 2003). Porosity log developed from different sources (sonic, neutron etc.,) could significantly perform differently for pore pressure prediction (Tingay et al., 2009).
2. For the field data used in testing the methodologies in this work, compaction disequilibrium was the main cause of overpressure. Other mechanisms that cause overpressure are fluid expansion or transfer mechanism (Tingay et al., 2009). The value of empirical coefficients can be influenced by the overpressure causing mechanism. For example, in the article by Bowers, (1995), Eaton's coefficient was given a value far from

what was typically recommended in order to achieve good prediction. This is because fluid expansion was the main cause of overpressure.

4.6 Conclusions

This chapter presents two methodologies for pore pressure prediction; one of the methodologies utilizes only resistivity data and the other utilizes both resistivity and porosity data for pore pressure prediction. The aim of the research was to present methodologies which can perform better than popular means of utilizing resistivity or a combination of resistivity and porosity for pore pressure prediction.

Effective stress coefficient influences pore pressure prediction and our literature review showed that variation in effective stress coefficient from its traditional value of 1 can be expected in several reservoirs. Several factors such as cementation effect, shaliness can influence effective stress coefficient which would in turn affect pore pressure prediction. Based on this, the porosity-resistivity approach developed in this article was fashioned to predict pore pressure as a function of changes in Archie's cementation exponent. This approach was termed cementation-exponent approach.

Field data was used to test the cementation-exponent approach against conventional means of pore pressure prediction. Two sets of analyses were performed. The first set of tests were aimed at observing the capability of the cementation-exponent approach to capture the relationship between pore pressure and resistivity or a combination of resistivity and porosity. The second set of analysis was done to observe how the cementation-exponent approach can accurately predict pore pressure when part of the field data was used for training/developing and the other part was used for testing/predicting. The results of these two sets of analyses showed that the cementation-exponent approach performed better than their conventional counterparts in terms of mean absolute percentage error and root mean square error. Machine learning was also utilized to explore the interrelationship between resistivity and porosity data for pore pressure prediction. Artificial neural network was utilized as the machine learning approach and the following inputs were used in training the ANN: normal shale resistivity, observed shale resistivity from log, normal shale porosity, observed shale porosity obtained from logging parameter, overburden gradient, and depth. Field data was also used for training and testing with ANN. The results from ANN testing performed best. Thus, this chapter also demonstrated that ANN can be used as a viable means of pore pressure prediction.

The pore pressure prediction methodologies (cementation-exponent approach) developed in this article introduces a parameter that is termed resistivity scaling factor. It is recommended that additional research be carried out to determine the most suitable values for the parameter. Although the methodologies presented in this article were successfully tested with data from one field, more testing is recommended to gain deeper insight about their performance at great reservoir depth where cementation effect is expected to be more significant.

4.7 Nomenclature

α	Effective stress coefficient
δ	Tortuosity factor
c	Compaction constant (ft^{-1})
C_v	Compressibility of the total volume (psi^{-1})

C_p	Pore compressibility (psi^{-1})
C_s	Compressibility of the solid material (psi^{-1})
D	Gain equalization factor
Δt_n	Normally expected transit time (microsecond/ft)
Δt_o	Observed transit time (microsecond/ft)
k	Eaton's coefficient
$log(S)$	Slope of the formation factor vs depth plot (ft^{-1})
m	Archie's cementation exponent
m_n	Normal Archie's cementation exponent
m_o	Observed Archie's cementation exponent
ρ	Formation bulk density (g/cm^3)
ρ_f	Density of fluid (g/cm^3)
ρ_m	Density of framework of rock (g/cm^3)
ρ_p	Equivalent density of formation pore pressure (g/cm^3)
P	Actual pore pressure to be predicted (psi)
P_n	Normal/hydrostatic pore pressure (psi)
R_n	Normal shale resistivity (ohm-m)
R_o	Observed shale resistivity (ohm-m)
R_{sf}	Resistivity scaling factor (ohm-m)
R_w	Formation water resistivity (ohm-m)
S_e	Vertical effective stress (psi)
S_{eMax}	Vertical effective stress at the onset of unloading (psi)
S_v	Overburden stress (psi)

T Temperature (°C)

φ Porosity

φ_g Mudline porosity

φ_n Normal porosity

φ_o Shale porosity obtained from logging parameter

v_{sh} Shale volume

V Sonic velocity (kft/s)

V_f Sonic velocity in the formation fluid (kft/s)

V_m Sonic velocity in the shale matrix (kft/s)

V_{ml} Sonic velocity at ground surface (kft/s)

V_p Sonic compressional velocity (kft/s)

V_{Max} Velocity at the onset of unloading (kft/s)

λ Empirically obtained parameter which defines the rate of velocity increase with effective stress

x_o/x_n Ratio of observed to the normally expected pore pressure indicating parameter

Z True vertical depth (ft)

4.8 Acronyms

ANN artificial neural network

MAPE mean absolute percentage error

NHPG normal hydrostatic pressure gradient

OBG overburden gradient

R square	Pearson product moment correlation coefficient
RMSE	root mean square error
RFT	measured pore pressure gradient from the repeat formation tests

4.9 APPENDIX A: Deriving Pore pressure as a function of Archie's cementation coefficient

Equation (4.A1) is the same as Equation (4.34).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{log_e\left(\frac{\varphi_g}{\varphi_n}\right)} \times \frac{log_e\left(\frac{R_{sf}}{R_n}\right)}{log_e\left(\frac{R_{sf}}{R_o}\right)}\right)^{\frac{1}{2}}, \quad (4.A1)$$

Equation (4.A1) can also be written as Equation (4.A2) using the law that $1 = -1 \div -1$,

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\frac{-log_e\left(\frac{\varphi_g}{\varphi_o}\right)}{-log_e\left(\frac{\varphi_g}{\varphi_n}\right)} \times \frac{log_e\left(\frac{R_{sf}}{R_n}\right)}{log_e\left(\frac{R_{sf}}{R_o}\right)}\right)^{\frac{1}{2}}. \quad (4.A2)$$

Applying the laws of logarithm and indices transforms Equation (4.A2) into Equation (4.A3) and Equation (4.A4).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\frac{log_e\left(\frac{\varphi_g}{\varphi_o}\right)^{-1}}{log_e\left(\frac{\varphi_g}{\varphi_n}\right)^{-1}} \times \frac{log_e\left(\frac{R_{sf}}{R_n}\right)}{log_e\left(\frac{R_{sf}}{R_o}\right)}\right)^{\frac{1}{2}}, \quad (4.A3)$$

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\frac{log_e\left(\frac{\varphi_o}{\varphi_g}\right)}{log_e\left(\frac{\varphi_n}{\varphi_g}\right)} \times \frac{log_e\left(\frac{R_{sf}}{R_n}\right)}{log_e\left(\frac{R_{sf}}{R_o}\right)}\right)^{\frac{1}{2}}. \quad (4.A4)$$

Rearranging Equation (4.A4) results in Equation (4.A5).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\frac{log_e\left(\frac{\varphi_o}{\varphi_g}\right)}{log_e(\frac{R_{sf}}{R_o})} \times \frac{log_e(\frac{R_{sf}}{R_n})}{log_e(\frac{\varphi_n}{\varphi_g})}\right)^{\frac{1}{2}}. \quad (4.A5)$$

Applying the laws of logarithm transforms Equation (4.A5) into Equation (4.A6).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\frac{log_e(\varphi_o) - log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_o})} \times \frac{log_e(\frac{R_{sf}}{R_n})}{log_e(\varphi_n) - log_e(\varphi_g)}\right)^{\frac{1}{2}}. \quad (4.A6)$$

Equation (4.A6) can be mathematically written as Equation (4.A7) and Equation (4.A8).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\frac{log_e(\varphi_o) - log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_o})} \div \frac{log_e(\varphi_n) - log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_n})}\right)^{\frac{1}{2}}, \quad (4.A7)$$

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\left(\frac{log_e(\varphi_o)}{log_e(\frac{R_{sf}}{R_o})} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_o})}\right) \div \left(\frac{log_e(\varphi_n)}{log_e(\frac{R_{sf}}{R_n})} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_n})}\right)\right)^{\frac{1}{2}}. \quad (4.A8)$$

Equation (4.A9) is the same as Equation (4.27).

$$\frac{R_o}{R_w} = \frac{\delta}{\varphi_o^m}. \quad (4.A9)$$

Equation (4.A10) can be obtained from Equation (4.A9) by making m the subject of Equation (4.A9).

$$m = \frac{log_e(\delta \times R_w / R_o)}{log_e(\varphi_o)}. \quad (4.A10)$$

If R_{sf} of Equation (4.A8) is equal to $(\delta \times R_w)$ of Equation (4.A10), then Equation (4.A11) and Equation (4.A12) can presented based on Equation (4.A10).

$$m_o = \frac{log_e(R_{sf} / R_o)}{log_e(\varphi_o)}, \quad (4.A11)$$

$$m_n = \frac{log_e(R_{sf} / R_n)}{log_e(\varphi_n)}. \quad (4.A12)$$

Equation (4.A11) and Equation (4.A12) can be substituted into Equation (4.A8) to yield Equation (4.A13).

$$\frac{P}{Z} = \frac{S_v}{Z} - \left(\frac{S_v}{Z} - \frac{P_n}{z}\right)\left(\left(\frac{1}{m_o} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_o})}\right) \div \left(\frac{1}{m_n} - \frac{log_e(\varphi_g)}{log_e(\frac{R_{sf}}{R_n})}\right)\right)^{\frac{1}{2}}. \quad (4.A13)$$

4.10 APPENDIX B: MAPE, RMSE and R square performance obtained by using all data for training with summary presented in Table 4.4

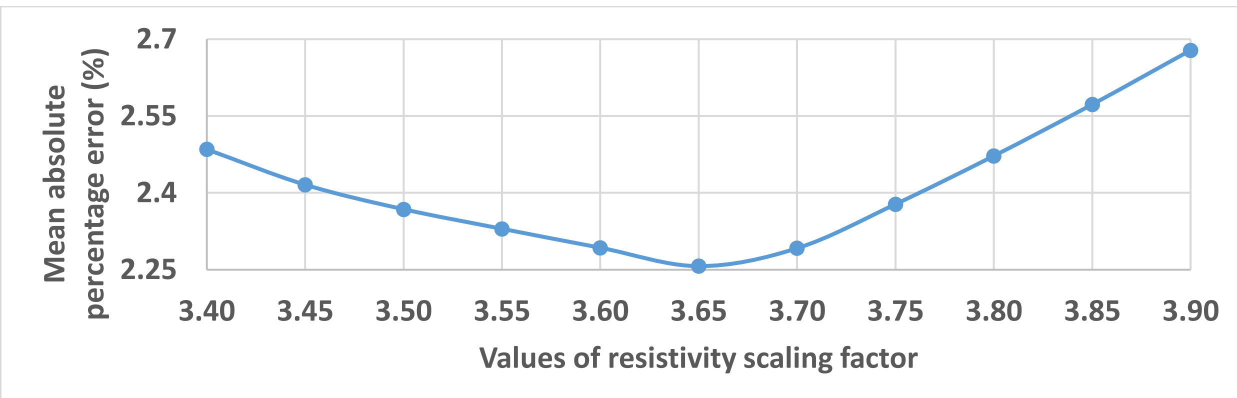

Fig. 4.7 MAPE performance of the resistivity component of the cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

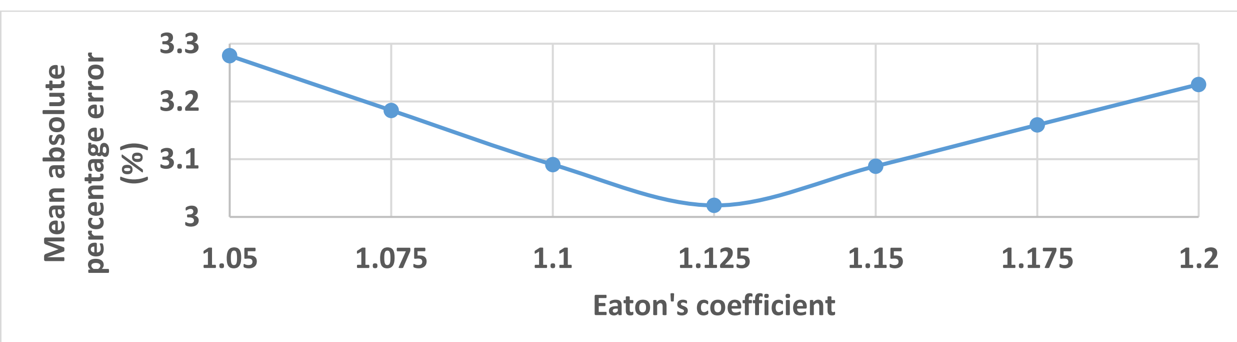

Fig. 4.8 MAPE performance of Eaton's resistivity approach as a function of Eaton's coefficient (k)

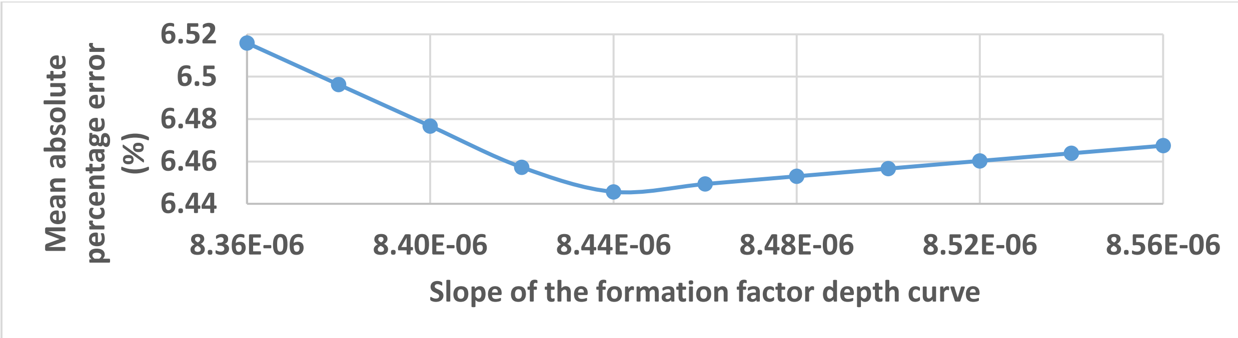

Fig. 4.9 MAPE performance of Foster & Whelan's resistivity approach as a function of the Slope of formation factor depth curve

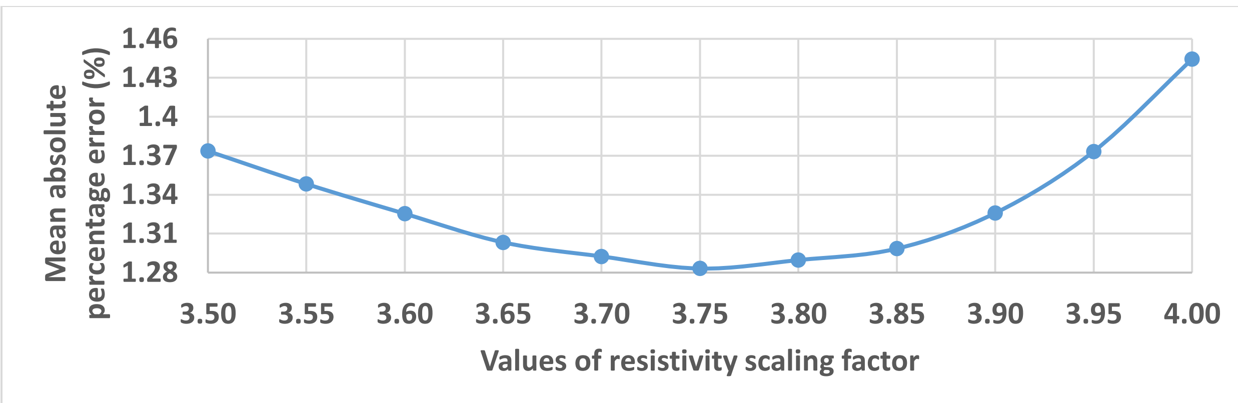

Fig. 4.10 MAPE performance of cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

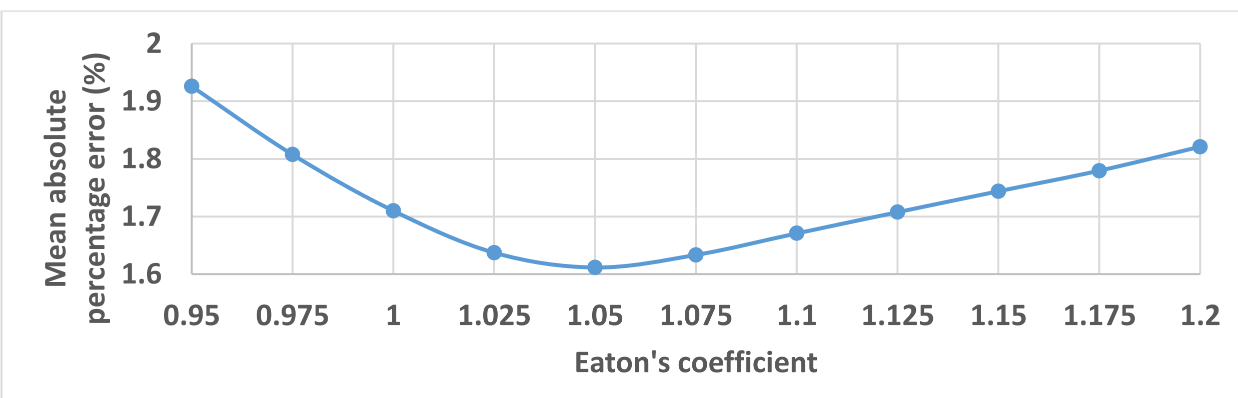

Fig. 4.11 MAPE performance of conventional simple averaging approach as a function of Eaton's coefficient (k)

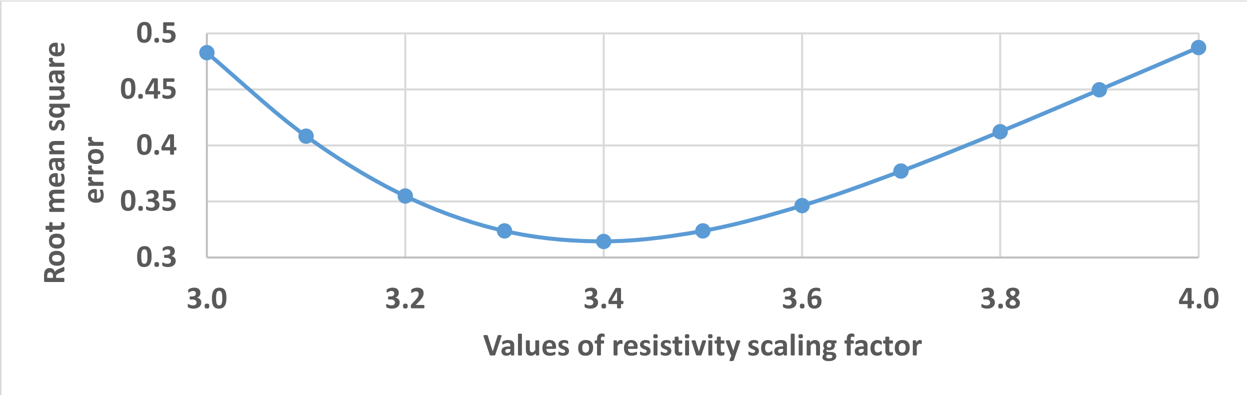

Fig. 4.12 RMSE performance of the resistivity component of the cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

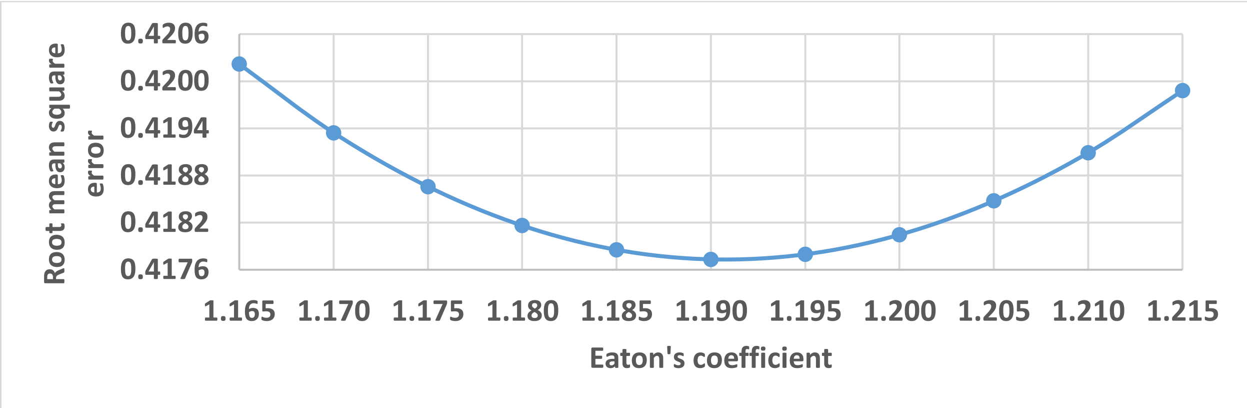

Fig. 4.13 RMSE performance of Eaton's resistivity approach as a function of Eaton's coefficient (k)

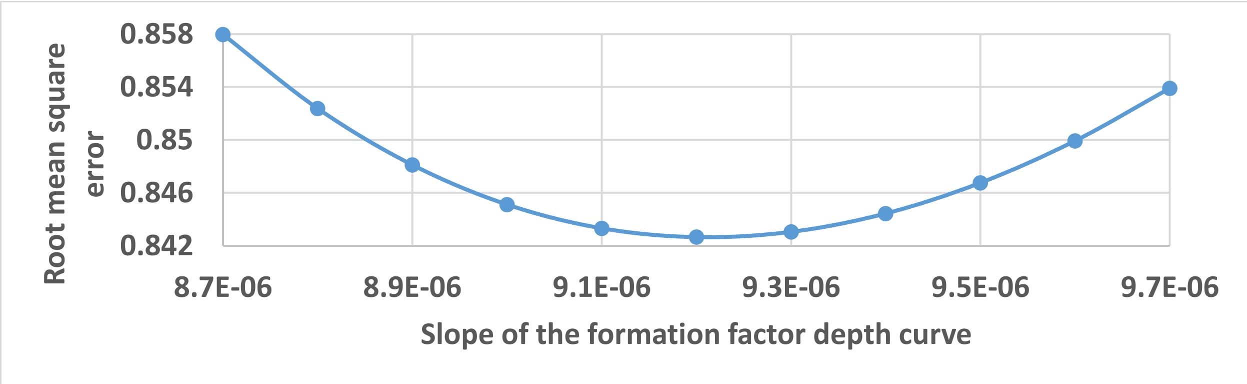

Fig. 4.14 RMSE performance of Foster & Whelan's resistivity approach as a function of the Slope of formation factor depth curve

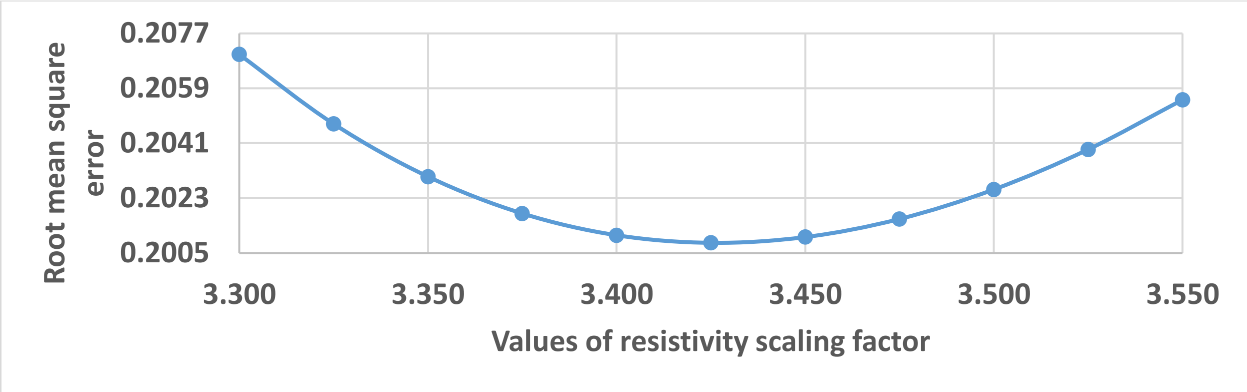

Fig. 4.15 RMSE performance of cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

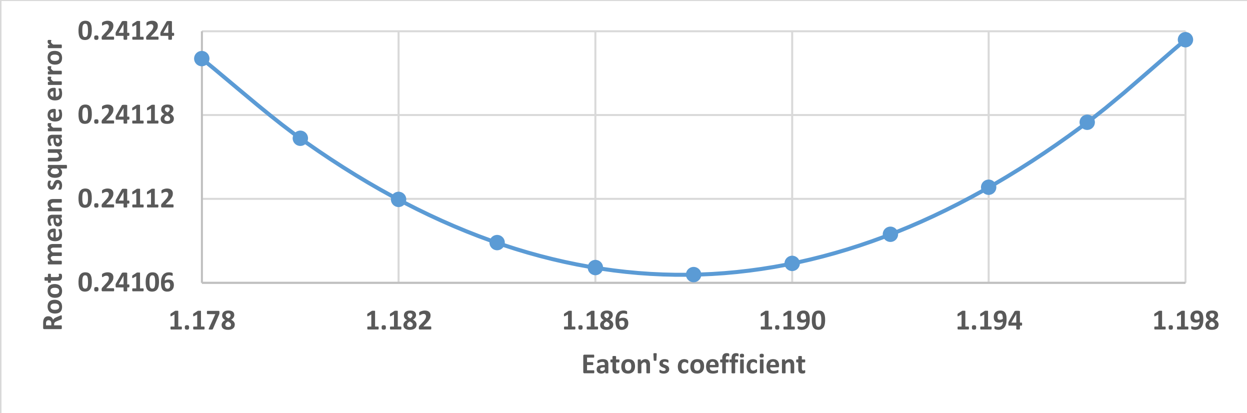

Fig. 4.16 RMSE performance of conventional simple averaging approach as a function of Eaton's coefficient (k)

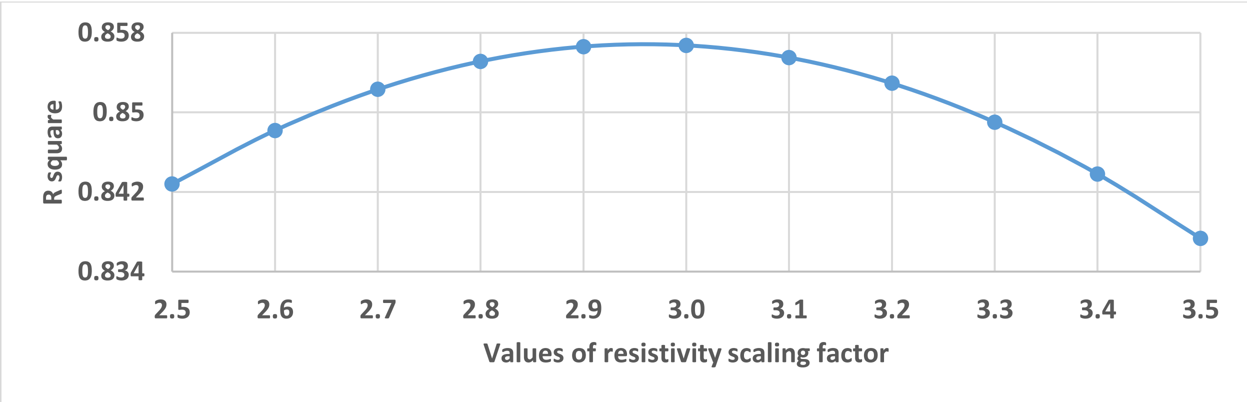

Fig. 4.17 R square performance of the resistivity component of the cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

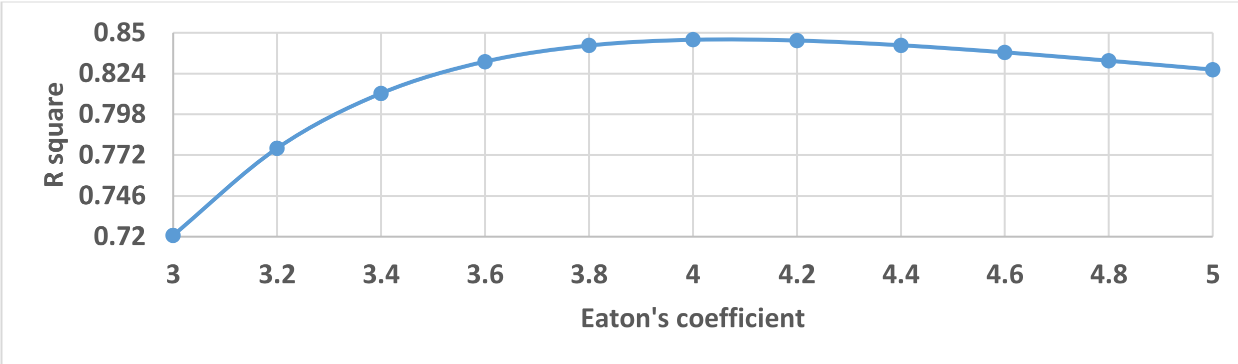

Fig. 4.18 R square performance of Eaton's resistivity approach as a function of Eaton's coefficient (k)

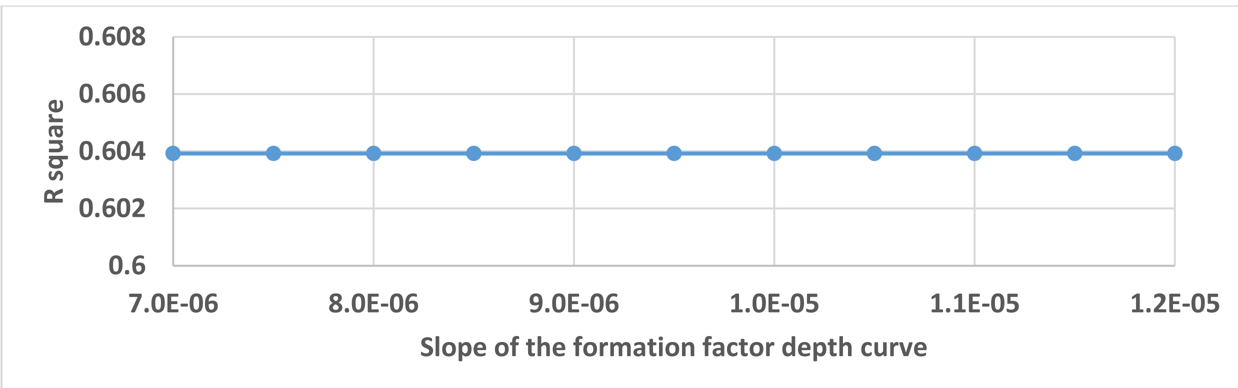

Fig. 4.19 R square performance of Foster & Whelan's resistivity approach as a function of the Slope of formation factor depth curve

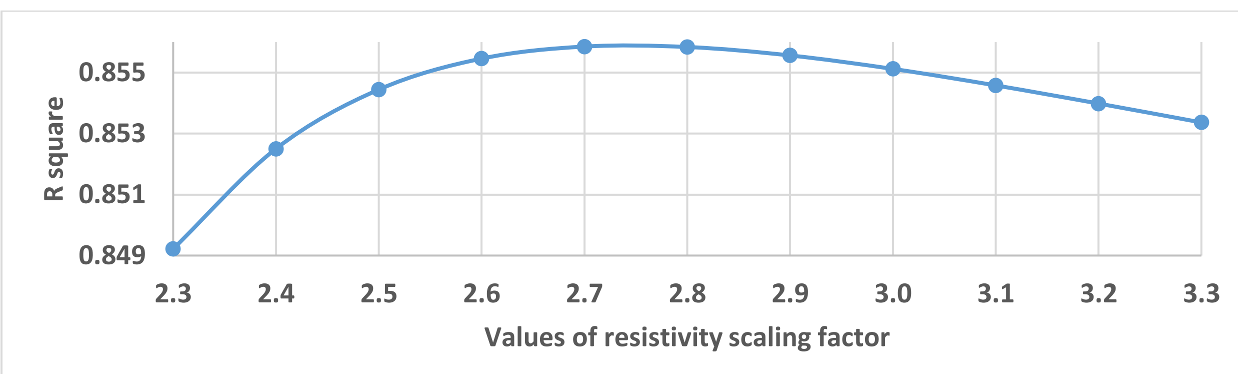

Fig. 4.20 R square performance of cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

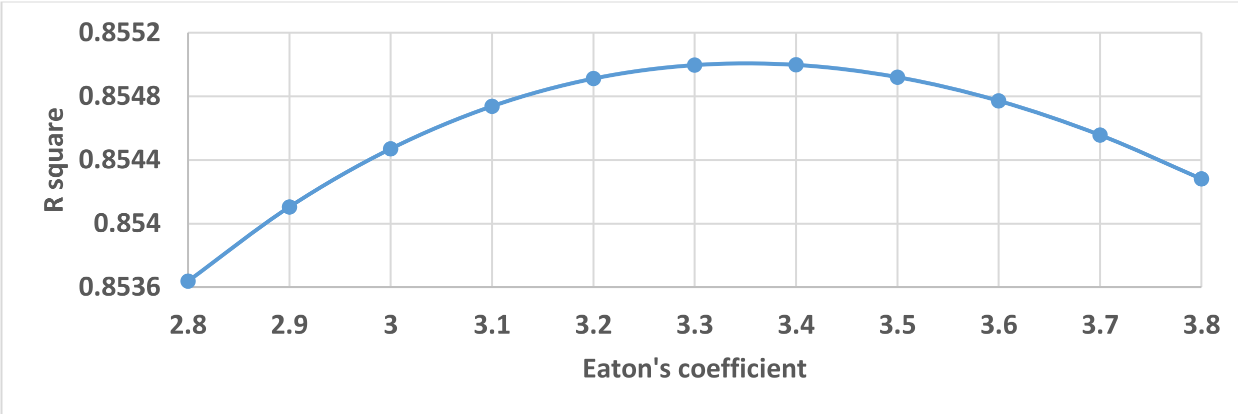

Fig. 4.21 R square performance of conventional simple averaging approach as a function of Eaton's coefficient (k)

4.11 APPENDIX C: MAPE, RMSE and R square performance obtained by using 75% of data for training with summary presented in Table 4.7

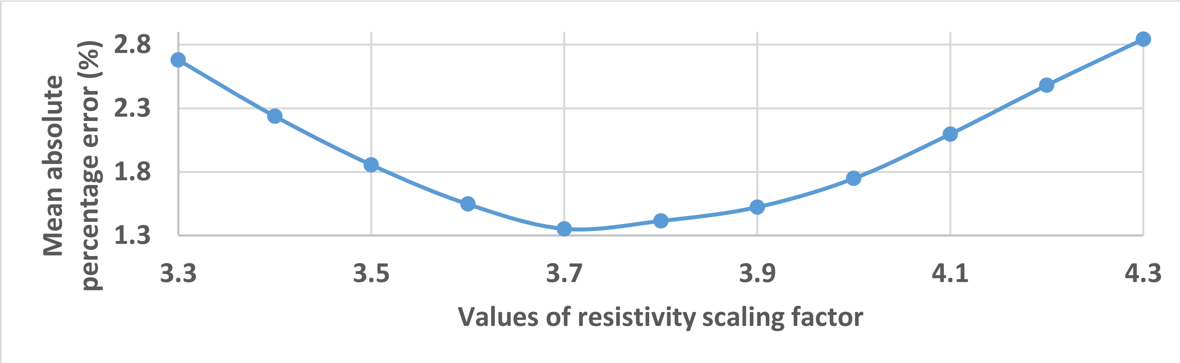

Fig. 4.22 MAPE training performance of the resistivity component of the cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

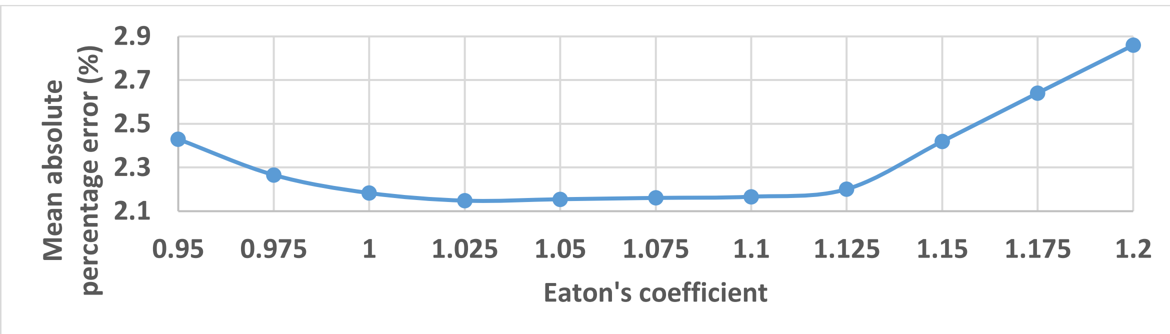

Fig. 4.23 MAPE training performance of Eaton's resistivity approach as a function of Eaton's coefficient (k)

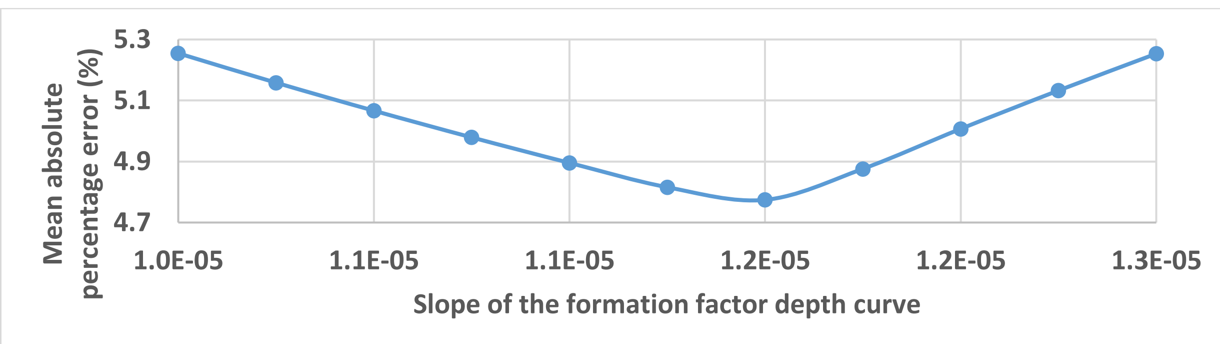

Fig. 4.24 MAPE training performance of Foster & Whelan's resistivity approach as a function of the Slope of formation factor depth curve

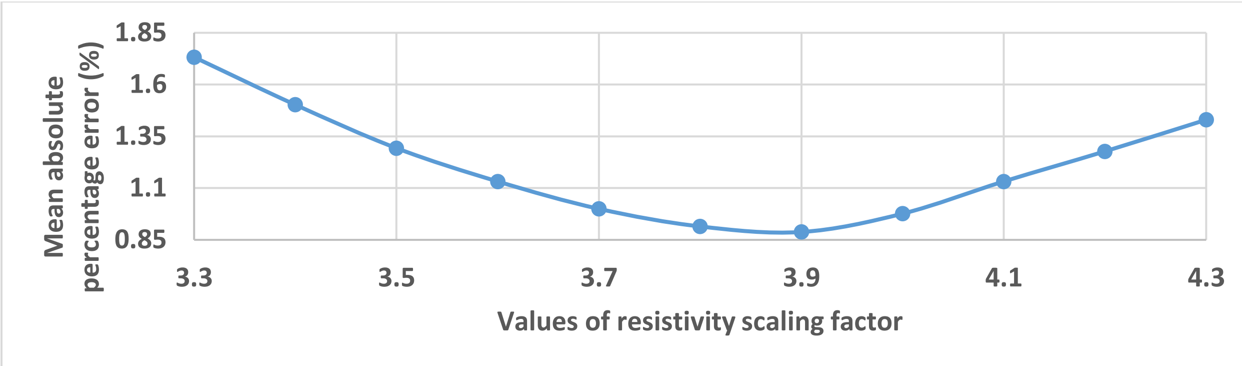

Fig. 4.25 MAPE training performance of cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

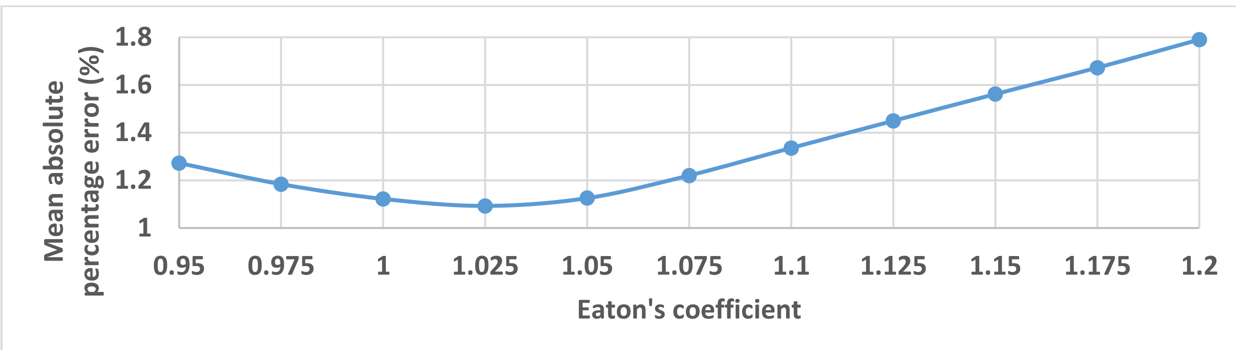

Fig. 4.26 MAPE training performance of conventional simple averaging approach as a function of Eaton's coefficient (k)

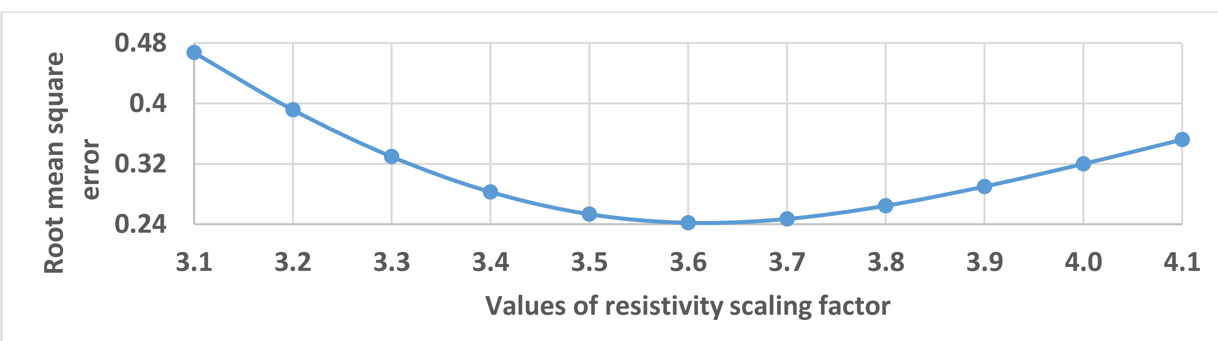

Fig. 4.27 RMSE training performance of the resistivity component of the cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

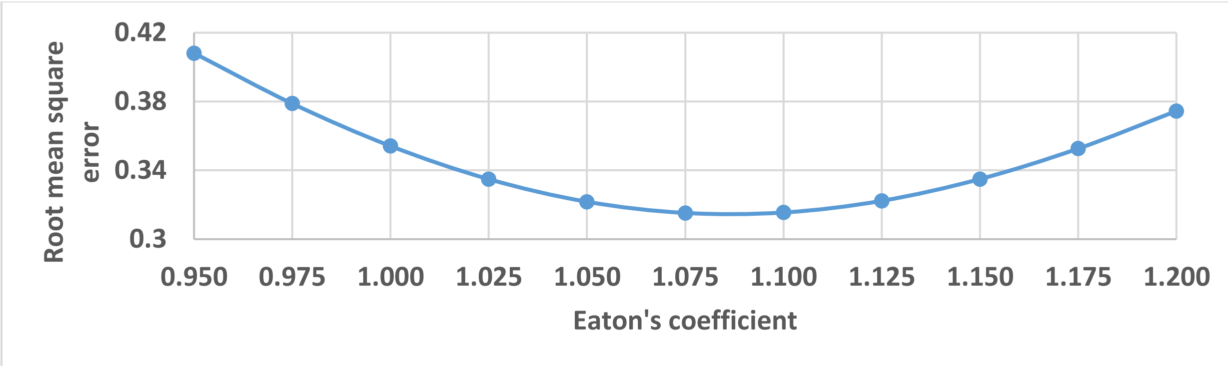

Fig. 4.28 RMSE training performance of Eaton's resistivity approach as a function of Eaton's coefficient (k)

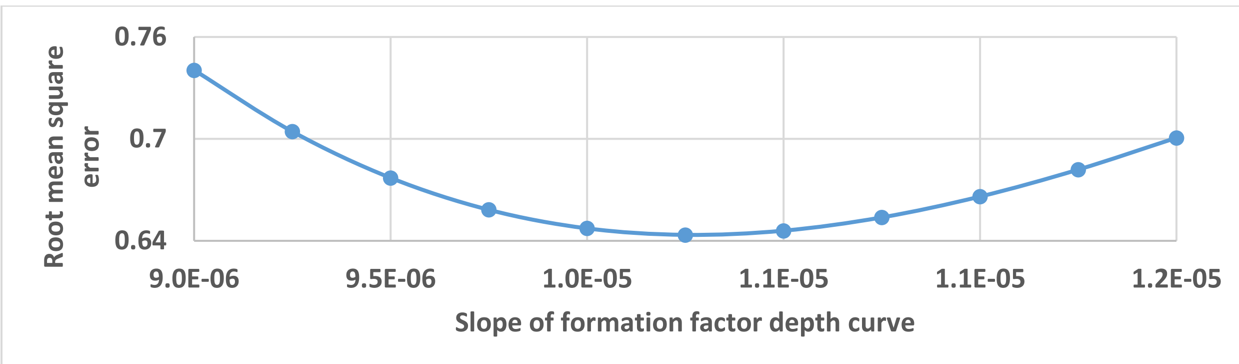

Fig. 4.29 RMSE training performance of Foster & Whelan's resistivity approach as a function of the Slope of formation factor depth curve

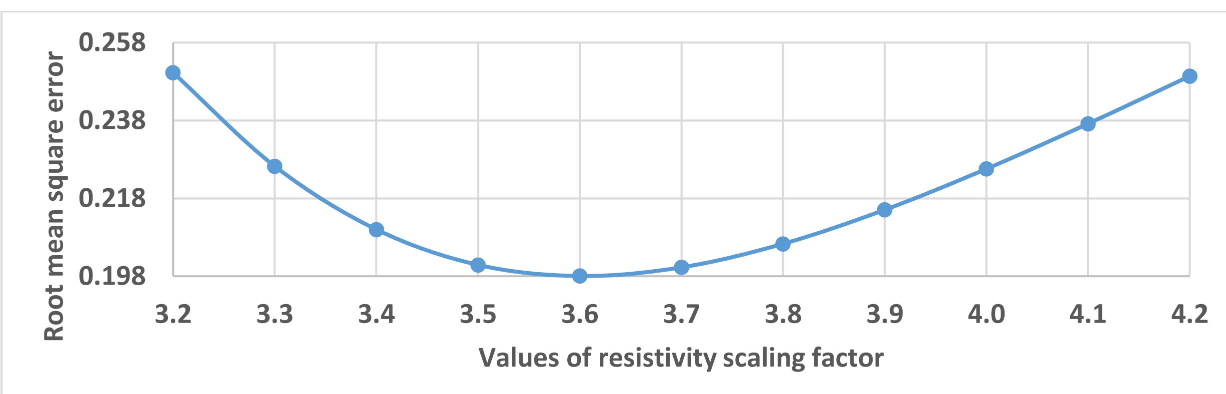

Fig. 4.30 RMSE training performance of cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

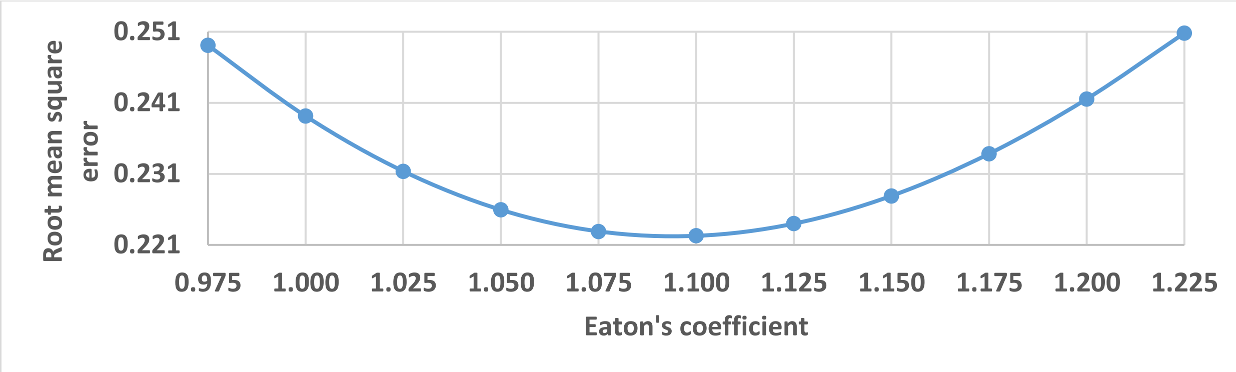

Fig. 4.31 RMSE training performance of conventional simple averaging approach as a function of Eaton's coefficient (k)

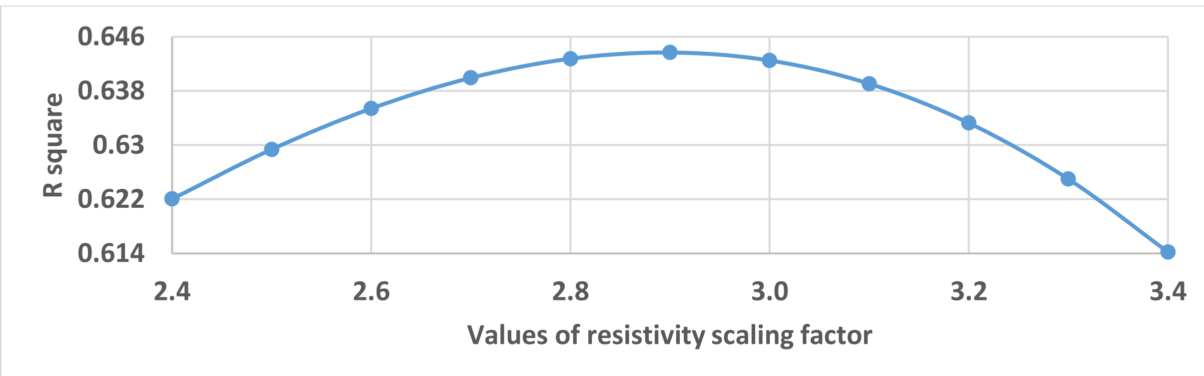

Fig. 4.32 R square training performance of the resistivity component of the cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

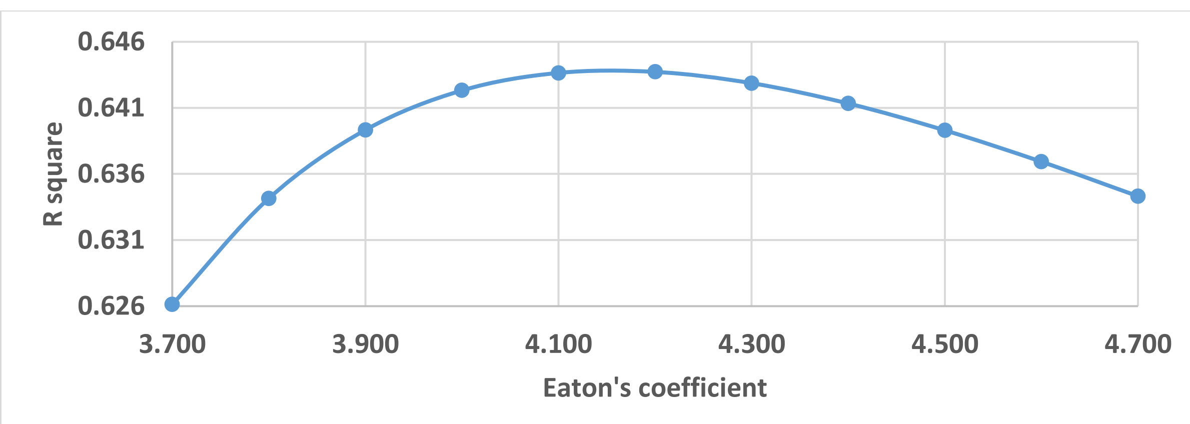

Fig. 4.33 R square training performance of Eaton's resistivity approach as a function of Eaton's coefficient (k)

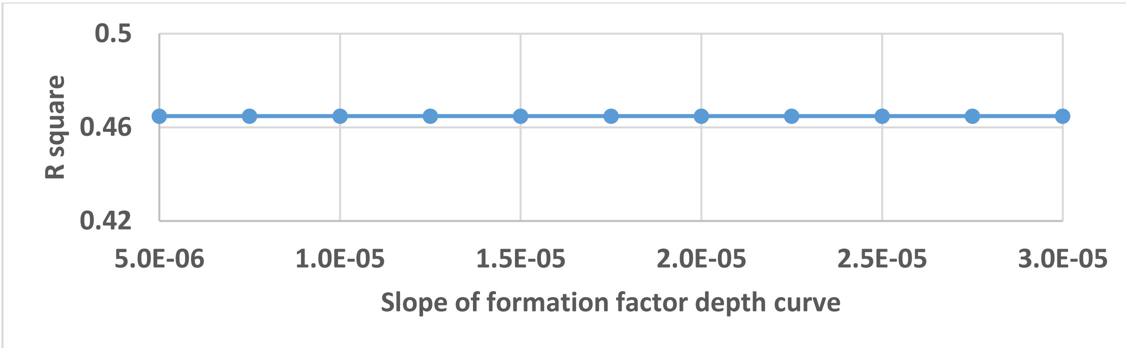

Fig. 4.34 R square training performance of Foster & Whelan's resistivity approach as a function of the Slope of formation factor depth curve

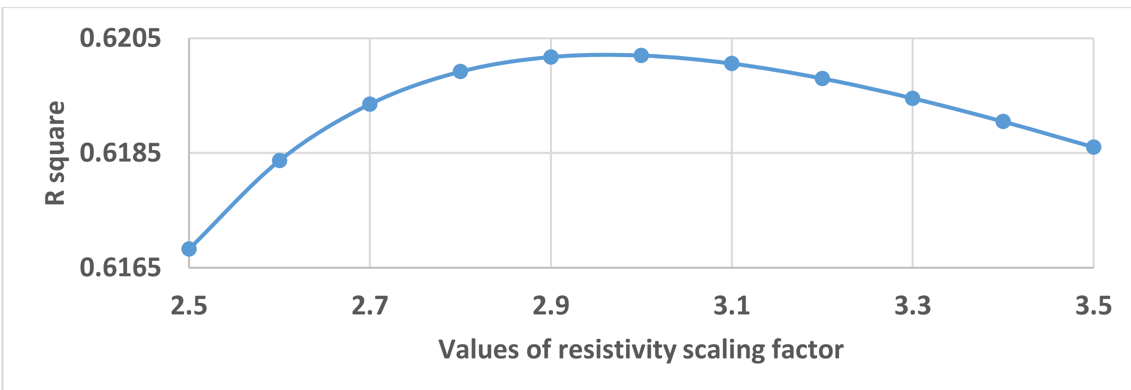

Fig. 4.35 R square training performance of cementation-exponent approach as a function of resistivity scaling factor (R_{sf})

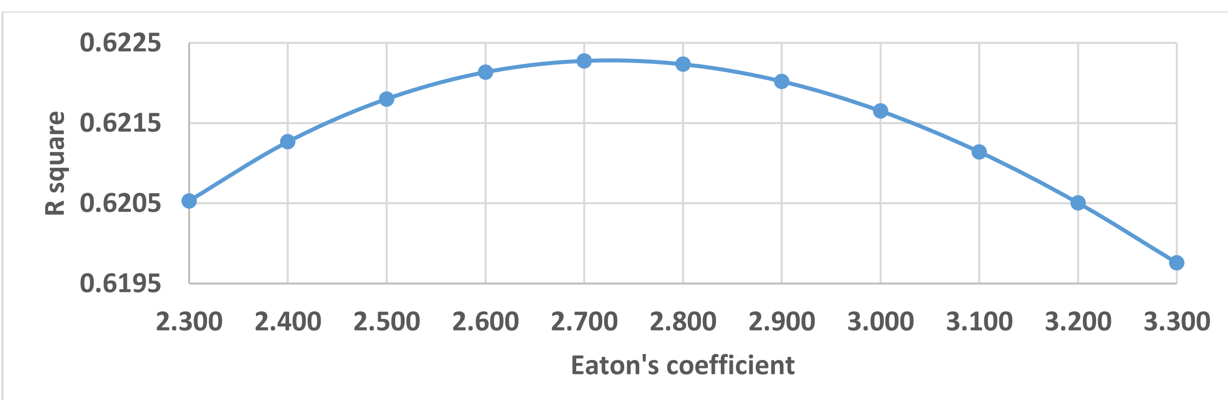

Fig. 4.36 R square training performance of conventional simple averaging approach as a function of Eaton's coefficient (k)

4.12 APPENDIX D: ANN configuration and training performance with best result presented in Table 4.7

Table 4.8 ANN training performance for different configurations

ANN structure	MAPE (%)	RMSE	R square
One hidden layer with 5 nodes	0.64	0.14	0.75
One hidden layer with 10 nodes	0.65	0.15	0.74
One hidden layer with 20 nodes	0.66	0.15	0.74
One hidden layer with 30 nodes	0.66	0.15	0.74
Two hidden layer. 1st hidden layer has 20 nodes, second hidden layer has 10 nodes	1.33	0.22	0.74

4.10 References

Abimbola, M., Khan, F., Khakzad, N., & Butt, S. (2015). Safety and risk analysis of managed pressure drilling operation using Bayesian network. Safety Science, 76, 133–144.

Ahmed, A., Elkatatny, S., Ali, A., Mahmoud, M., & Abdulraheem, A. (2019). New model for pore pressure prediction while drilling using artificial neural networks. Arabian Journal for Science and Engineering, 44(6), 6079–6088.

Alam, M. M., Fabricius, I. L., & Christensen, H. F. (2012). Static and dynamic effective stress coefficient of chalk: Geophysics, 77. L1–L11.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. Electronics, 8(3), 292.

Amiri, M., Lashkaripour, G. R., Ghabezloo, S., Moghaddas, N. H., & Tajareh, M. H. (2019). 3D spatial model of Biot's effective stress coefficient using well logs, laboratory experiments, and geostatistical method in the Gachsaran oil field, southwest of Iran. Bulletin of Engineering Geology and the Environment, 78(6), 4633–4646.

Archie, G. E. (1942). The electrical resistivity log as an aid in determining some reservoir characteristics. Transactions of the AIME, 146(01), 54–62.

Azadpour, M., Manaman, N. S., Kadkhodaie-Ilkhchi, A., & Sedghipour, M.-R. (2015). Pore pressure prediction and modeling using well-logging data in one of the gas fields in south of Iran. Journal of Petroleum Science and Engineering, 128, 15–23.

Biot, M. A. (1941). General theory of three-dimensional consolidation. Journal of Applied Physics, 12(2), 155–164.

Bowers, G. L. (1995). Pore pressure estimation from velocity data: Accounting for overpressure mechanisms besides undercompaction. SPE Drilling & Completion, 10(02), 89–95.

Brahma, J., & Sircar, A. (2018). Design of safe well on the top of Atharamura anticline, Tripura, India, on the basis of predicted pore pressure from seismic velocity data. Journal of Petroleum Exploration and Production Technology, 8(4), 1209–1224.

Dassanayake, A. B. N., Fujii, Y., Fukuda, D., & Kodama, J. (2015). A new approach to evaluate effective stress coefficient for strength in Kimachi sandstone. Journal of Petroleum Science and Engineering, 131, 70–79.

Deng, S., Fan, H., Liu, Y., He, Y., Zhang, S., Yang, J., & Fu, L. (2017). Two innovative pore pressure calculation methods for shallow deep-water formations. Journal of Applied Geophysics, 146, 208–213.

Eaton, B. A. (1975). The equation for geopressure prediction from well logs. Fall Meeting of the Society of Petroleum Engineers of AIME.

Feng, Y., Arlanoglu, C., Podnos, E., Becker, E., & Gray, K. E. (2015). Finite-element studies of hoop-stress enhancement for wellbore strengthening. SPE Drilling & Completion, 30(01), 38–51.

Foresee, F. D., & Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. Proceedings of International Conference on Neural Networks (ICNN'97), 3, 1930–1935.

Foster, J. B., & Whalen, H. E. (1965). Estimation of Formation Pressures From Electrical Surveys-Offshore Louisiana. JOURNAL OF PETROLEUM TECHNOLOGY, 17(9), 1062.

Frempong, P. K., & Butt, S. D. (2006). Non-destructive techniques to determine the effective stress coefficient of sandstone formations. Journal of Canadian Petroleum Technology, 45(07).

Glover, P. (2009). What is the cementation exponent? A new interpretation. The Leading Edge, 28(1), 82–85.

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). Neural network design. PWS Publishing Co.

Kamel, M. H., & Mabrouk, W. M. (2003). Estimation of shale volume using a combination of the three porosity logs. Journal of Petroleum Science and Engineering, 40(3–4), 145–157.

Liu, L., Shen, G., Wang, Z., Yang, H., Han, H., & Cheng, Y. (2018). Abnormal formation velocities and applications to pore pressure prediction. Journal of Applied Geophysics, 153, 1–6.

Luo, X., Were, P., Liu, J., & Hou, Z. (2015). Estimation of Biot's effective stress coefficient from well logs. Environmental Earth Sciences, 73(11), 7019–7028.

Magara, K. (1980). Comparison of porosity-depth relationships of shale and sandstone. Journal of Petroleum Geology, 3(2), 175–185.

Mao, X., Liu, Y., Guan, W., Liu, S., & Li, J. (2018). A new effective stress constitutive equation for cemented porous media based on experiment and derivation. Arabian Journal of Geosciences, 11(13), 337.

Mathwork. (2020). 1-D data interpolation (table lookup). In https://www.mathworks.com/help/matlab/ref/interp1.html.

Nhabanga, O. J., & Ringrose, P. S. (2019). Assessment of mudstone compaction in exploration wells in the Rovuma Basin, offshore Mozambique. Heliyon, 5(10), e02641.

Njiekak, G., & Schmitt, D. R. (2019). Effective Stress Coefficient for Seismic Velocities in Carbonate Rocks: Effects of Pore Characteristics and Fluid Types. Pure and Applied Geophysics, 176(4), 1467–1485.

Nur, A., & Byerlee, J. (1971). An exact effective stress law for elastic deformation of rock with fluids. Journal of Geophysical Research, 76(26), 6414–6419.

Oloruntobi, O., Adedigba, S., Khan, F., Chunduru, R., & Butt, S. (2018). Overpressure prediction using the hydro-rotary specific energy concept. Journal of Natural Gas Science and Engineering, 55, 243–253.

Oloruntobi, O., & Butt, S. (2019). Energy-based formation pressure prediction. Journal of Petroleum Science and Engineering, 173, 955–964.

Oloruntobi, O., Falugba, O., Ekanem-Attah, O., Awa, C., & Butt, S. (2020). The Niger Delta basin fracture pressure prediction. Environmental Earth Sciences. https://doi.org/10.1007/s12665-020-09081-5

Oloruntobi, O. S. (2019). The pore pressure, bulk density and lithology prediction (Doctoral dissertation, Memorial University of Newfoundland). Memorial University of Newfoundland.

Osarogiagbon, A.U., Khan, F., Venkatesan, R., & Gillard, P. (2021). Review and analysis of supervised machine learning algorithms for hazardous events in drilling operations. Process Safety and Environmental Protection, 147. https://doi.org/10.1016/j.psep.2020.09.038

Osarogiagbon, Augustine Uhunoma, Oloruntobi, O., Khan, F., Venkatesan, R., & Butt, S. (2020). Gamma ray log generation from drilling parameters using deep learning. Journal of Petroleum Science and Engineering, 195, 107906.

Sadiq, T., & Nashawi, I. S. (2000). Using neural networks for prediction of formation fracture gradient. SPE/CIM International Conference on Horizontal Well Technology.

Saleh, S., Williams, K. E., & Rizvi, A. (2013). Resistivity-Based Pore-Pressure Prediction-Pitfalls and Solutions. SPE Annual Technical Conference and Exhibition.

Saner, S., Al-Harthi, A., & Htay, M. T. (1996). Use of tortuosity for discriminating electro-facies to interpret the electrical parameters of carbonate reservoir rocks. Journal of Petroleum Science and Engineering, 16(4), 237–249.

Sarker, R., & Batzle, M. (2008). Effective stress coefficient in shales and its applicability to Eaton's equation. The Leading Edge, 27(6), 798–804.

Sayers, C. M., Johnson, G. M., & Denyer, G. (2002). Predrill pore-pressure prediction using seismic data. Geophysics, 67(4), 1286–1292.

Sethi, D. K. (1979). Some considerations about the formation resistivity factor-porosity relations. SPWLA 20th Annual Logging Symposium.

Shen, Y., Luan, G., Zhang, H., Liu, Q., Zhang, J., & Ge, H. (2017). Novel method for calculating the effective stress coefficient in a tight sandstone reservoir. KSCE Journal of Civil Engineering, 21(6), 2467–2475.

Swarbrick, R. E. (2001). Pore-pressure prediction: pitfalls in using porosity. Offshore Technology Conference.

Tingay, M. R. P., Hillis, R. R., Swarbrick, R. E., Morley, C. K., & Damit, A. R. (2009). Origin of overpressure and pore-pressure prediction in the Baram province, Brunei. Aapg Bulletin, 93(1), 51–74.

Ugwu, S. A., & Nwankwo, C. N. (2014). Integrated approach to geopressure detection in the X-field, Onshore Niger Delta. Journal of Petroleum Exploration and Production Technology, 4(3), 215–231.

Worthington, P. F. (1993). The uses and abuses of the Archie equations, 1: The formation factor-porosity relationship. Journal of Applied Geophysics, 30(3), 215–228.

Xu, X., Hofmann, R., Batzle, M., & Tshering, T. (2006). Influence of pore pressure on velocity in low-porosity sandstone: Implications for time-lapse feasibility and pore-pressure study. Geophysical Prospecting, 54(5), 565–573.

Yu, G., & Aguilera, R. (2011). Use of Pickett plots for evaluation of shale gas formations. SPE Annual Technical Conference and Exhibition.

Zhang, J. (2011). Pore pressure prediction from well logs: Methods, modifications, and new approaches. Earth-Science Reviews, 108(1–2), 50–63.

Zhang, J. (2013). Effective stress, porosity, velocity and abnormal pore pressure prediction accounting for compaction disequilibrium and unloading. Marine and Petroleum Geology, 45, 2–11.

Zhang, J. J. (2019). Applied Petroleum Geomechanics. Gulf Professional Publishing.

Zhao, J., Li, J., & Xu, Z. (2018). Advances in the origin of overpressures in sedimentary basins. Petroleum Research, 3(1), 1–24.

Chapter 5

5. Gamma ray log generation from drilling parameters using deep learning

Preamble

This chapter addresses an objective of this dissertation as outlined in Section 1.3 which is exploiting how drilling parameters can be utilized by deep learning algorithm for real time lithology identification. The use of drilling parameters for gamma ray log generation offers an economic means of improving the accuracy of shale lithology identification. Section 1.1. and Section1.6. show that shale lithology identification is a necessary step for pore pressure prediction which is vital for drilling safety.

I (Augustine Uhunoma Osarogiagbon) have contributed to Conceptualization, Methodology, Formal Analysis, Software, Investigation, Writing - Original Draft, and Writing - Review & Editing of this work, while Dr. Olalere Oloruntobi contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing; Dr. Faisal Khan contributed to Conceptualization, Methodology, Formal Analysis, Writing - Review & Editing and Supervision; Dr. Ramachandran Venkatesan contributed to Methodology, Formal Analysis, Writing - Review & Editing and Supervision; and Dr. Stephen Butt contributed to Methodology, Writing - review & editing and Project administration. A version of this chapter has been published in the Journal of Petroleum Science and Engineering, Volume 195, December 2020, 107906, https://doi.org/10.1016/j.petrol.2020.107906.

Abstract

Lithology identification plays a vital role in defining the petroleum reservoir. Although well logging represents the traditional means of obtaining petrophysical data for lithology identification, there could be cases where logging while drilling instruments may fail during drilling. This chapter presents an approach that utilizes drilling parameters obtained from mud logging and measurement while drilling (MWD) for real-time prediction of gamma ray log which is used as a lithology identifier. In this chapter, several machine learning methodologies, such as simple recurrent neural network (RNN), long short-term memory recurrent neural network (LSTM-RNN), temporal convolution network (TCN), gated recurrent unit (GRU) network, nonlinear autoregressive network with exogenous inputs (NARX) and simple artificial neural network (ANN) were tested for their ability to capture the relationship between hydro-mechanical specific energy (computed from drilling parameters) and gamma ray log. A recently drilled exploration gas well in the tertiary deltaic system of the Niger delta basin will be used as a case study. Base on the field data, the results show that the TCN and simple RNN performs best. The receptive field of the TCN plays a significant role in its performance, and therefore, the LSTM-RNN can be made to perform comparable to that of TCN/simple RNN if the LSTM-RNN is manually made to work with an optimal window of input data points for each output data point. The size and nature of the data (volume, velocity, variety, and veracity) is likely a significant factor in the performance of the machine learning methodologies. Thus, it is recommended to focus on obtaining the best receptive field of the data during the machine learning development phase.

Keywords: Deep learning; Logging while drilling; Measurement while drilling; Lithology; Drilling parameters; Gamma ray log; NARX; RNN; TCN; LSTM; GRU; ANN

5.1 Introduction

Lithology identification provides a means of defining the size, boundary, and property (e.g. permeability) of petroleum reservoirs. The gamma ray log provides a means of identifying changes in lithology in the siliciclastic environment. Equations which relate the volume of shale and gamma ray index have been developed (Assaad, 2008; Clavier et al., 1971; Larionov, 1969; Olayiwola & Bamford, 2019; O. Oloruntobi & Butt, 2019; Oloruntobi, 2019; Stieber, 1970; Yusuf et al., 2019). While the use of logging while drilling (LWD) offers a useful means of identifying subsurface lithology in real-time, there are several downhole drilling conditions where the application of LWD may prove insufficient. For example, using the conventional LWD sensors (e.g. gamma ray log sensor) that are often placed behind a mud motor or rotary steerable system to determine the coring point of a very thin reservoir. Under this scenario, the entire thickness of the reservoir may be unknowingly drilled before the LWD sensors are able to pick the reservoir tops. While the near bit LWD sensors allow lithology identification a few distances behind the bit, it is costly and often not run. Therefore, the possibility of obtaining drilling parameters at bit point offers significant benefits for lithology prediction (Oloruntobi & Butt, 2020a).

In poor borehole conditions (excessive breakouts and washouts), the LWD data may produce erroneous readings. Therefore, the objective of this work is to provide a means of predicting gamma ray log in real-time using a parameter obtained from the drilling data in siliciclastic environments. This will provide a means of identifying subsurface lithology at relatively no extra cost since drilling parameters are readily available in real-time. The new method will allow engineers to predict subsurface lithology when LWD data is missing/incomplete due to LWD instrument failure, borehole enlargement, or economic reason (Salehi et al., 2017; D. Zhang et al., 2018).

Generating values of logging parameters with machine learning represents an interesting area of petroleum Geoscience/Engineering. For example, in the article by Zhang et al. (2018), logging parameters (high-resolution acoustic log, borehole compensated sonic log and density) were generated from other logging parameters (gamma ray, caliper, spontaneous potential and amplitude difference of micro potential and micro gradient) using LSTM-RNN. In the article by Gholami & Ansari (2017), porosity was generated from seismic attributes using a committee of optimized ANN, optimized support vector regression, and optimized fuzzy logic. There have also been research interests in generating logging parameters from drilling parameters. In the article by Kanfar et al. (2020), logging parameters were generated from drilling parameters using an Inception-based convolutional neural network combined with TCN. i.e., porosity and density were generated from depth, rate of penetration, weight on bit, flow rate, and mechanical specific energy, while compressional sonic was generated from the rate of penetration, weight on bit, torque and mechanical specific energy.

The work described in this chapter will test the performance of several conventional machine learning algorithms (especially those with sequence modeling capabilities) on their ability to generate gamma ray log using drilling parameters. This is industrially significant as it offers an additional means of predicting gamma ray log data at the bit and reconstructing missing sections within the gamma ray log when sufficient gamma ray log and drilling parameter data is available for a well. The chapter is structured as follows. Section 5.2 describes the drilling parameters used in this work for gamma ray log generation. Section 5.3 introduces the machine learning algorithms used. Section 5.4 presents the methodology. Section 5.5 describes the results obtained. Section 5.6 provides a conclusion.

5.2 Drilling parameter data for lithology identification

Several attempts have been made to predict lithology using drilling parameters such as the rate of penetration (ROP) and d-exponent. However, these parameters do not account for factors such as bit type, bit wear, torque, and bit hydraulic energy. To overcome these shortcomings (except for bit hydraulic energy), the concept of mechanical specific energy (MSE) was proposed by Teale (1965). It is given by:

$$MSE = \frac{WOB}{A_b} + \frac{120\pi NT}{A_b ROP}. \tag{5.1}$$

The MSE, as shown in Equation (5.1), is the sum of the axial and rotary energy required to remove a unit of rock. WOB refers to the downhole weight on bit (lb), A_b refers to bit area (in^2), N refers to rotary speed (rpm), ROP refers to the rate of penetration (ft/hr), and T refers to torque (lb-ft). In order to account for the effect of weakening the rock ahead of the bit, the hydraulic energy in the bit can be added to the MSE. This results in the hydro-mechanical specific energy (HMSE), as shown in Equation (5.2) (Oloruntobi et al., 2018; Oloruntobi & Butt, 2019a).

$$HMSE = \frac{WOB}{A_b} + \frac{120\pi NT}{A_b ROP} + \frac{1154\eta \Delta P_b Q}{A_b ROP}. \tag{5.2}$$

$$Bit\ hydraulic\ energy = \frac{1154\eta \Delta P_b Q}{A_b ROP}. \tag{5.3}$$

Here, η refers to hydraulic energy reduction factor, ΔP_b refers to bit pressure drop at the nozzle (psi), Q refers to flow rate (gpm).

For polycrystalline diamond compact (PDC) bits, η can be computed using Equation (5.4).

$$\eta_{PDC\ bit} = 1 - \left(\frac{JSA}{TFA}\right)^{-0.122}. \tag{5.4}$$

Here, JSA refers to junk slot area (in^2) and TFA refers to flow area (in^2).

For roller cone bits (RCB), η can be computed using Equation (5.5).

$$\eta_{RCB} = 1 - \left(\frac{0.15\ Bit\ area}{TFA}\right)^{-0.122}. \tag{5.5}$$

The pressure drop at the nozzle can be computed using Equation (5.6).

$$\Delta P_b = \frac{MW\ Q^2}{10858\ TFA^2}. \tag{5.6}$$

Here, MW refers to mud weight (PPG) and Q refers to flow rate (gpm),

In addition to lithology changes, several other factors, such as rock compaction and bit wear, can also influence HMSE. For example, rock compaction typically increases with depth, which leads to an increase in HMSE. Also, bit wear, which usually increases with depth, can lead to an increase in HMSE. To account for the rock compaction effect, a porosity compaction model proposed by Athy (1930) is applied to Equation (5.2) to yield Equation (5.7).

$$HMSE_{dn} = \left(\frac{WOB}{A_b} + \frac{120\pi NT}{A_b ROP} + \frac{1154\eta \Delta P_b Q}{A_b ROP}\right)\emptyset_0 e^{-KZ}. \tag{5.7}$$

Here, $HMSE_{dn}$ represents the depth-effect normalized HMSE. $\emptyset_0$ refers to the surface or mudline porosity, Z refers to true vertical depth (ft), and K refers to the compaction coefficient (1/ft).The values of $\emptyset_0$ and K can be obtained from offset wells. Oloruntobi & Butt (2020a) showed an observable correlation between gamma ray log and $HMSE_{dn}$. Therefore, the goal of the work described in this chapter is to capture the relationship between $HMSE_{dn}$ and gamma ray log using machine learning. One benefit of using $HMSE_{dn}$ is that it provides an expert means of capturing the inter-relationship between the various drilling parameters (torque, weight on bit, bit size etc.,) before applying machine learning. This is beneficial when training data is limited in size.

5.3 Machine learning algorithms

The motivation for considering several machine learning algorithms is because of the possibility of having different relative performance of machine learning algorithms for different task. For example, LSTM-RNN, which has achieved success for a complex task such as language modeling (Sundermeyer et al., 2015), performed poorer than Autoregressive Integrated Moving Average (ARIMA) for time series forecasting (Han, 2018). In an article by Chen, (2020); LSTM-RNN, ARIMA and XGBOOST (T. Chen & Guestrin, 2016; J. Tang et al., 2020; K. Zhou et al., 2020) were used for time series forecasting. The results in the article showed that ARIMA performed best in terms of root mean square error whereas XGBOOST and LSTM-RNN had better results than ARIMA in terms of mean absolute percentage error.

Tang et al., (2018) showed that the relative performance of machine learning methodologies can vary as a function of data size. Therefore, it is not out of place to observe the performance of different machine learning algorithms. In machine learning, the concept of big data is not only defined in terms of volume, but also with respect to velocity, variety, and veracity (L'heureux et al., 2017).

5.3.1 Artificial neural network (ANN)

A simple or shallow ANN is made up of an input layer, one or few simple hidden layers and an output layer (Alom et al., 2019). An example is presented in Fig. 5.1.

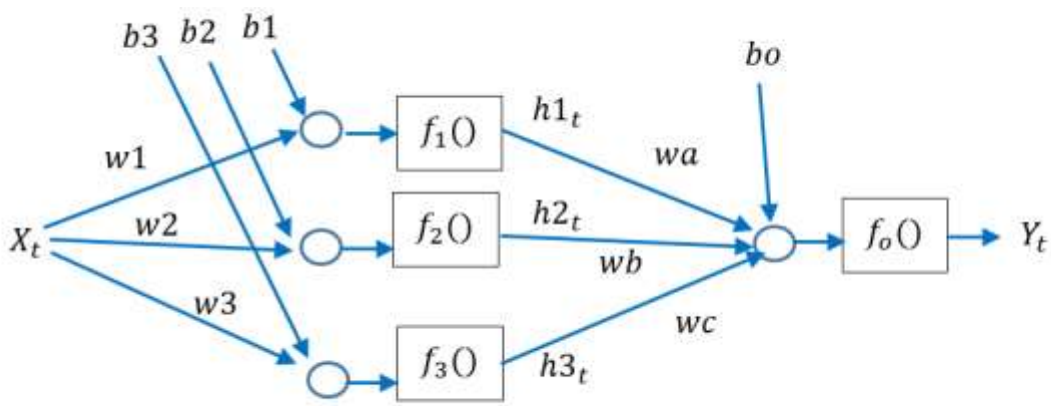

Fig. 5.1 An ANN with one hidden layer, one input (X_t) and one output (Y_t)

In Fig. 5.1, the input layer comprises of the input. The hidden layer comprises of the following weights w1, w2 and w3, biases b1, b2 and b3, node activation functions $f_1()$, $f_2()$ and $f_3()$ and computed values $h1_t$, $h2_t$ and $h3_t$. The output layer comprises weights wa, wb and wc, node activation function $f_o()$ and computed value Y_t. The node represented by the circular shape ○ in Fig. 5.1 is a summation point. Equations 5.8 to 5.11 is required to compute the output or target value Y_t as a function of input value X_t. Note that X_t means parameter X varies with a sequence indicating parameter, likewise $Y_t, h1_t, h2_t$ and $h3_t$. The sequence indicating parameter could be time or space. For example, if Y_t represents output at time t, then Y_{t-1} represents output at time $t - 1$.

$$h1_t = f_1(w1 \times X_t + b1), \tag{5.8}$$

$$h2_t = f_2(w2 \times X_t + b2), \tag{5.9}$$

$$h3_t = f_3(w3 \times X_t + b3), \tag{5.10}$$

$$Y_t = f_o(wa \times h1_t + wb \times h2_t + wc \times h3_t + bo). \tag{5.11}$$

The hidden layer activation functions e.g. $f_1()$, $f_2()$ and $f_3()$ of Fig. 5.1 are non-linear. Recommended non-linear activation functions for hidden layer include hyperbolic tangent (Hagan et al., 1996), SELU (Klambauer et al., 2017), RELU and ELU (Clevert et al., 2016). If the task is regression or fitting (in contrast to classification), a linear activation function can be used in the output layer, i.e. $f_o()$ of Fig. 1 (Hagan et al., 1996).

5.3.2 Recurrent neural network (RNN)

The main difference between a simple ANN and a simple RNN is a feedback connection in the hidden layer of a RNN. Fig. 5.2 shows this difference.

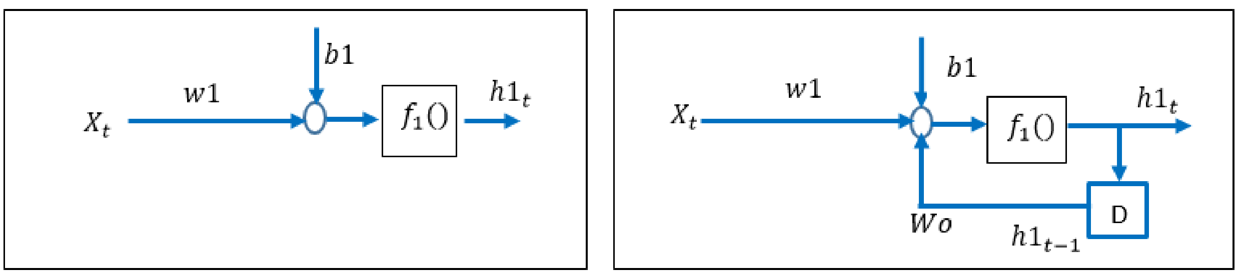

Fig. 5.2a Fig. 5.2b
Fig. 5.2 A hidden layer node connection in ANN and RNN a. ANN b. RNN

Fig. 5.2a represents one of the nodes in the hidden layer of Fig. 5.1 and Fig. 5.2b shows an addition of a unit delay D which passes the output of $f_1()$ back into the node in order to realize a simple RNN. This results in the modification of Equations 5.8, 5.9 and 5.10. For example, $h1_t$ for a simple RNN is given by Equation 5.12 (Alom et al., 2019; Elman, 1990).

$$h1_t = f_1(w1 \times X_t + wo \times h1_{t-1} + b1). \tag{5.12}$$

From Equation 5.12, it can be observed that $h1_t$ is a function of $h1_{t-1}$and X_t. Likewise, $h1_{t-1}$ is a function of $h1_{t-2}$ and X_{t-1}. This shows that $h1_t$ can capture the present and all past values of input X. From Equation 11, Y_t is a function of $h1_t$. Therefore for a simple RNN, the output at present time is a function of the present and all past values of input.

5.3.3 Nonlinear autoregressive network with exogenous inputs (NARX)

A NARX model predicts the value of an output variable as a function of past values of output as well as past and present values of input (Hagan et al., 1996). The NARX is very similar to a simple RNN. The main differences are: (i) in the feedback connection for NARX, Y_{t-1} is fed back to the hidden layer nodes instead of feeding back $h1_{t-1}$, $h1_{t-2}$ or $h1_{t-3}$; (ii) delay lines are added to take in past input values. A hidden layer node connection for a NARX implementation that uses the last two previous input values in addition to the present input value and immediate previous output value is shown in Fig 5.3.

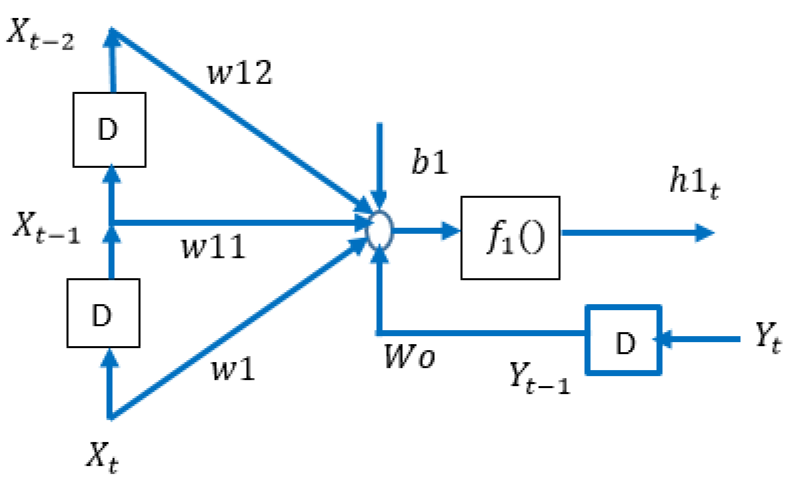

Fig. 5.3 A hidden layer node for NARX

Equation 5.12 is modified to give Equation 5.13 in order to compute $h1_t$ for the NARX implementation of Fig. 5.3.

$$h1_t = f_1(w1 \times X_t + w11 \times X_{t-1} + w12 \times X_{t-2} + wo \times Y_{t-1} + b1). \qquad (5.13)$$

5.3.4 Long short-term memory recurrent neural network (LSTM-RNN)

The problem of vanishing gradient limits the capability of simple RNN to capture sequential relationship in a very long series of data. The article by Hochreiter & Schmidhuber (1997), gives a detailed explanation of how the LSTM implementation overcomes this challenge. Fig. 4 shows an LSTM unit which replaces the hidden layer of a simple RNN shown in Fig. 2b in order to obtain the LSTM-RNN.

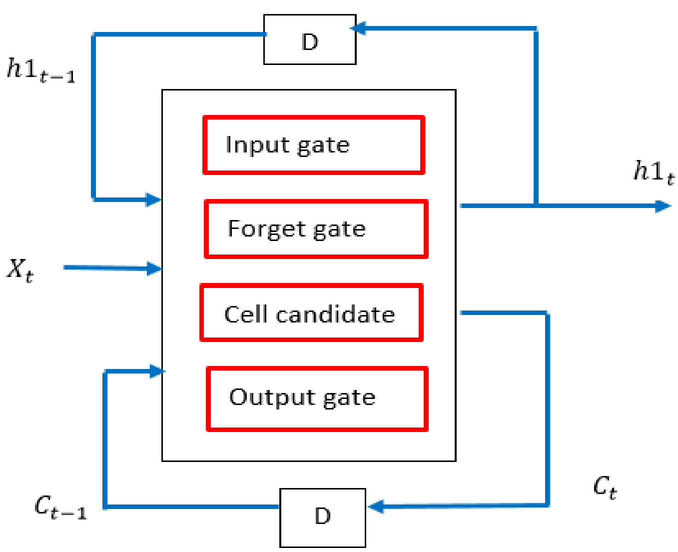

Fig. 5.4 Hidden layer unit of LSTM-RNN

In the LSTM of Fig. 5.4, the following components aid learning during training. They are: input gate Gi_t, forget gate Gf_t, cell candidate Gc_t and output gate Go_t. Each of the four components operates as a function of the current input value X_t and the previous $h1$ value i.e. $h1_{t-1}$. Equations (5.14) to (5.17) summarise this (Alom et al., 2019; D. Zhang et al., 2018).

$$Gi_t = F_s\{(X_t \times wi) + (h1_{t-1} \times vi) + bi\}, \tag{5.14}$$

$$Gf_t = F_s\{(X_t \times wf) + (h1_{t-1} \times vf) + bf\}, \tag{5.15}$$

$$Gc_t = F_h\{(X_t \times wc) + (h1_{t-1} \times vc) + bc\}, \tag{5.16}$$

$$Go_t = F_s\{(X_t \times wo) + (h1_{t-1} \times vo) + bo\}. \tag{5.17}$$

wi, wf, wc, wo, vi, vf, vc and vo are weights associated with the respective components, while bi, bf, bc and bo are the corresponding bias values for their respective components.

It should be noted that F_s and F_h are sigmoid (logistic) and hyperbolic tangent activation functions, as shown by Equations (5.18) and (5.19).

$$F_s\{z\} = \frac{1}{1+ e^{-z}}, \tag{5.18}$$

$$F_h\{z\} = \tanh(z). \tag{5.19}$$

The LSTM has a memory cell value C which is updated during training. The memory cell value at t is a function of the input gate, forget gate, cell candidate at t and memory cell value at $t - 1$. This is shown by Equation (5.20).

$$C_t = (Gf_t \times C_{t-1}) + (Gc_t \times Gi_t). \tag{5.20}$$

For the LSTM unit, $h1_t$ is a function of output gate and memory cell value as shown by Equation (5.21).

$$h1_t = Go_t \times F_h\{C_t\}. \tag{5.21}$$

5.3.5 Gated recurrent unit (GRU) network

The GRU can be considered as a simplified derivative of LSTM with some variations. Despite the relative simplification of GRU, its performance is still comparable to that of LSTM-RNN (Alom et al., 2019). The GRU replaces the input gate, forget gate, output gate and cell candidate of LSTM with update gate Gu_t, reset gate Gr_t and hidden state candidate Gh_t. These are shown by Equations (5.22)-(5.24) (Alom et al., 2019; Chung et al., 2014; G.-B. Zhou et al., 2016).

$$Gu_t = F_s\{(X_t \times wu) + (h1_{t-1} \times vu) + bu\}, \tag{5.22}$$

$$Gr_t = F_s\{(X_t \times wr) + (h1_{t-1} \times vr) + br\}, \tag{5.23}$$

$$Gh_t = F_h\{(X_t \times wh) + ((Gr_t \times h1_{t-1}) \times vh) + bh\}. \tag{5.24}$$

$h1_t$ for the GRU unit is computed as shown by Equation (5.25).

$$h1_t = h1_{t-1} \times (1 - Gu_t) + Gh_t \times Gu_t. \tag{5.25}$$

5.3.6 Temporal convolution network (TCN)

The convolutional neural network (CNN) which uses convolutional layers in deep learning has achieved state of the art success in image processing (Kumar et al., 2017). While the CNN is traditionally designed for 2D input data (W. Liu et al., 2017), convolutional layers have also found success in deep learning architectures for 1 D sequential data. The temporal convolutional network (TCN) architecture (Bai et al., 2018) and WaveNet architecture (Oord et al., 2016) represent examples of such deep learning architectures for 1 D sequential data. Fig. 5.5 shows a convolution unit of a TCN/WaveNet required to learn sequential relationship in 1 D data.

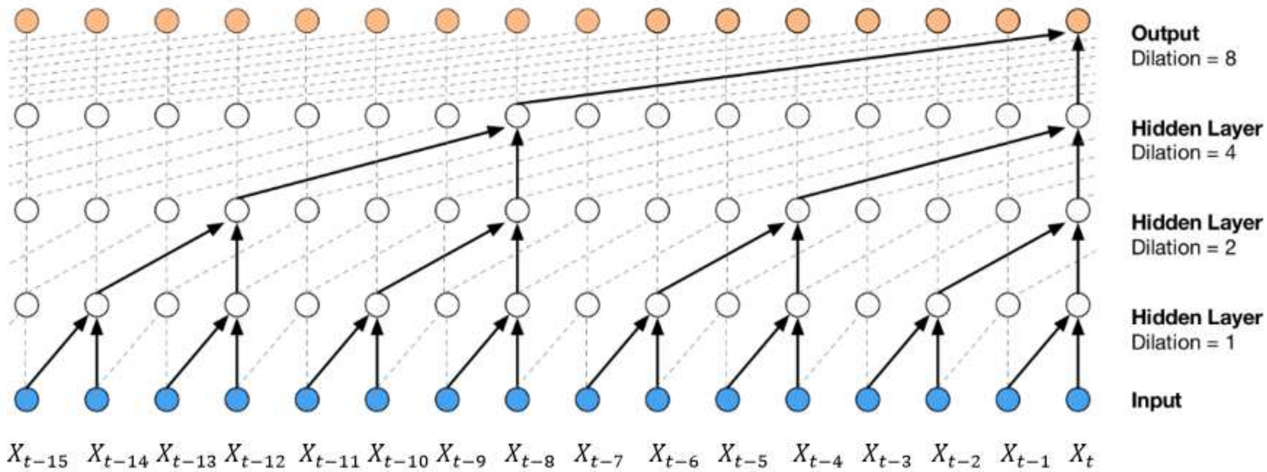

Fig. 5.5 A dilated causal convolution unit with dilation factors of d = 1, 2, 4, 8 and filter size of 2 (Oord et al., 2016)

The dilated causal convolution unit can be defined by its filter size (number of input to filter shown by the circle/ellipse shape in the hidden and output layers of figure 5.5), dilation factors (as observed by the fixed intervals between input entries to a filter) and number of hidden layers. Thus, the dilation factors, number of hidden layers and filter (also called kernel) size can be designed to capture a required receptive field. For example, the dilated causal convolutional unit of Fig. 5.5 is designed to learn the relationship between 16 consecutive sequential input data points (receptive field) and an output data point. Equation (5.27) shows how the receptive field (R_F) of a causal convolution unit can be evaluated for an architecture where the dilation factor d of the n^{th} convolutional layer is given by Equation (5.26) (Mathwork, 2020c).

$$d = 2^{n-1}, \tag{5.26}$$

$$R_F = ((F_s - 1) \times (2^L - 1)) + 1. \tag{5.27}$$

In Equation (5.27), F_s represents filter size, L represents the number of convolutional layers ($L =$ 4 for Fig 5.5).

It was reported that the TCN can achieve better performance than LSTM-RNN for complex machine learning task (Bai et al., 2018). One drawback of TCN in comparison to simple RNN, NARX, LSTM-RNN and GRU network is that the receptive field is required to be specified in TCN. This represents an additional parameter which can significantly affect performance if not properly selected, e.g. TCN can perform poorly if the kernel size and dilation are smaller than what is required to capture the effective memory required for a task (Bai et al., 2018). More detailed information on TCN can be found in Bai et al., (2018); Mathwork, (2020b); and Rémy, (2020).

5.4 Methodology

The flowchart of the methodology is presented in Fig. 5.6

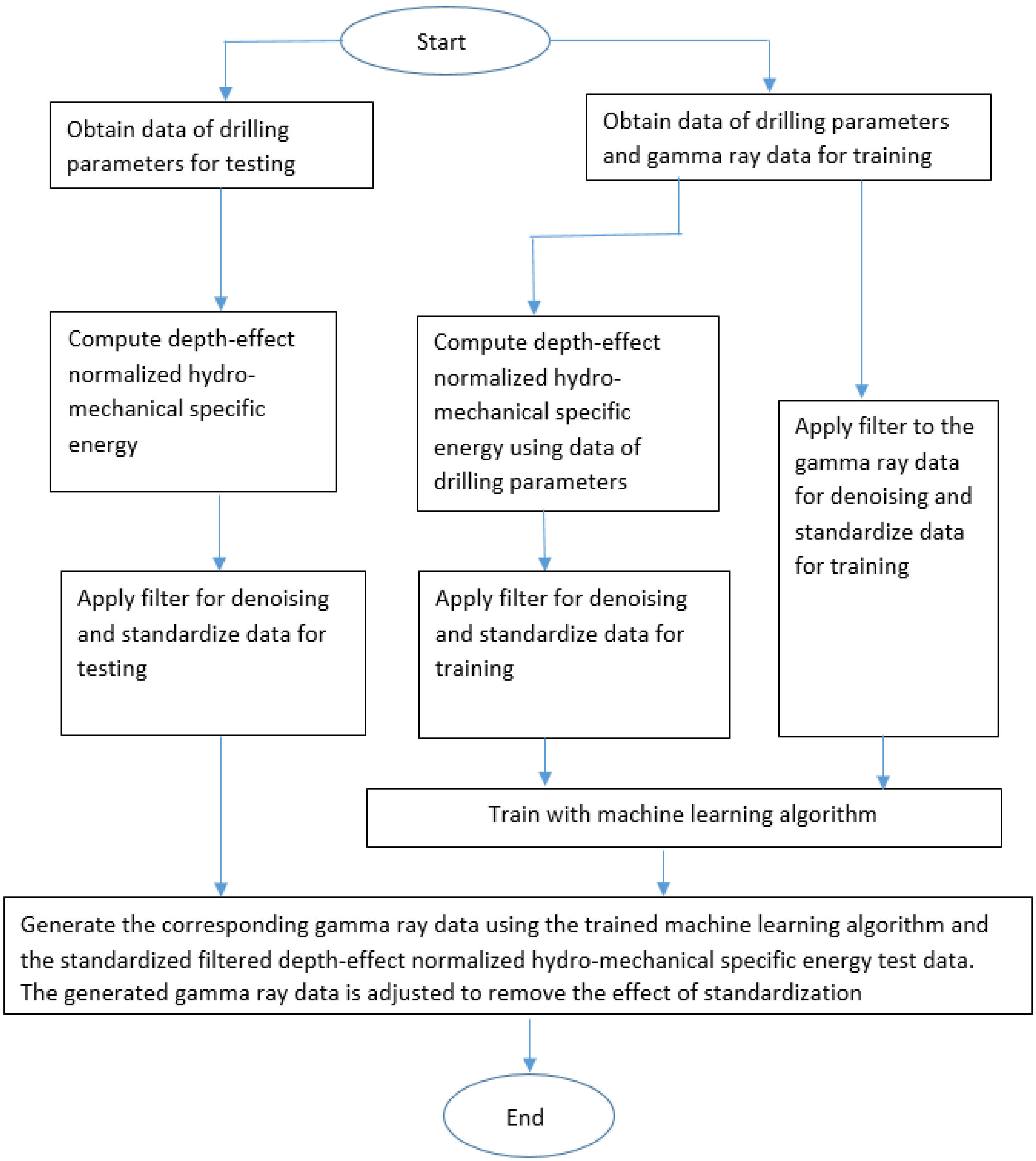

Fig. 5.6 Methodology for gamma ray log generation using drilling parameters and machine learning

5.4.1 Drilling parameters and gamma ray data for training and testing

The data of drilling parameters needed to compute $HMSE_{dn}$ is obtained along with the gamma ray log. It is expected that the $HMSE_{dn}$ and gamma ray log used for training would be sufficient to capture the relationship in the test data. Equation (5.7) is used in computing the $HMSE_{dn}$ of the well. Detailed explanation on the computation of $HMSE_{dn}$ for a well can be found in the article by Oloruntobi & Butt, (2020).

5.4.2 Filtering and standardization

Moving average (mean) with a length of 5 was applied as a filter to the gamma ray log in order to suppress noisy variations. The same filtering operation was done to the $HMSE_{dn}$ data. After filtering, the resulting gamma ray log and $HMSE_{dn}$ data are standardized for machine learning (Hagan et al., 1996). The data are standardized using the mean and standard deviation of the training data as shown by Equations (5.28), (5.29) and (5.30). Assume that the gamma ray log for training after filtering is $Gtrain$, the $HMSE_{dn}$ data for training after filtering is $Htrain$, the $HMSE_{dn}$ data for testing after filtering is $Htest$. μ_G and σ_G represent the mean and standard deviation of $Gtrain$, respectively, and μ_H and σ_H represent the mean and standard deviation of $Htrain$, respectively. $Gtrain_n$ and $Htrain_n$ represent the standardized input and target for training, while $Htest_n$ represents the standardized input for testing.

$$Gtrain_n = (Gtrain - \mu_G)/\sigma_G, \tag{5.28}$$

$$Htrain_n = (Htrain - \mu_H)/\sigma_H, \tag{5.29}$$

$$Htest_n = (Htest - \mu_H)/\sigma_H. \tag{5.30}$$

After training the neural network, $Htest_n$ will be fed into the neural network, and this will produce an output ($Goutput$). This output is re-adjusted by removing the effect of standardization using Equation (5.31).

$$Ggenerate = (\sigma_G \times Goutput) + \mu_G. \quad (5.31)$$

Here, $Ggenerate$ represents the generated gamma ray log that can be used for identifying changes in lithology.

5.4.3 Configuration and training of the machine learning algorithm

The summary of the configurations and training options for the various machine learning algorithms used is presented in Table 5.1. It should be noted that several numbers of hidden units for TCN, LSTM and GRU were tried as well as different dilations for TCN to observe performance. The results are reported in section 5.2.

Table 5.1 Summary of machine learning configuration and training

S/N	Parameter	TCN	LSTM and GRU	Simple ANN, simple RNN and NARX
1	Number of LSTM/GRU units (equivalent number for TCN)	(i) 4 (ii) 10 (iii) 20	For LSTM: (i) 3 (ii) 10 For GRU: (i) 4 (ii) 10	-
2	Number of hidden layer nodes for ANN, simple RNN and NARX	-	-	15 for simple ANN, 10 for simple RNN, 10 for NARX
3	Delay lines added to consider past input for NARX (Fig. 5.3)			9

4	Filter (kernel) size for TCN	2	-	-
5	Dilations for TCN. 5 different configurations were tried in order to test the performance of the following receptive fields (R_F): 8,16,32,64 and 128	(i) 1,2,4 (ii) 1,2,4,8 (iii) 1,2,4,8,16 (iv) 1,2,4,8,16,32 (v) 1,2,4,8,16,32,64	-	-
6	Number of hidden layers (number of stacks of residual blocks for TCN)	1	1	1
7	Padding for TCN	Causal	-	-
8	Use skip connections	False	-	-
9	Dropout rate	0	0	-
10	Hidden layer activation (activation used in the residual blocks for TCN, cell candidate for LSTM and hidden state candidate for GRU)	Hyperbolic tangent	Hyperbolic tangent	Hyperbolic tangent
11	Optimization algorithm	Adam (Kingma & Ba, 2014)	Adam	GNBR (Foresee & Hagan, 1997)
12	use batch normalization in the residual layers	False	-	-
13	Maximum training iteration	100	100	100

In order to minimize error due to random initialization of the weights, each training and testing is repeated thirty times and the mean result is reported. This is recommended when data set is small (Mathwork, 2020b). MATLAB and Keras (Rémy, 2020) were used for implementing the algorithms.

5.5 The results obtained by applying the methodology on a well data

To demonstrate the application of the proposed methodology, a gas well (Fig. 5.7) in the central swamp region of the Niger Delta basin is used as the case study. The Niger Delta is an extensional rift basin that consists of clastic sediments up to about 12 km thick at the central of the basin (Oloruntobi et al., 2019; Oloruntobi & Butt, 2020b; Oloruntobi et al., 2020).

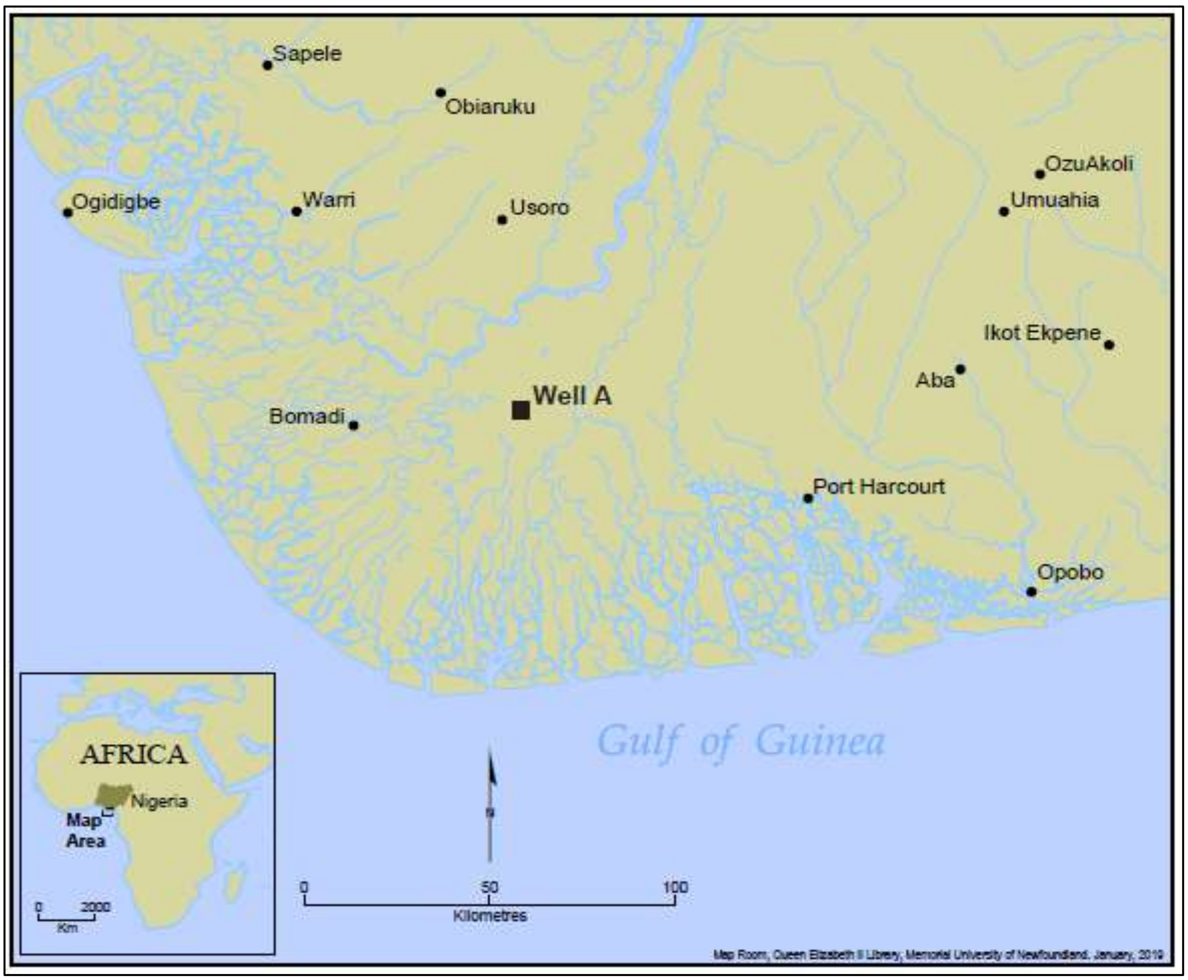

Fig. 5.7 The location map for well A used for training and testing

Fig. 5.8 shows the drilling parameter data (torque, rotary speed, flow rate, rate of penetration, weight on bit, mud weight and equivalent circulating density i.e. ECD), the LWD parameter (gamma ray), pore pressure (PP) and bottom hole pressure (BHP) of the well.

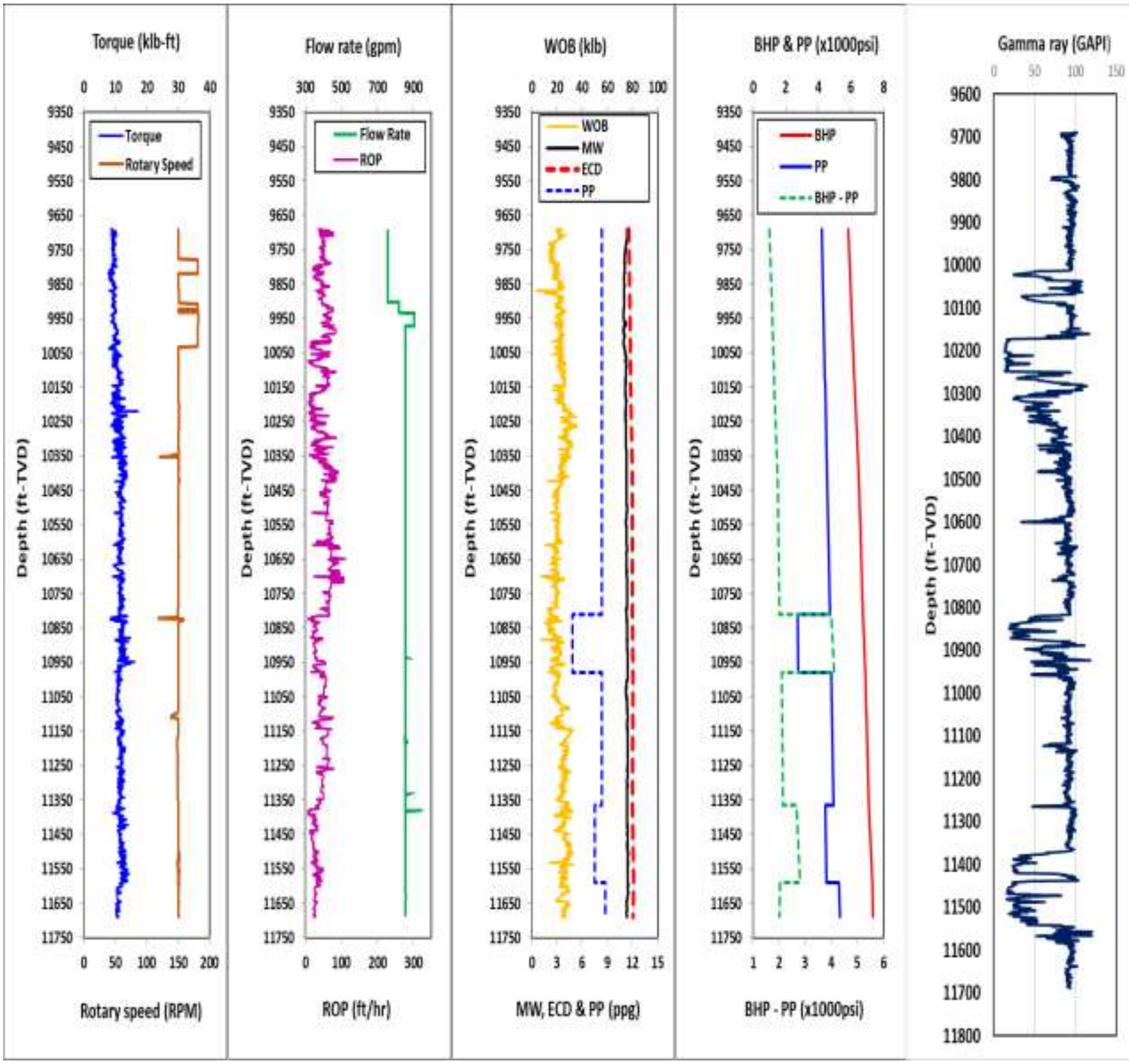

Fig. 5.8 Drilling and logging data of well A

In order to perform depth-effect normalization as shown by Equation (5.7), an offset well was used to obtain the appropriate values of $\emptyset_0$ and K. This yielded $\emptyset_0$ as 0.54 and K as 0.0001 (Oloruntobi & Butt, 2020a).

5.5.1 Denoising

For easy interpretation, outliers in the gamma ray log can be suppressed using low pass filter described in Section 5.4.2. The result of filtering is shown in Fig. 5.10 (compare with Fig. 5.9 before filtering).

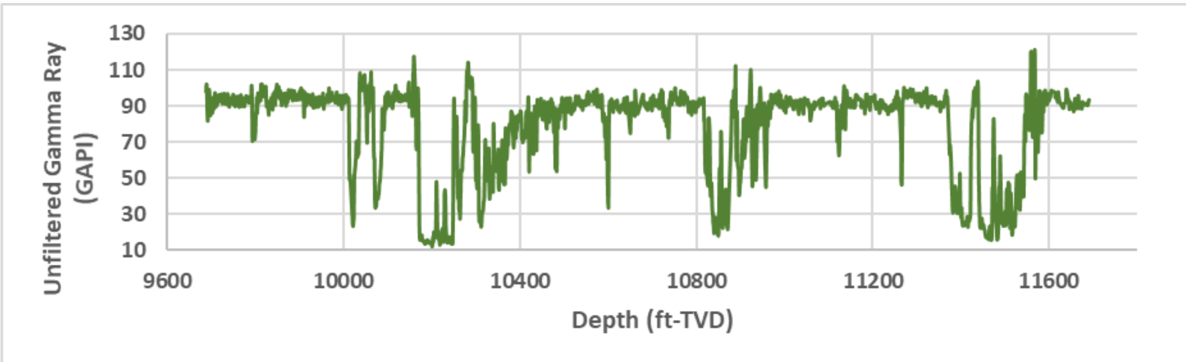

Fig. 5.9 Gamma ray log before filtering

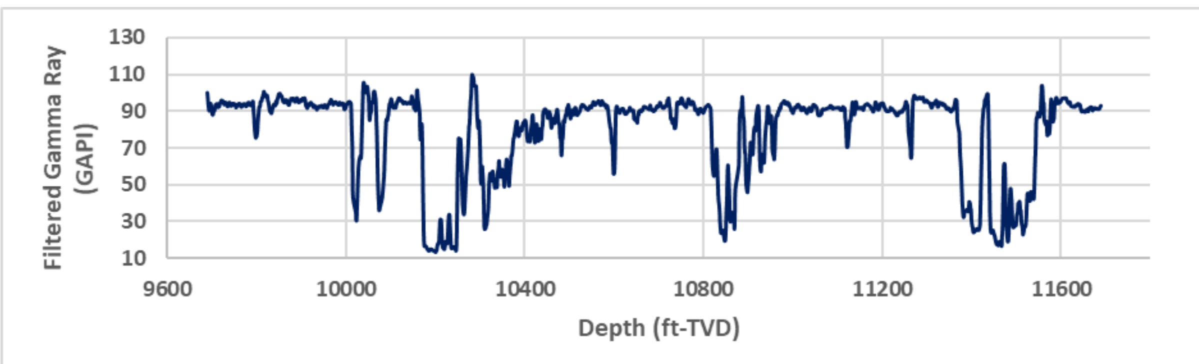

Fig. 5.10 Gamma ray log after filtering

The $HMSE_{dn}$ result obtained using Equation (5.7) is also filtered using the same filter described in Section 5.4.2. Fig. 5.11 compares the filtered gamma ray log and filtered $HMSE_{dn}$ data for well A.

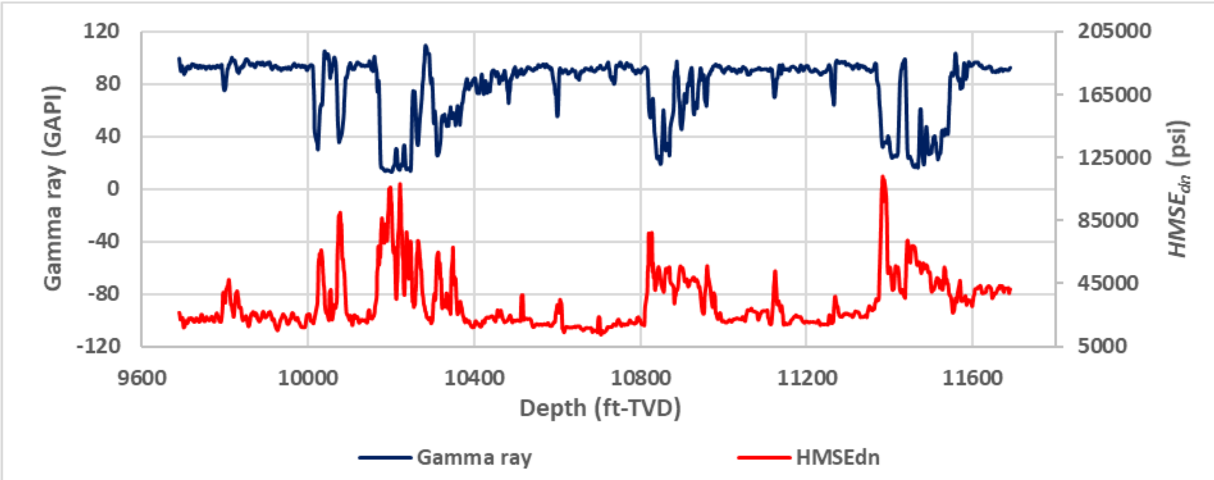

Fig. 5.11 Comparing depth normalized hydro-mechanical specific energy with gamma ray log

Visual inspection shows a good match between gamma ray and $HMSE_{dn}$ in Fig. 5.11. In order to get more insight into the relationship between the gamma ray and $HMSE_{dn}$ data of Fig 5.11, a plot of the $HMSE_{dn}$vs Gamma ray is shown in Fig. 5.12.

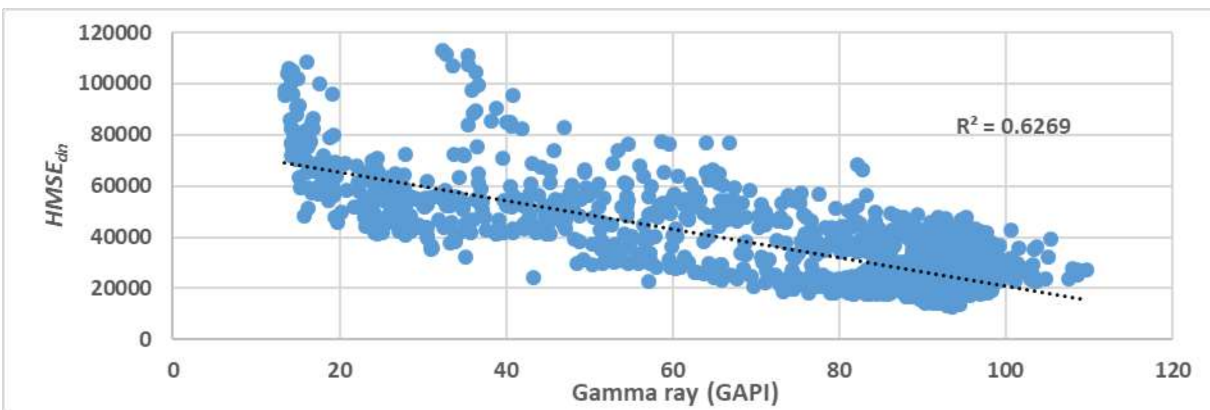

Fig. 5.12 Plot of $HMSE_{dn}$ versus gamma ray log to observe correlation

Due to the smilingly linear relationship in Fig. 5.12, results obtained using linear, quadratic, exponential, power and logarithm regression will also be compared with that obtained using the machine learning algorithms described in Section 5.3. Two categories of training and testing will be performed. The first category is for far end gamma ray log generation, the second category is for missing log interval generation.

5.5.2 Far end gamma ray log generation

This task is equivalent to real time prediction of logging parameter values at drilling depth where we already have values for drilling parameters. For this task, both $HMSE_{dn}$ and gamma ray log of Fig. 5.11 are split into training and testing parts as shown in Fig. 5.13. The initial 75% of the data (1428 data points) is used for training and the remaining 25% of the data (477 data points) is used for testing.

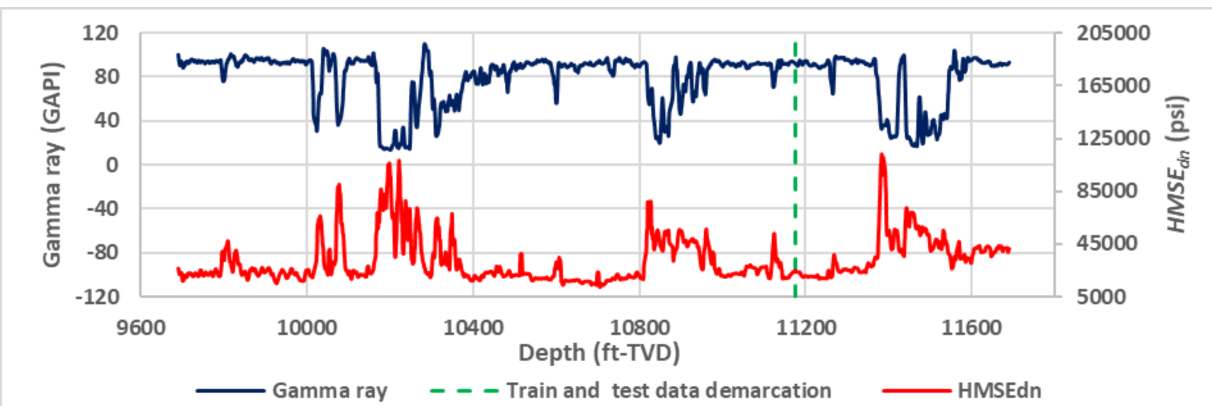

Fig. 5.13 Separating the gamma ray log and $HMSE_{dn}$ data into training and testing part

When testing for simple RNN, NARX, LSTM-RNN and GRU, all past available standardized filtered $HMSE_{dn}$ (training data) are first fed into the network before feeding in the standardized filtered test $HMSE_{dn}$ data. This is because we anticipate the first test output value to also depend on previous sequence step input (explained in the last paragraph of section 5.3.2).

The training and testing results are presented in Table 5.2 and the results are sorted based on mean absolute percentage error (MAPE) on test data. It should be noted that the results of Table 5.2 is not intended for comparing the superiority of machine learning algorithms, rather the results shows that size and complexity of data (as explained in Section 5.3.0) as well as configuration of machine learning algorithms can significantly affect performance. Although TCN and simple RNN performed better than LSTM-RNN (Table 5.2), it was observed that when the LSTM-RNN was forced to use a given window of input data (present plus previous 31 input data points) for each output data point as shown in Table 5.3, the LSTM-RNN performance improved significantly. Although we expect LSTM-RNN to appropriately learn the extent of long/short term sequential dependencies without defining a fixed window of input, the data used in this work may not be large enough for such learning. To observe the benefit/statistical stability of repeating each simulation 30 times, the ratio of standard deviation to mean output of each simulation (σ/μ) was also evaluated. Table 5.2 also includes performance of different regression models such as linear, exponential, logarithmic etc., (rows with S/N 23-27). The performance of these regression models are poor in comparison to the other machine learning methodologies. In Table 5.2, N/A means not applicable.

Table 5.2 Summary of result for far end prediction

S/N	Method used	Train			Test		
		σ/μ (%)	RMSE	MAPE (%)	σ/μ (%)	RMSE	MAPE (%)
1	TCN with 20 units and R_F =32	3.02	3.17	2.59	10.62	15.20	20.69
2	TCN with 4 units and R_F =32	4.89	8.76	10.55	8.98	14.03	20.72
3	TCN with 10 units and R_F =32	4.36	5.04	5.16	11.78	15.05	20.77
4	Simple RNN	4.80	9.01	10.59	18.25	12.96	20.94
5	TCN with 4 units and R_F=64	5.79	7.98	10.06	9.97	14.76	23.39
6	TCN with 10 units and R_F =16	4.54	7.05	8.06	9.12	16.11	24.34
7	NARX	4.78	7.33	7.93	21.81	13.99	24.42
8	TCN with 10 units and R_F = 128	4.04	2.70	2.73	15.17	15.79	24.80
9	TCN with 4 units and R_F =128	6.51	6.93	8.60	13.31	15.54	24.99
10	TCN with 10 units and R_F = 64	4.20	4.16	3.89	13.67	15.61	25.02
11	TCN with 4 units and R_F=16	4.42	9.72	12.23	6.93	15.52	25.14
12	TCN with 20 units and R_F=16	3.76	4.99	4.84	10.36	16.50	25.70
13	TCN with 20 units and R_F =128	2.33	0.85	0.88	12.80	16.09	26.13
14	TCN with 20 units and R_F =64	2.93	1.90	1.65	12.23	16.86	26.23
15	TCN with 4 units and R_F = 8	3.38	10.89	14.54	4.83	15.95	27.29
16	TCN with 10 units and R_F =8	3.74	9.33	11.69	5.58	16.86	27.56
17	LSTM-RNN with 3 units	0.06	12.53	17.01	0.08	15.88	27.99
18	TCN with 20 units and R_F=8	4.39	7.67	9.35	7.66	17.63	28.52
19	GRU with 4 units	1.31	12.38	16.94	1.72	16.81	30.68
20	LSTM-RNN with 10 units	0.04	12.38	16.42	0.06	17.37	32.00
21	Simple ANN	0.92	11.88	15.52	1.67	18.12	32.84
22	GRU with 10 units	0.50	12.42	16.69	0.82	17.67	32.88
23	Regression: Exponential	N/A	13.19	19.84	N/A	19.08	35.72
24	Regression: Logarithmic	N/A	13.76	20.98	N/A	19.64	36.28
25	Regression: Quadratic	N/A	12.78	18.41	N/A	19.17	36.58
26	Regression: Linear	N/A	12.78	18.45	N/A	19.20	36.70
27	Regression: Power	N/A	14.71	24.01	N/A	20.73	39.31

Table 5.3 Result obtained by constraining the LSTM-RNN to use a fixed window of input

S/N	Method used		Train			Test	
		σ/μ (%)	RMSE	MAPE (%)	σ/μ (%)	RMSE	MAPE (%)
1	LSTM-RNN with 10 units and R_F= 32	4.57	7.97	9.15	12.04	12.63	20.57
2	LSTM-RNN with 3 units and R_F= 32	2.77	11.05	13.44	5.49	13.54	22.17

Fig. 5.14 a, 5.14 b, and 5.14 c respectively show the gamma ray by TCN (20 units and RF =32), actual gamma ray (ground truth) and gamma ray by simple RNN for the test region of the far end log generation task.

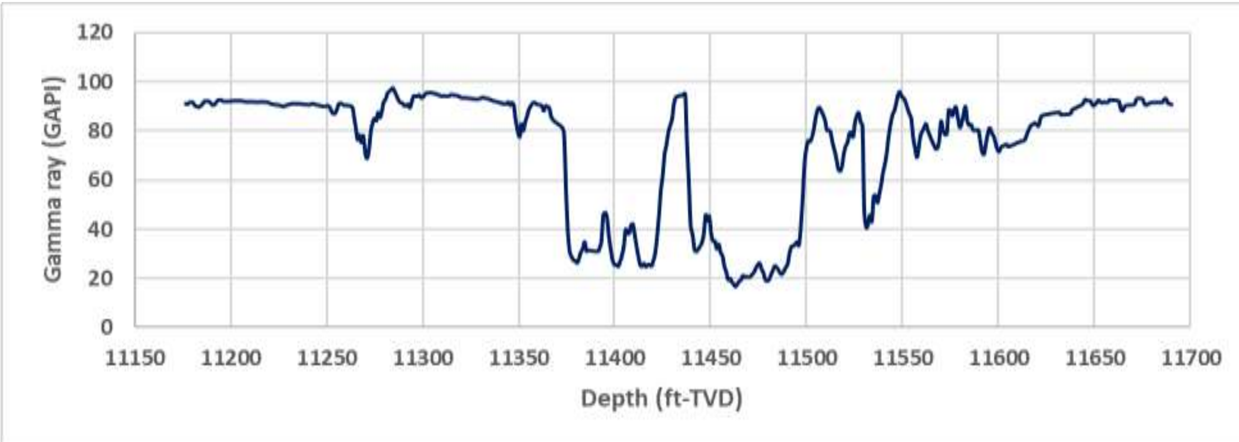

Fig. 5.14 a: Far end gamma ray log by TCN

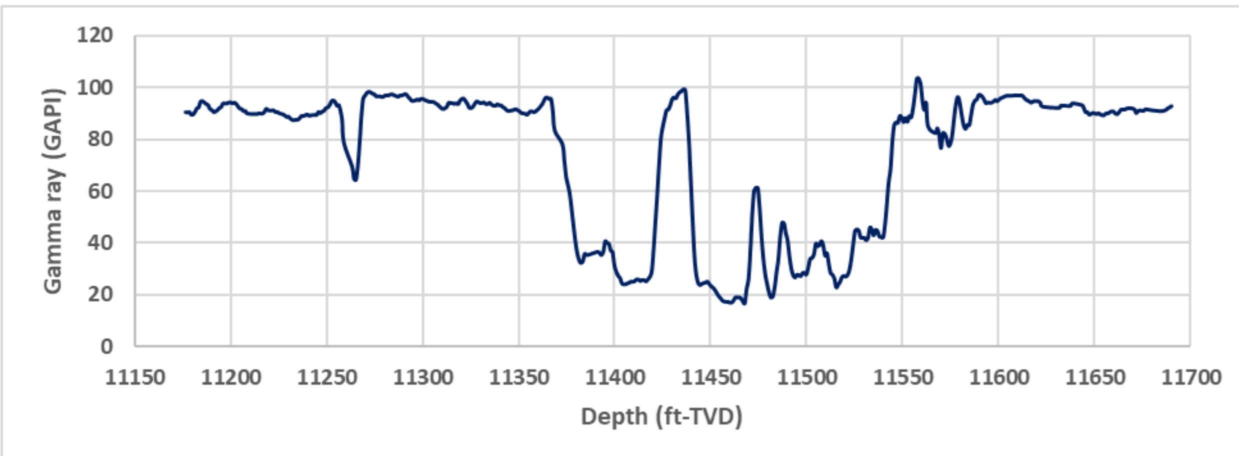

Fig. 5.14 b: Ground truth for far end gamma ray log

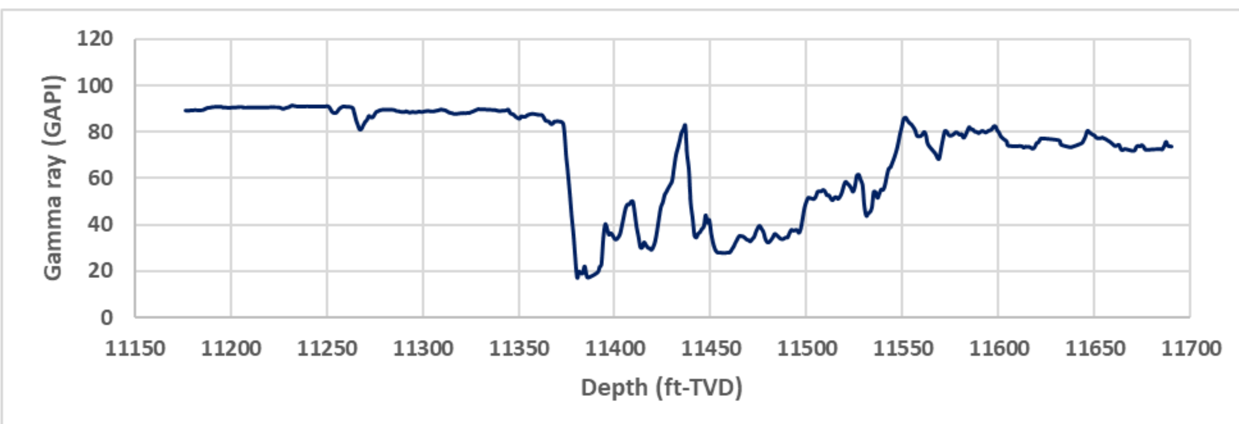

Fig. 5.14 c: Far end gamma ray log by simple RNN

Fig. 5.14 Far end gamma ray log result

Fig. 5.14 a, b and c shows that the gamma ray generated by simple RNN and TCN though not perfect, still has a good resemblance to the actual gamma ray data.

5.5.3 Window interval gamma ray log generation

This defines a situation where there are missing sections at the beginning or within gamma ray log. For this category of task, only the simple RNN and TCN will be considered due to the performance they achieved in Section 5.5.2. In this section, three different tasks which involve prediction of different intervals are described. Figures 5.15-5.17 respectively shows the sections of the gamma ray log that are used for training and testing.

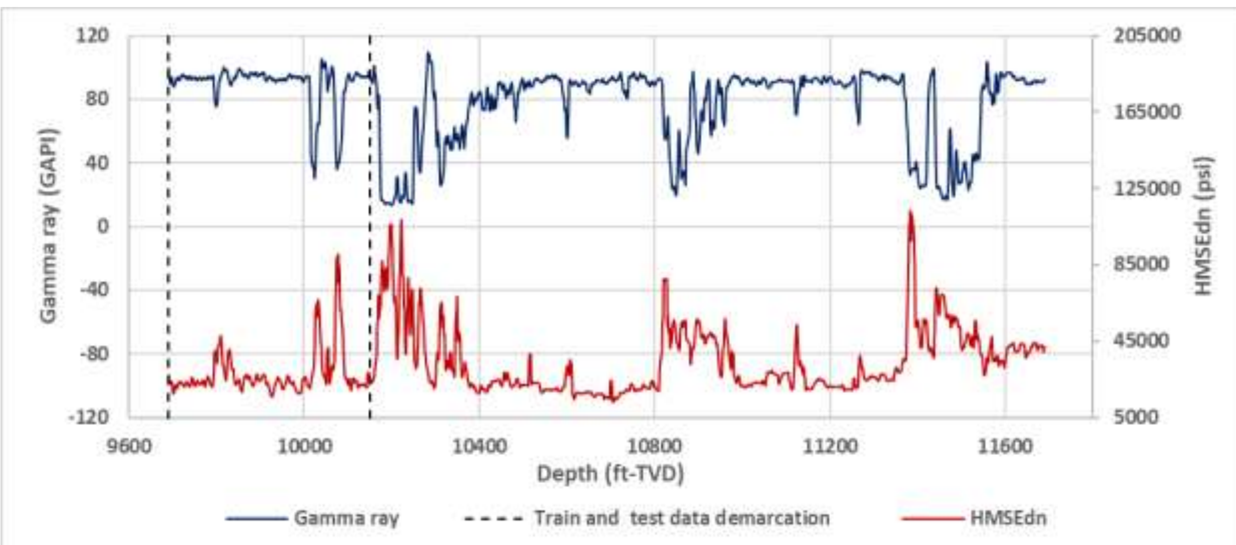

Fig. 5.15 Train and test data for window interval task 1

The missing log for window interval task 1 was selected to verify if simple RNN can generate the beginning section of a log, even though it was trained with the later part of the log. For interval task 1, the training section included 1439 data points and the test section included 466 data points. Window interval task 2 and 3 zooms into other sections of the gamma ray log with great changes as shown in Fig. 5.16 and Fig. 5.17 respectively. The missing section to be predicted for interval task 2 is about 47.71 ft and that for task 3 is 54.75 ft. For interval task 2, the training section included 1855 data points and the test section included 50 data points. Also, for interval task 3, the training section included 1855 data points and the test section included 50 data points.

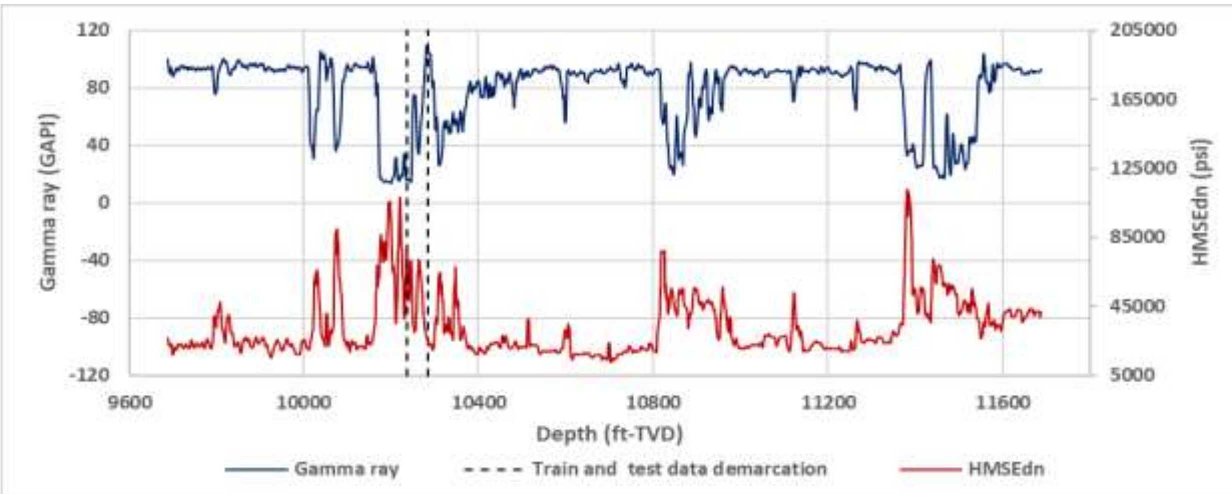

Fig. 5.16 Train and test data for window interval task 2

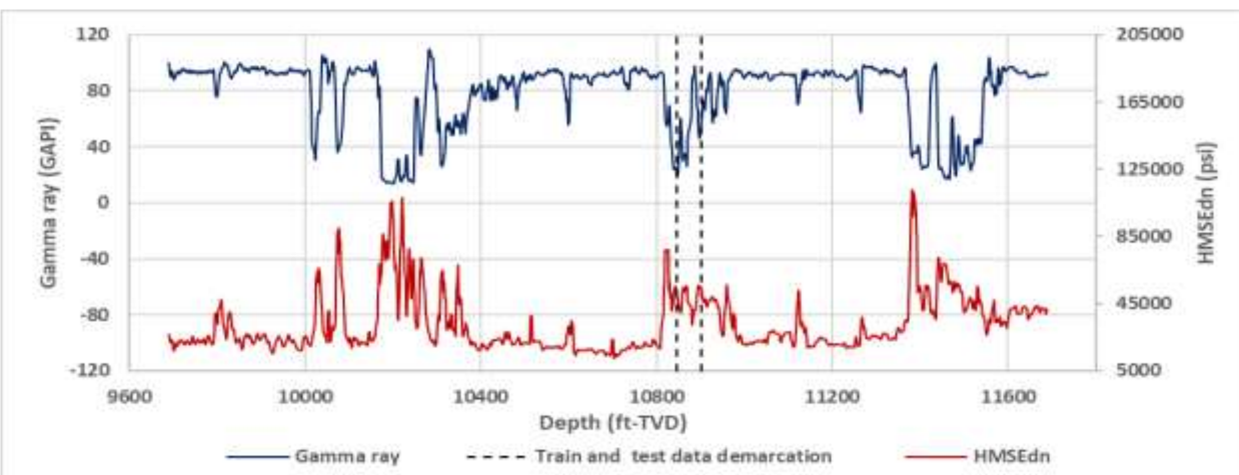

Fig. 5.17 Train and test data for window interval task 3

Table 4 summarizes the training and testing performance for window interval task 1, 2 and 3.

Table 5.4 Summary of window interval task

Task 1					Task 2				Task 3			
	Train		Test		Train		Test		Train		Test	
	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)
RNN	8.73	11.56	12.54	11.26	8.57	10.64	20.81	28.40	8.97	11.20	12.81	23.94
TCN	3.23	2.95	13.13	11.24	3.86	3.74	15.47	24.68	3.73	3.41	14.82	26.69

Table 5.4 shows an overall similarity in performance between simple RNN and TCN.

Fig. 5.18 a, 5.18 b and 5.18 c show the gamma ray by TCN, actual gamma ray (ground truth) and gamma ray by simple RNN for the test region of interval task 1, for visual comparison.

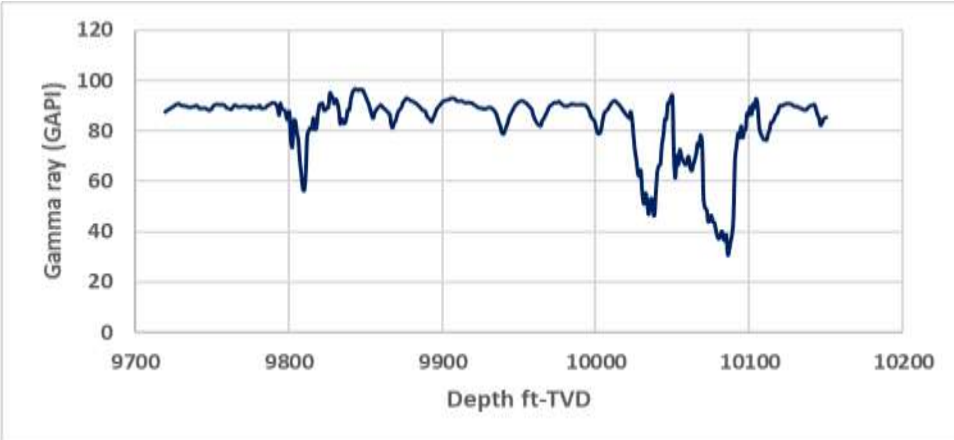

Fig. 5.18 a: Interval task 1 gamma ray log by TCN

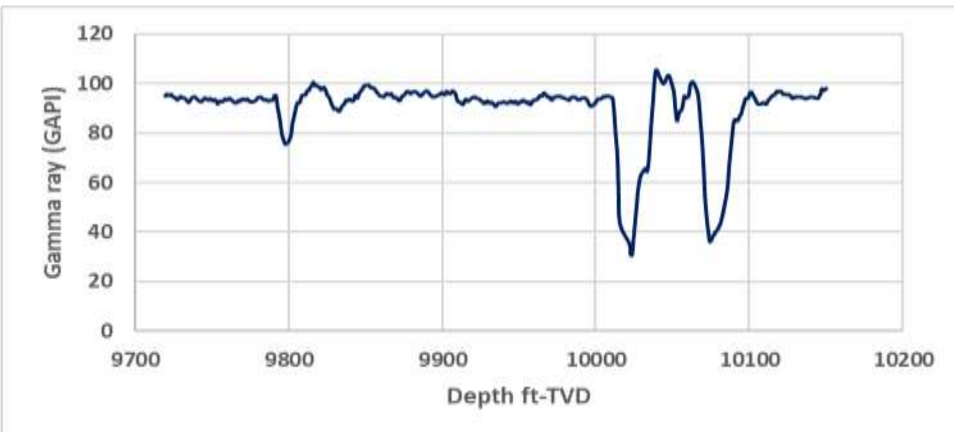

Fig. 5.18 b: Ground truth for interval task 1 gamma ray log

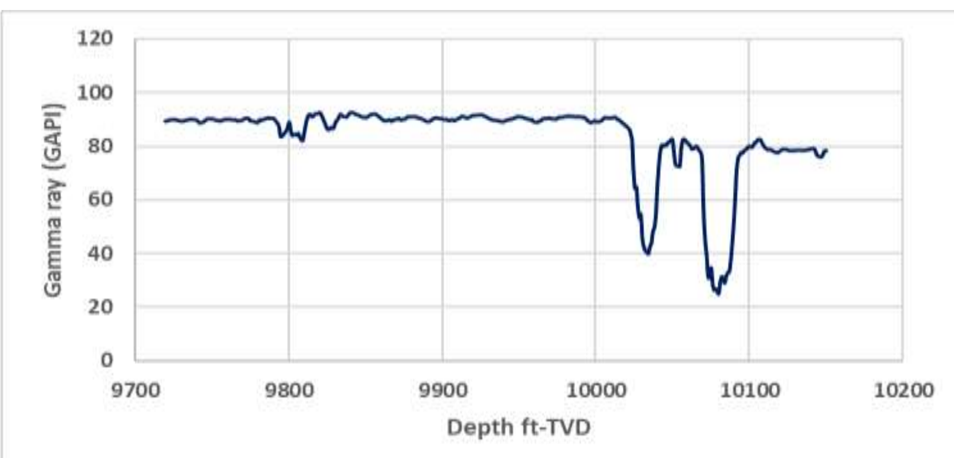

Fig. 5.18 c: Interval task 1 gamma ray log by simple RNN

Fig. 5.18 Interval task 1 gamma ray log result

Fig. 5.19 a, 5.19 b and 5.19 c show the gamma ray by TCN, actual gamma ray (ground truth) and gamma ray by simple RNN for the test region of interval task 2, for visual comparison.

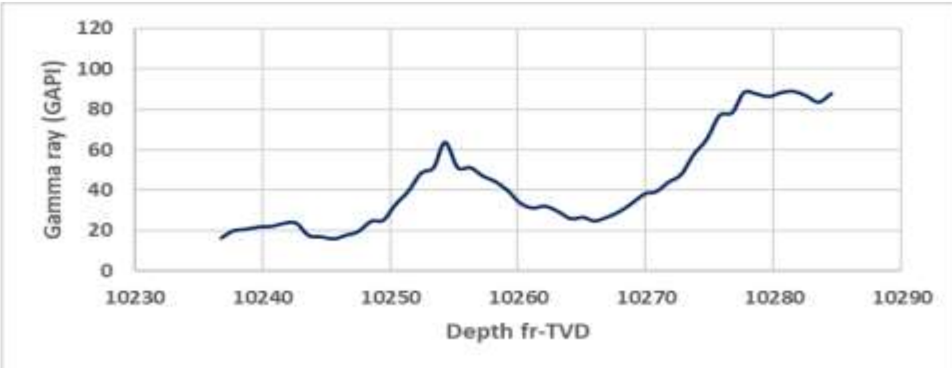

Fig. 5.19 a: Interval task 2 gamma ray log by TCN

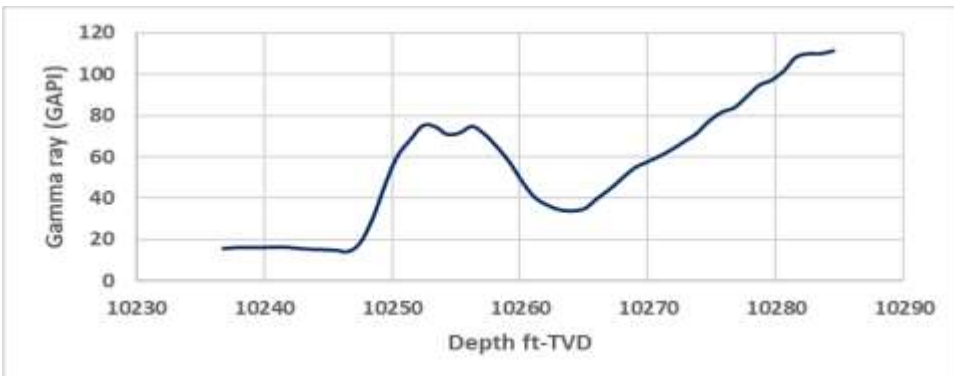

Fig. 5.19 b: Ground truth for interval task 2 gamma ray log

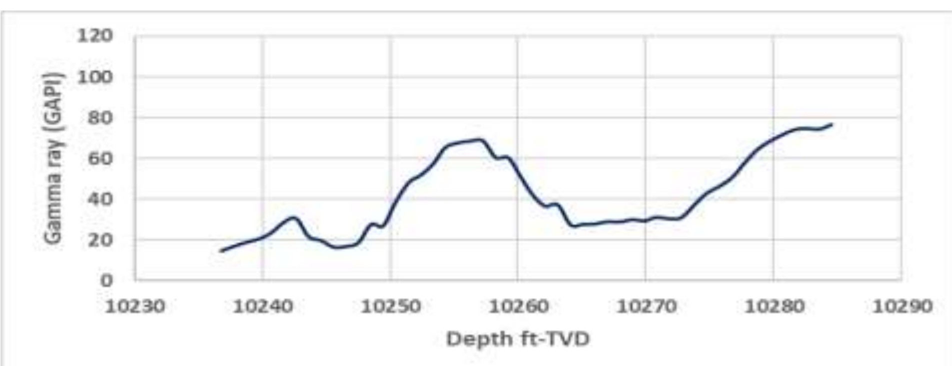

Fig. 5.19 c: Interval task 2 gamma ray log by simple RNN

Fig. 5.19 Interval task 2 gamma ray log result

Fig. 5.20 a, 5.20 b and 5.20 c show the gamma ray by TCN, actual gamma ray and gamma ray (ground truth) by simple RNN for the test region of interval task 3, for visual comparison.

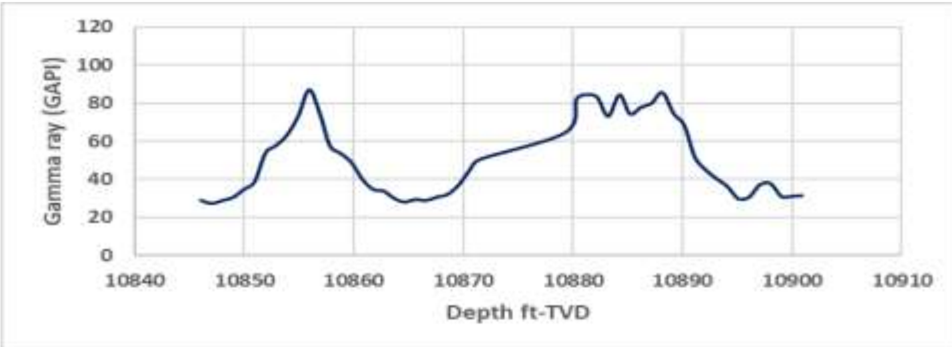

Fig. 5.20 a: Interval task 3 gamma ray log by TCN

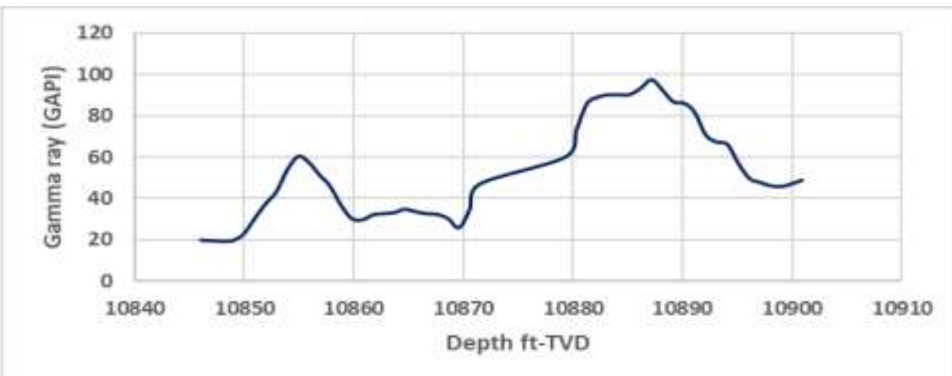

Fig. 5.20 b: Ground truth for interval task 3 gamma ray log

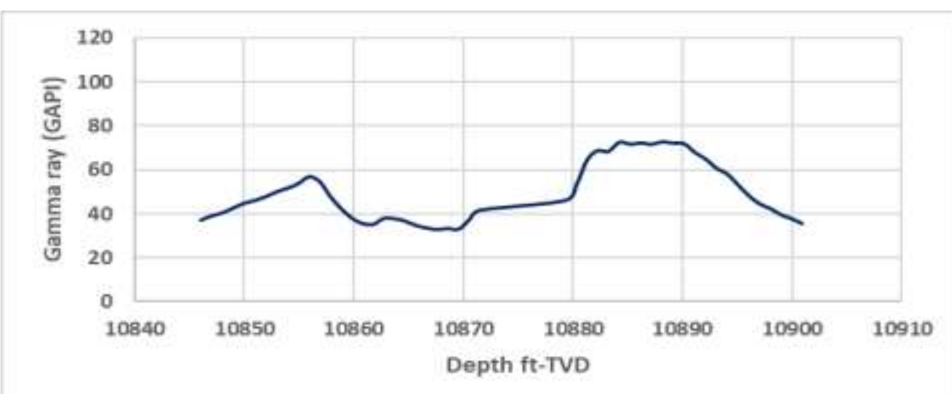

Fig. 5.20 c: Interval task 3 gamma ray log by simple RNN

Fig. 5.20 Interval task 3 gamma ray log result

The comparative overall performance of simple RNN with respect to TCN shows the ability of simple RNN to learn the extent of sequential dependency of input parameters for the field data considered.

5.6 Discussion

Although the work in this section was on the use of $HMSE_{dn}$ for gamma ray log generation, the separate parameters which are used in computing $HMSE_{dn}$ can be independently used to generate gama ray log using machine learning. A comparison was done using $HMSE_{dn}$ and the separate parameters (weight on bit, torque, rate of penetration, mud weight, true vertical depth, flow rate, and rotary speed). Other parameters such as bit area were excluded because they had constant values. A summary of the comparison using far end gamma ray log generation (Section 5.5.2) using the same machine learning configuration is presented in Table 5.5.

Table 5.5 Comparing performance between $HMSE_{dn}$ and the separate use of drilling parameters for gamma ray log generation

S/N	Method used	Train		Test	
		RMSE	MAPE (%)	RMSE	MAPE (%)
1	Simple ANN with $HMSE_{dn}$	11.88	15.52	18.12	32.84
2	Simple ANN with separated parameters	4.68	5.35	19.22	35.77
3	Simple RNN with $HMSE_{dn}$	9.01	10.59	12.96	20.94
4	Simple RNN with separated parameters	2.77	3.08	20.00	36.51

Table 5.5 results show that the use of $HMSE_{dn}$ performed better. This result does not aim to rule out any possibility of obtaining better result with the use of separated parameter, but rather it shows that significant benefit can be achieved using $HMSE_{dn}$.

Some factors that can affect performance of machine learning are:

1. Data size i.e. sufficient training data to capture the anticipated input-output relationship in the test data. For example, neural networks can perform erroneously when extrapolating (Hettiarachchi et al., 2005).
2. Data density i.e. regular availability of data at short depth intervals (Tresp & Briegel, 1998). For example, data which have sensor values recorded at every 1 ft during drilling will likely be better for analysis than having data for which sensor values are recorded every 10 ft. In this work, the data depth interval had a mean value of 1.0504 ft and standard deviation of 0.6243 ft. The total data point is 1905 covering a depth from about 9690 ft to 11690 ft (values given to the nearest foot).

5.7 Conclusions

This chapter presents an approach for utilizing drilling parameters for generating gamma ray log. This involved computing the depth-effect normalized hydro-mechanical specific energy ($HMSE_{dn}$) from drilling parameters. Machine learning was used to capture the relationship between $HMSE_{dn}$ and gamma ray log in order to perform two categories of task: (i) generation of gamma ray log at far end of the well which is required for real time lithology analysis, and (ii) generation of missing sections within the gamma ray log.

Several machine learning algorithms (simple ANN, simple RNN, NARX, LSTM-RNN, GRU network and TCN) were used for generation of gamma ray log at the far end of the well. The test result shows GRU and LSTM-RNN performing poorer than simple RNN. This performance is likely due to the nature of the data used. However, the performance of LSTM-RNN significantly improved when made to use a window of input points for each output point. By nature, the TCN requires its receptive field to be defined and it was observed that the receptive field played a significant role in the performance of the TCN. Therefore, it is recommended that emphases should be placed on obtaining the best receptive field during the development phase of the deep learning algorithm before applying it for prediction. Also, it is important to consider several machine learning algorithms (both simple and complex) and their configurations especially when dealing with data from a single well.

The simple RNN and TCN were used for generating different missing sections within the gamma ray log. The overall performances of both simple RNN and TCN for this category of task were similar. Further investigation is recommended to explore the possibility of using data from several wells to aid machine learning performance.

5.8 Nomenclature

ΔP_b	bit pressure drop (psi)
$\emptyset_0$	mudline porosity
A_b	bit area (in^2)
D_b	bit diameter (in^2)
R_F	receptive field
$HMSE$	hydro-mechanical specific energy (psi)
$HMSE_{dn}$	depth-effect normalized $HMSE$ (psi)
JSA	junk slot area (in^2)
K	compaction coefficient (1/ft)

MSE	mechanical specific energy (psi)
MW	mud weight (PPG)
N	rotary speed (rpm)
η	hydraulic energy reduction factor
Q	flow rate (gpm)
PDC	polycrystalline diamond compact
ROP	rate of penetration (ft/hr)
T	torque (lb-ft)
TFA	flow area (in^2)
WOB	weight on bit (lb)
Z	true vertical depth (ft)

5.9 Acronyms

ANN	artificial neural network
CNN	convolutional neural network
GRU	gated recurrent unit
LSTM	long short-term memory
LWD	logging while drilling
MAPE	mean absolute percentage error
MWD	measurement while drilling
NARX	nonlinear autoregressive network with exogenous inputs
RMSE	root mean square error
RNN	recurrent neural network

TCN temporal convolution network

5.10 References

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essen, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. Electronics, 8(3), 292.

Assaad, F. A. (2008). Field methods for petroleum geologists: A guide to computerized lithostratigraphic correlation charts case study: Northern Africa. Springer Science & Business Media.

Athy, L. F. (1930). Density, porosity, and compaction of sedimentary rocks. Aapg Bulletin, 14(1), 1–24.

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. ArXiv Preprint ArXiv:1803.01271.

Chen, L. (2020). Stock Price Prediction using Adaptive Time Series Forecasting and Machine Learning Algorithms. UCLA.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv Preprint ArXiv:1412.3555.

Clavier, C., Hoyle, W., & Meunier, D. (1971). Quantitative interpretation of thermal neutron decay time logs: part I. Fundamentals and techniques. Journal of Petroleum Technology, 23(06), 743–755.

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (elus). arXiv 2015. ArXiv Preprint ArXiv:1511.07289, 2.

Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179–211.

Foresee, F. D., & Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. Proceedings of International Conference on Neural Networks (ICNN'97), 3, 1930–1935.

Gholami, A., & Ansari, H. R. (2017). Estimation of porosity from seismic attributes using a committee model with bat-inspired optimization algorithm. Journal of Petroleum Science and Engineering, 152, 238–249.

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). Neural network design. PWS Publishing Co.

Han, J. H. (2018). Comparing Models for Time Series Analysis.

Hettiarachchi, P., Hall, M. J., & Minns, A. W. (2005). The extrapolation of artificial neural networks for the modelling of rainfall—runoff relationships. Journal of Hydroinformatics, 7(4), 291–296.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

Kanfar, R., Shaikh, O., Yousefzadeh, M., & Mukerji, T. (2020). Real-Time Well Log Prediction From Drilling Data Using Deep Learning. ArXiv Preprint ArXiv:2001.10156.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. ArXiv Preprint ArXiv:1412.6980.

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. Advances in Neural Information Processing Systems, 971–980.

Kumar, A., Kim, J., Lyndon, D., Fulham, M., & Feng, D. (2017). An ensemble of fine-tuned convolutional neural networks for medical image classification. IEEE Journal of Biomedical and Health Informatics, 21(1), 31–40.

L'heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. M. (2017). Machine learning with big data: Challenges and approaches. IEEE Access, 5, 7776–7797.

Larionov, V. V. (1969). Radiometry of boreholes. Nedra, Moscow, 127.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234, 11–26.

Mathwork. (2020a). Improve Shallow Neural Network Generalization and Avoid Overfitting: Multiple Neural Networks. In https://www.mathworks.com/help/deeplearning/ug/improve-neural-network-generalization-and-avoid-overfitting.html#bt0cnqo-1.

Mathwork. (2020b). Sequence-to-Sequence Classification Using 1-D Convolutions: Define Deep Learning Model. In https://www.mathworks.com/help/deeplearning/ug/sequence-to-sequence-classification-using-1-d-convolutions.html#SeqToSeqClassificationUsing1DConvAndModelFunctionExample-2 .

Olayiwola, A. M., & Bamford, K. M. (2019). Depositional environment and reservoir characterization of the deep offshore Upper Miocene to Early Pliocene Agbada Formation, Niger delta, Nigeria. Journal of African Earth Sciences, 159, 103578.

Oloruntobi, O., Adedigba, S., Khan, F., Chunduru, R., & Butt, S. (2018). Overpressure prediction using the hydro-rotary specific energy concept. Journal of Natural Gas Science and Engineering, 55, 243–253.

Oloruntobi, O., & Butt, S. (2019). Energy-based formation pressure prediction. Journal of Petroleum Science and Engineering, 173, 955–964. https://doi.org/10.1016/j.petrol.2018.10.060

Oloruntobi, O., & Butt, S. (2019). The New Formation Bulk Density Predictions for Siliciclastic Rocks. Journal of Petroleum Science and Engineering. https://doi.org/10.1016/j.petrol.2019.05.017

Oloruntobi, O., & Butt, S. (2020). Application of specific energy for lithology identification. Journal of Petroleum Science and Engineering, 184, 106402.

Oloruntobi, O., & Butt, S. (2020). The shear-wave velocity prediction for sedimentary rocks. Journal of Natural Gas Science and Engineering, 76, 103084.

Oloruntobi, O., Falugba, O., Ekanem-Attah, O., Awa, C., & Butt, S. (2020). The Niger Delta basin fracture pressure prediction. Environmental Earth Sciences. https://doi.org/10.1007/s12665-020-09081-5

Oloruntobi, O., Onalo, D., Adedigba, S., James, L., Chunduru, R., & Butt, S. (2019). Data-driven shear wave velocity prediction model for siliciclastic rocks. Journal of Petroleum Science and Engineering, 106293. https://doi.org/https://doi.org/10.1016/j.petrol.2019.106293

Oloruntobi, O. S. (2019). The pore pressure, bulk density and lithology prediction. Memorial University of Newfoundland.

Oord, A. van den, Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. ArXiv Preprint ArXiv:1609.03499.

Rémy, P. (2020). Keras, Temporal Convolutional Network. In https://github.com/philipperemy/keras-tcn.

Salehi, M. M., Rahmati, M., Karimnezhad, M., & Omidvar, P. (2017). Estimation of the non records logs from existing logs using artificial neural networks. Egyptian Journal of Petroleum, 26(4), 957–968.

Stieber, S. J. (1970). Pulsed neutron capture log evaluation-Louisiana Gulf Coast. Fall Meeting of the Society of Petroleum Engineers of AIME.

Sundermeyer, M., Ney, H., & Schlüter, R. (2015). From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(3), 517–529.

Tang, A., Tam, R., Cadrin-Chênevert, A., Guest, W., Chong, J., Barfett, J., Chepelev, L., Cairns, R., Mitchell, J. R., & Cicero, M. D. (2018). Canadian Association of Radiologists white paper on artificial intelligence in radiology. Canadian Association of Radiologists Journal, 69(2), 120–135.

Tang, J., Fan, B., Xu, G., Xiao, L., Tian, S., Luo, S., & Weitz, D. (2020). A New Tool for Searching Sweet Spots by Using Gradient Boosting Decision Trees and Generative Adversarial Networks. International Petroleum Technology Conference, doi:10.2523/IPTC-19941-Abstract.

Teale, R. (1965). The concept of specific energy in rock drilling. International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts, 2(1), 57–73.

Tresp, V., & Briegel, T. (1998). A solution for missing data in recurrent neural networks with an application to blood glucose prediction. Advances in Neural Information Processing Systems, 971–977.

Yusuf, B., Oloruntobi, O., & Butt, S. (2019). The formation bulk density prediction for intact and fractured siliciclastic rocks. Geodesy and Geodynamics. https://doi.org/https://doi.org/10.1016/j.geog.2019.05.005

Zhang, D., Chen, Y., & Meng, J. (2018). Synthetic well logs generation via Recurrent Neural Networks. Petroleum Exploration and Development, 45(4), 629–639.

Zhou, G.-B., Wu, J., Zhang, C.-L., & Zhou, Z.-H. (2016). Minimal gated unit for recurrent neural networks. International Journal of Automation and Computing, 13(3), 226–234.

Zhou, K., Zhang, J., Ren, Y., Huang, Z., & Zhao, L. (2020). A gradient boosting decision tree algorithm combining synthetic minority oversampling technique for lithology identification. Geophysics, 85(4), WA147–WA158.

Chapter 6

6. Thesis conclusion

6.1 Summary of work done in dissertation

This dissertation presents an integrated approach for monitoring downhole conditions in order to detect drilling hazard based on geo pressure (reservoir pore pressure) during drilling, with deep learning being the primary methodology of interest for this dissertation.

Based on the focus of dissertation, a survey was done to understand the trend in the use of supervised machine learning methodology for pore pressure-based hazard during drilling. Some important outcomes of the research are:

- Deep learning, random forest and support vector machine methodologies are gaining significant increase in their use. This could be due to capabilities of these algorithms i.e. deep learning is recommended for large amount of data while SVM and random forest are recommended for smaller scale data classification.
- The survey presented machine learning methodologies, input parameters and size of training/testing data for kick, fracture, lost circulation, stuck pipe, pore pressure and equivalent circulation density.
- The trend/dynamic nature of drilling data can be exploited in machine learning.
- Researchers are now beginning to use deep learning on drilling parameters for lithology identification, drilling rig state determination, drilling event identification, generating logging/other drilling parameters and detecting abnormality in data (data pre-processing).

Also, CNN and RNN (including its variants e.g. LSTM-RNN) appears to be the most commonly used deep learning algorithms.

- There is absence of publicly accessible global database of drilling data. Researchers typically perform machine learning based on data from a particular oil field. The results of such analysis may not be generalizable.

Based on the survey, machine learning algorithms with the ability to capture trend or sequential relationship in data such as RNN were used in this dissertation. Kick occurrence represents the primary event which can lead to blow out if not well managed. Hence, a data driven approach built on the use of LSTM-RNNs was developed for kick detection. The approach involves obtaining relevant attributes from d-exponent and standpipe pressure data. These attributes were fed to an ensemble of LSTM-RNNs in order to train them for kick detection. The objective of the kick detection approach was to ensure early kick detection and as much as possible to prevent occurrence of false alarms. Field data was used in testing the kick detection methodology developed and the methodology was successful i.e., early kick detection without false alarm were achieved. The sequential relationship learning capabilities of the LSTM-RNN was beneficial. This is because, the use of different configurations (different number of nodes, different number of hidden layers) of simple ANN performed less successfully in comparison to the use of LSTM-RNN.

Kick occurrence can be prevented if pore pressure can be accurately predicted and the right drilling mud weight is used. Based on this, a contribution towards pore pressure prediction was included in this dissertation. A methodology which combines porosity and resistivity was developed for pore pressure prediction. Although porosity and resistivity can be individually used for pore pressure prediction, the aim of the methodology was to explore interrelationship between porosity and resistivity for pore pressure prediction. Reviews of previous articles show that the effect of cementation on pore pressure prediction represents an area of exploration. Archie's cementation factor uses the relationship between porosity and resistivity to explain degree of cementation in a rock formation. This therefore served as a motivation to develop pore pressure prediction as a function of Archie cementation factor (which is a function of both porosity and resistivity). The methodology developed was tested with field data and promising results were achieved i.e., the methodology yielded a better result in comparison to simply averaging of pore pressure obtained by conventional porosity and resistivity approach. However, more work is recommended in order to observe the performance of the methodology as a function of depth (or regions with expected high cementation effects), as well as the most suitable value(s) for the resistivity scaling factor introduced in the methodology. Although machine learning was not used in this part of the dissertation, it is anticipated that machine learning will become viable for pore pressure prediction when much data is made available.

Gamma ray log generation from drilling parameters (hydro-mechanical specific energy) using deep learning is presented in this dissertation. Gamma ray log serves as a means of lithology identification e.g. shaliness of a rock formation. Pore pressure prediction is done in shale lithology; thus it is important to accurately determine if the drilling bit is in a shale formation. The use of drilling parameters offers the benefit of being able to determine lithology at bit point. During machine learning development phase, it is recommended that effort is placed in determining the best window size of sequential input points for each output point.

The contribution of this dissertation towards monitoring downhole conditions during drilling for safety operation can be summarized as:

- Survey work on the application of machine learning for pressure based downhole safety conditions during drilling.
- The use of LSTM-RNN for kick detection using standpipe pressure and d-exponent data.
- Pore pressure prediction in shale lithology using a methodology which integrates porosity and resistivity values.
- Use of deep learning in generating gamma ray log from drilling parameters for identifying shale lithology.

6.2 Suggested future research

Several suggestions are presented below:

6.2.1 Quantitative analyses for kick

In this dissertation, kick detection was qualitative (kick or no-kick). The use of deep learning for quantitative estimation of kick offers the benefit of tracking progress in reservoir fluid influx for efficient well control.

6.2.2 Additional parameter to explore for pore pressure prediction

In this dissertation, the interrelationship between sonic porosity and resistivity were explored for pore pressure prediction. The corrected d-exponent represents another parameter which has been used for pore pressure prediction. To enhance pore pressure prediction, the interrelationship between corrected d-exponent, sonic porosity/resistivity can be explored for pore pressure prediction either with the use of machine learning or by hand handcrafting/theoretical approach or a combination of both.

6.2.3 Other logs for lithology identification

Although the focus in this dissertation in terms of lithology detection was to generate gamma ray log from drilling parameters using deep learning, there are other logs outside of gamma ray log which can be used for identifying lithology. Such logs offer optional means of detecting changes in lithology by the application of deep learning on drilling parameters. Example of such logs include photoelectric absorption log, self-potential log and neutron log.

6.2.4 Data availability

Machine learning (especially deep learning) thrives on the availability of large data for training. Thus, the availability of more open source drilling data from different oil fields is very much welcome. With the availability of more data, transfer learning/pre-training could be explored in utilizing multiple field and well data for gamma ray log generation from drilling parameters (with emphases on hydro mechanical specific energy) and kick detection using relevant attributes of standpipe pressure and d-exponent data. Pore pressure prediction also represents a drilling engineering area which has not benefited much from machine learning.