2020

A Machine Learning and Data-Driven Prediction and Inversion of Reservoir Brittleness from Geophysical Logs and Seismic Signals: A Case Study in Southwest Pennsylvania, Central Appalachian Basin

Tobi Micheal Ore
West Virginia University, tmo0005@mix.wvu.edu

A Machine Learning and Data-Driven Prediction and Inversion of Reservoir Brittleness from Geophysical Logs and Seismic Signals: A Case Study in Southwest Pennsylvania, Central Appalachian Basin

Tobi Ore

**Thesis submitted
to the Eberly College of Arts and Sciences
at West Virginia University**

in partial fulfillment of the requirements for the degree of

**Master of Science in
Geology**

**Dengliang Gao, Ph.D., Chair
Tim Carr, Ph.D.
Jamie Toro, Ph.D.**

Department of Geology and Geography

**Morgantown, West Virginia
2020**

Keywords: machine learning, seismic texture, brittleness, unconventional reservoir

ABSTRACT

A Machine Learning and Data-Driven Prediction and Inversion of Brittleness: A Case Study in Southwest Pennsylvania, Central Appalachian Basin

Tobi Ore

In unconventional reservoir sweet-spot identification, brittleness is an important parameter that is used as an easiness measure of production from low permeability reservoirs. In shaly reservoirs, production is realized from hydraulic fracturing, which depends on how brittle the rock is–as it opens natural fractures and also creates new fractures. A measure of brittleness, brittleness index, is obtained through elastic properties of the rock. In practice, problems arise using this method to predict brittleness because of the limited availability of elastic logs.

To address this issue, machine learning techniques are adopted to predict brittleness at well locations from readily available geophysical logs and spatially using 3D seismic data. The geophysical logs available as input are gamma ray, neutron, sonic, photoelectric factor, and density logs while the seismic is a post-stack time migrated data of high quality.

Support Vector Regression, Gradient Boosting, and Artificial Neural Network are used to predict the brittleness from the geophysical logs and Texture Model Regression to invert the brittleness from the seismic data. The Gradient Boosting outperformed the other algorithms in predicting brittleness. The result of this research further demonstrates the application of machine learning, and how these tools can be leveraged to create data-driven solutions to geophysical problems. Also, the seismic inversion of brittleness shows promising results that will be further investigated in the future.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Dr. Dengliang Gao for accepting me into his research lab and guiding me throughout this research. I greatly benefited from his wealth of experience. I also cherish the opportunity to interact with Dr. Jaime Toro, from whom I have gained from his perspective about issues regarding this research. During the initial conception of this research, I hit a bottleneck due to the unavailability of data but Dr. Tim Carr assisted me by making data available without which this research would never have existed, for this I am thankful.

I would like to thank the Department of Geology and Geography for providing my funding in the form of a teaching assistantship and Occidental Corporation for sponsoring the joint industry bright-spot technology consortium that funded me in the form of a research assistantship. Also, I will like to thank the Society of Exploration Geophysicists and the Geophysical Society of Pittsburgh for their generosity in offering me a scholarship. Without these financial supports, I would not have been able to complete my degree because of the financial burden.

To the staff of the Department of Geology and Geography that has made my stay so far smooth; Lisa, Hope, and Josh, I say a very big thank you. To friends I have met along the way that have made my stay worthwhile, I am grateful.

To the people I hold dear to my heart, whose prayer and support keeps me going; Capt. & Mrs. Dele Ore, Mr. & Mrs. Tarka, my siblings, and Ademide Mabadeje, I love you all.

Finally, thank you to Energy Corporation of America and the Marcellus Shale Energy and Environment Laboratory for the provision of data used in this research, and to Schlumberger for making Petrel® available.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 – GENERAL OVERVIEW

1.0 INTRODUCTION

Advancement and sophistication of technology have significantly impacted the commercial production of hydrocarbons from unconventional reservoirs. Technologies such as horizontal drilling and multistage hydraulic fracturing (stimulation) have made the exploitation of previously untapped resources not only possible but profitable. Hydraulic fracturing is a technique in which fluids are injected at high pressure into the target formation to activate or create a fracture network, thereby increasing the ability for fluids to flow. These techniques have transformed the shale reservoirs into playing a major role in the oil and gas industry and also contributed to tremendous shale gas development and production, thus the shale boom. Shale has become an important reservoir rock, as opposed to the preconceived notion of it being a source rock. To better understand the reservoir properties, several studies have been carried out using various data types such as cores, well logs, seismic, etc.

A useful parameter in unconventional resource (shale) exploration is the brittleness, as brittle rocks will most likely exhibit natural fractures with a high chance of induced fractures. Brittleness which is a function of mineralogy and elastic rock properties is the rock's ability to fail under stress and maintain a fracture once the rock fractures (Rickman et al., 2008). This property is very crucial in the exploitation of unconventional oil and gas, as differentiating between ductile and brittle rocks has become key to stimulation success. The area with high brittleness can be drilling targets and candidates for hydraulic fracturing, as the injected proppant during stimulation will keep fractures open and also fill newly created fractures. However, ductile rocks are believed to deform plastically, therefore when stimulated, seal-off before the proppant can fill the created fracture inhibiting fluid flow.

The estimation of the brittleness of rocks is a contentious issue that has been up for debate over the decade. Several techniques have been proposed, but no one method is universally accepted. This may be due to the contrast between the mineralogy of the same type of rocks or the imminent geomechanical difference due to the rheology of the rocks. However, some of the proposed methods of estimation have theoretical justification and are therefore adopted as a generalization, but care must be taken.

This research attempts to bridge the gap in estimating the spatial and temporal variability of brittleness by adopting a machine learning approach. The major focus is the Middle Devonian Marcellus shale which is part of the Hamilton group. It is located in the Appalachian basin which spans approximately 75,000 square miles in Pennsylvania, New York, Ohio, West Virginia, Virginia, and Maryland (EIA, 2017). The Marcellus shale is a carbonaceous, brittle, soft, and black unit with high pyrite content and beds of limestone. It has a low clay content, high quartz content, and total organic carbon (TOC) of 1-9% (Engelder, 2008).

1.1 STATEMENT OF PROBLEM

Hydraulic fracturing, which is a stimulation technique used to produce hydrocarbon from tight, unconventional reservoirs, depends on the brittleness of the target reservoirs. Brittleness, the ability of a rock to fail under stress, is thus an important parameter that is of high significance in the evaluation of unconventional reservoirs. It is usually estimated from cores and well log data. However, this method only provides estimates around the borehole not to mention the cost of acquiring the data. It is, therefore, imperative to develop a workflow that can effectively predict the brittleness for wells without the right data and away from the well location. This will aid in the optimization of fracture propagation during the stimulation process, and will also serve as an input for reservoir modeling.

1.2 AIM

The goal of this research is to develop a workflow/template that can be adopted in basins with limited data where the brittleness estimation is an important parameter in the sweet spot identification.

1.3 OBJECTIVES

The objectives of this research are to:

1). Estimate the brittleness from the geomechanical information of the well using existing relationships.

2). Develop a machine learning model that predicts the brittleness from other conventional well logs.

3). Process the seismic cube through waveform spectral analysis to sample the reservoir signal at different frequencies to find a relationship between the seismic texture and the brittleness estimates.

1.4 LITERATURE REVIEW

Machine Learning techniques are being applied to solve numerous problems in geosciences. Its effectiveness stems from the fact that compared to conventional numerical models, they do not need to be explicitly programmed, but allow the use of basic mathematical operations such as additions and multiplications “to solve complex, mathematically ill-defined problems, nonlinear problems or stochastic problems” (Graupe, 2007).

In the area of reservoir description, Salehi et al. (2017), Rolon et al. (2009), and Zhang et al. (2018) applied Artificial Neural Networks (ANN) in the generation of synthetic well logs. Tan et al. (2015) applied a support vector regression to predict the total organic carbon content of shale from wireline logs further displaying the predictive capabilities of these algorithms when dealing with geological and geophysical data.

A plethora of techniques exist for the estimation of brittleness. Some involve laboratory stress-strain measurements using core, elastic properties, and mineral content analysis using well logs to estimate brittleness. Wang and Gale (2009) estimated brittleness using mineralogical techniques on core measurements. Grieser and Bray (2007) proposed a brittleness cut-off using Young's modulus and Poisson's ratio calculated from well logs. This method provides a fast and easy estimate because it is much more common to have log data covering the entire interval that will be hydraulically fractured and also widespread. Jarvie et al. (2007), Altindag (2003), Rickman et al. (2008), Goodway et al. (2010) have also attempted to estimate brittleness, but there is no universally accepted technique.

Negara et al. (2017) applied support vector regression in the prediction of brittleness from elemental spectroscopy and petrophysical properties. They found that the laboratory-measured brittleness matched well with the support vector regression prediction. The result of the research proved that the support vector regression is promising in building a brittleness prediction model based on mineralogy logs and petrophysical logs.

In this area of study, Donahoe and Gao (2016) used waveform regression attribute to enhance the visualization of subsurface structures by defining structural discontinuities and deformational fabrics from the seismic data that will be used in this proposed study. This technique was incorporated with variance and ant tracking to resolve a finer map of the fault and fracture network in the study area. This research observed a fore-thrust to back-thrust pattern and detachment faults within the Devonian intervals, including the Marcellus Shale. The primary stress orientation during the Devonian was interpreted to be approximately 105°–120° azimuth, which may affect drilling orientations in the hydraulic fracturing process of the Marcellus gas shale reservoir.

Utilizing the 3D seismic data from the study area, Gao et al. (2018) demonstrated that the structural style in the Devonian section is dominated by east-vergent folds and reverse faults, which contrasts with that in the Valley and Ridge Province where west-vergent folds and thrusts dominate. The Acadian hinterland-vergent faults were linked to the shale gas preservation through a detailed analysis of seismic structures and seismic facies.

The development of the lower Marcellus shale has been very successful because of its siliceous nature which makes it brittle and easy to fracture, and because of its natural fractures (Sumi, 2008). There are two different types of natural fractures present in the Marcellus shale: J1 fractures with a strike of 60-75^{o} and J2 fractures with a strike of 315-345^{o}.

Efforts have been made to expand the characterization of brittleness into three dimensions by the application of seismic data. Perez and Marfurt (2014) used prestack seismic inversion calibrated with a mineralogy-based brittleness template from elemental capture spectroscopy and elastic log measurements. Zhang et al. (2015) applied machine learning algorithms to obtain a classification pattern between elastic properties and brittleness index to generate 3D brittleness estimates through seismic inversion.

Hermana et al. (2019) used a novel method that utilized the scaled inverse quality factor of P-wave (SQp) and scaled inverse quality factor of S-wave (SQs) attributes to estimate the brittleness of rocks. This technique, which was cross-validated with fracture density from formation-imager (FMI) log, showed that there is a significant correlation between brittleness estimated from elastic properties and fracture density.

In this research, for the spatial prediction of brittleness, I utilize a seismic texture analysis algorithm to qualitatively predict brittleness from the seismic signal (Gao, 2011). Seismic texture has several advantages over traditional seismic amplitude. Not only does the texture analysis increase the frequency of the seismic events, but also it involves a statistical filtering process that enhances the signal-to-noise ratio. Gao (2011) developed a texture model regression method, one of the machine-learning algorithms, to enhance the depositional and fracture facies and reservoir qualities related to reservoir brittleness. Since the brittleness is related to the fracture and depositional facies which can be feasibly derived from seismic texture, a correlation is hypothesized to exist between brittleness and seismic texture (Gao et al., 2020).

I begin with a review of the geology of the Appalachian Basin. Next, I estimate the brittleness from the elastic property of the rock calculated from the well logs. I then use Support Vector Regression, Gradient Boosting, and Artificial Neural Network to predict the brittleness for wells without elastic properties. The brittleness at the well locations is then correlated with the seismic texture attribute for the qualitative characterization of the brittleness throughout the reservoir.

1.5 GEOLOGY OF AREA

The study area is located in Greene County, southwestern Pennsylvania, in the Appalachian Plateau (Figure 1). The Appalachian Plateau is associated with three different orogenies that have resulted in the present structural and stratigraphic configuration. These orogenic events initiated in the Ordovician and extended through the Pennsylvanian are the Taconic (Late Ordovician—Early Silurian), Acadian (Middle Devonian—Early Mississippian) and Alleghenian (Late Mississippian—Late Permian) (Park et al., 2010). Paleozoic sediments were deformed as a result of these tectonic events. Underlying these sediments are basement-involved sequences initiated by the Grenville Orogeny (Kulander and Ryder, 2005) followed by the post–Grenville rifting and thermal subsidence (Shumaker and Wilson, 1996). Therefore, the study area is marked by episodes of compression, extension, and subsidence accompanied by transpression and transtension, which occurred during sequential orogenic and quiescent periods. In the Middle Devonian, present-day West Virginia, Pennsylvania, and parts of New York, Virginia, Maryland, and Ohio was a nearly enclosed epicontinental sea (Figure 2) which created restricted marine circulation within the Appalachian Basin that resulted in the deposition of the Marcellus shale (Woodrow, 1985).

The Marcellus shale is an organic-rich black shale that was deposited in a foreland basin during the Acadian orogeny in the Middle Devonian period. This Orogeny resulted in rapid subsidence toward the east near the Acadian mountains and uplift to the west caused by tectonic loading (Ettensohn, 1985). The tectonic loading and subsidence coupled with eustatic sea-level rise led to the accumulation of this organic-rich shale (Lash and Engelder, 2011). According to previous studies the primary minerals are 9% - 35% mixed-layer clays (more abundant in upper member); 10% - 60% quartz, 0% - 10% feldspar, 5% - 13% pyrite (more common toward the base of the formation), 3% - 48% calcite, 0% - 10% dolomite (carbonate minerals much more abundant in the lower Marcellus member), and 0% - 6% gypsum (Avary and Lewis, 2008; Roen, 1984; Wrightstone, 2009).

The Marcellus interval is divided into two members: the lower Marcellus/Union Springs Shale and the upper Marcellus/Oatka Creek Shale. This unit was deposited on the Onondaga Limestone and is overlay by the Mahantango Formation, which together is referred to as the Hamilton Group (Figure 3). The Marcellus exhibits several different pressure regimes across the Appalachian Basin. Generally, Marcellus is under pressure to the southwest and normal-pressured to potentially overpressured to the northeast, with a transitional area in between (Zagorski et al., 2012).

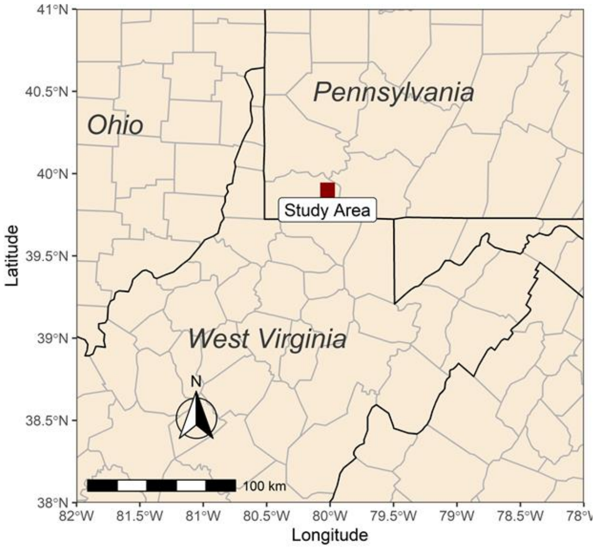

Figure 1. Location map of the study area.

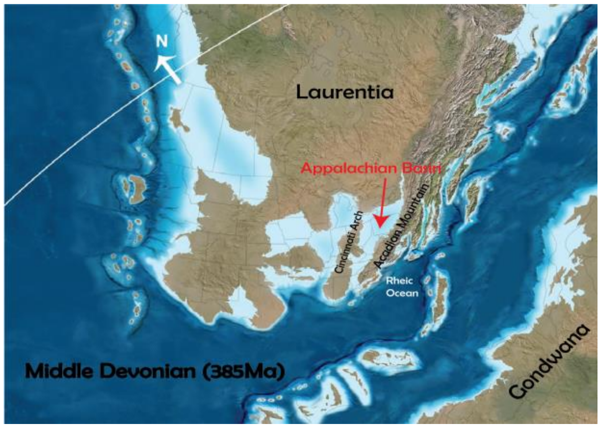

Figure 2. Paleogeography of the study area in the middle Devonian (385Ma) (Modified from Blakey (2010)).

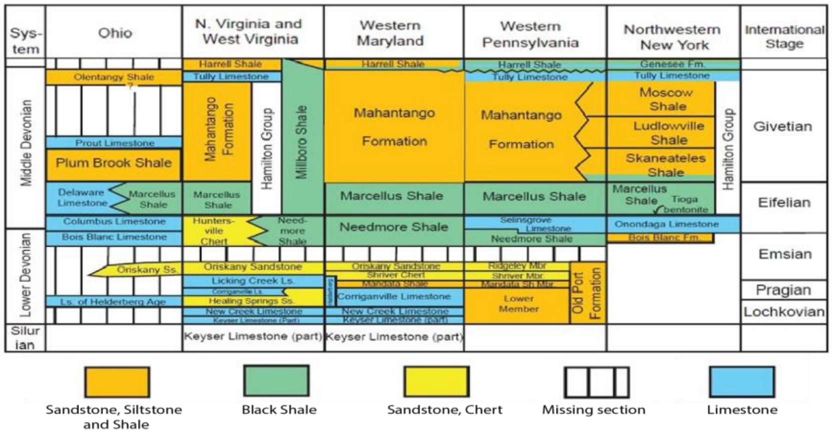

Figure 3. Stratigraphic column of the Appalachian Basin showing the interval of focus. (Modified from Milici and Swezey, 2006)

CHAPTER 2 – WORKFLOW FOR PREDICTION OF BRITTLENESS FROM GEOPHYSICAL LOGS

2.0 INTRODUCTION

In the oil and gas industry, one of the common tasks in the exploration of hydrocarbon is reservoir description. Rock properties, e.g., porosity, permeability, lithology, and formation pressure, are sought to be understood in terms of their spatial distribution and variability. A high-level description of the said reservoir rock is achieved by utilizing geophysical logs, seismic signals, cores, outcrops, and well testing.

A plethora of geophysical logs are in existence and the subsurface information obtained–as stand-alone or in combination with other logs–is enormous. Therefore, these logs are acquired to aid hydrocarbon resource evaluation. However, often, these logs contain incomplete information because of cost limitations and/or borehole problems. Due to the importance of these well logs, a robust workflow needs to be adopted to compensate for missing or incomplete readings.

With the advent of machine learning and sophisticated high-performance computing, a data-driven approach is now being adopted for the prediction of well log measurements. This data-driven approach utilizes the power of machine learning and its capability to solve complex problems without explicitly programming or assuming any initial and boundary conditions of subject matter in physics space. These techniques are fast becoming widely accepted and are now being applied to various aspects of the petroleum industry.

Numerous machine learning algorithms have been widely used in many areas of science. In this study, Support Vector Regression (SVR), Gradient Boosting (XGBoosting), and Artificial Neural Networks (ANN) are used in the prediction of brittleness from geophysical logs. These three techniques were selected out of a host of others because of its history in modeling similar problems, not to forget the computational constraints that have to be factored in when selecting suitable algorithms for a task.

The purpose of this study is to test and select the most effective machine learning workflow to predict the brittleness from geophysical logs. The resulting model is then used to forecast the unknown brittleness for other wells with the known geophysical logs. Our concept and workflow have general applicability to derive any unknown reservoir property logs through machine learning from available logs.

2.1 DATASET

There are 6 wells available in the seismic survey area of this research. These wells have geophysical logs that vary in availability. Geophysical logs are measurements of the physical properties of rock formations in the subsurface. The design of this project is greatly affected by the availability of logs in the wells (Table 1), as only logs present in all wells are used for the research. This is due to the nature of the model structure. To bridge the gap of increasing the dataset to train an unbiased model, 2 wells outside the seismic survey area with full sonic-wave log suites are included. These wells shown in figure 4, are the MIP3H well (20 miles from Whipkey well) and the Boggess well (15 miles from Whipkey well). Lateral correlation of the lithology using the Gamma Ray log also confirms the physical relationship between the different wells, and that these new two wells will act as new reliable data to train the model (Figure 5).

Table 1. Crossmatch table showing the availability of geophysical logs in the wells.

Well Name	Gamma Ray	Density	Photoelectric Effect	Neutron	Sonic (Compressional)	Sonic (Shear)
St Whipkey	🟩	🟩	🟩	🟩	🟩	🟩
MIP 3H	🟩	🟩	🟩	🟩	🟩	🟩
Boggess	🟩	🟩	🟩	🟩	🟩	🟩
Cree	🟩	🟩	🟥	🟩	🟥	🟥
37-059-24212	🟩	🟩	🟥	🟩	🟩	🟥
Corry	🟩	🟩	🟥	🟩	🟥	🟥
Mohr	🟩	🟩	🟥	🟩	🟥	🟥
V Virgili	🟩	🟩	🟩	🟩	🟥	🟥

These geophysical logs were selected to serve as a basis to predict the brittleness because of their ability to reflect geomechanical properties variation in the rocks.

A gamma-ray log records the natural radioactivity of a formation and is used primarily as a lithology identifier. The geomechanical properties of a rock depend on the mineralogical constituents which are in turn reflected as a variation in lithology.

The density log is a measurement of the rock’s bulk density. It is proportional to porosity, shale content, and the matrix rock type and serves as an important component in most geomechanical calculations.

Photoelectric factor logs are important mineralogy identifiers. They are acquired based on the principle that the photoelectric absorption properties of various elements are different, and can thus serve as a lithology identifier also.

Neutron log's main use is in the determination of porosity, and this is due to the sensitivity of the logs to hydrogen atoms in the formation. The logs are measurements from the interactions of a neutron flux from a source with the rocks and fluids.

Sonic logs provide the formation’s interval transit time, which is a measure of the formation capacity to transmit seismic waves. This capacity is also related to the formation’s lithology, porosity, and texture.

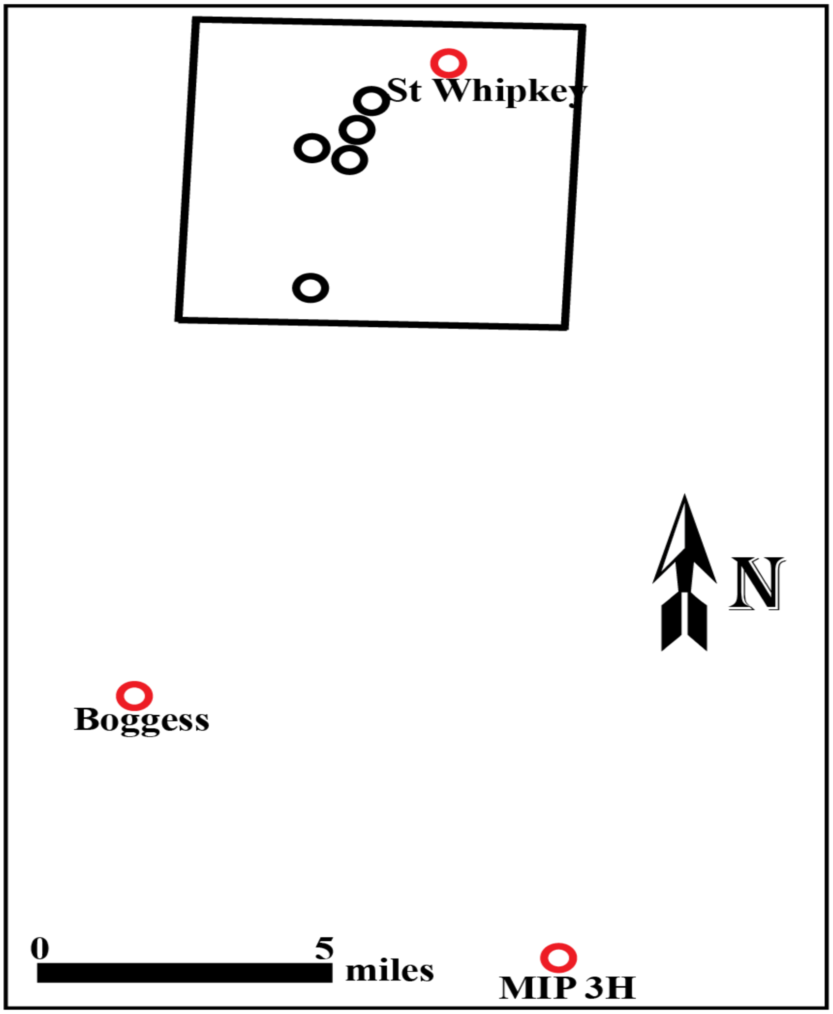

Figure 4. A base map that shows the relative location of all the wells used in this study.

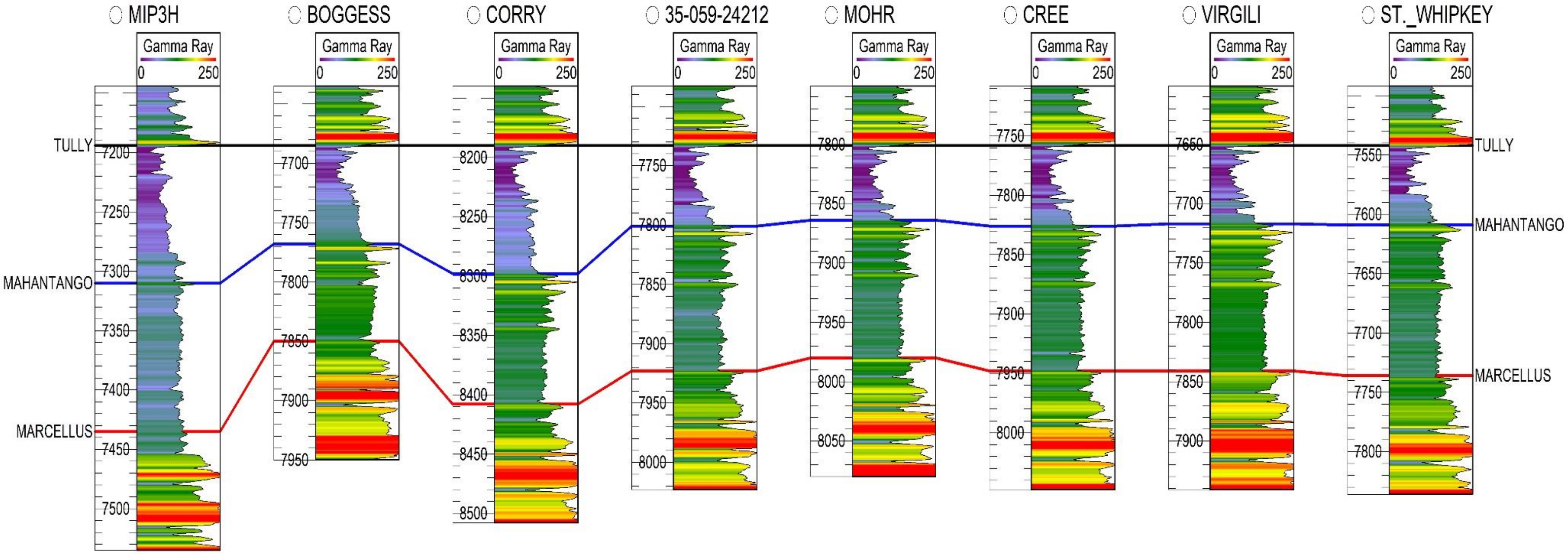

Figure 5. A well correlation pane of the eight wells showing the lateral continuity of the lithology.

2.2 BRITTLENESS ESTIMATION

Brittleness is the measurement of stored energy in rock before failure. It is a function of rock strength, lithology, texture, effective stress, temperature, fluid type (Handin and Hager, 1957, 1958; Handin et al., 1963; Davis and Reynolds, 1996), diagenesis and total organic carbon (Wells, 2004). Typically, brittle rocks fail under a certain level of differential stress which creates planes of weakness. They have a large region of elastic behavior but only a small region of ductile behavior (Figure 6). This is very important in hydraulic fracturing because these planes of weakness are then kept open by the injected proppant, causing secondary permeability in the rock. In contrast, malleable rocks deform plastically and can undergo significant strain before fracture. This causes the fractures in plastic rocks to close about the proppant, resulting in the sealing of pathways to fluid flow (Perez and Marfurt, 2013).

Rocks with high brittleness usually possess natural and hydraulically induced fractures. However, brittle rocks may lack natural fractures entirely if the loading history is insufficient to drive fracture growth (Gale et al., 2007). Generally, the design and implementation of the hydraulic stimulation of the well will determine if the rocks will exhibit hydraulically induced fractures.

There are many techniques used in estimating brittleness, due to the ambiguity in the definition and relativity in the estimation of brittleness. This has led to no standardized universally accepted brittleness concept or a measurement method defining or estimating the rock brittleness (Altindag and Guney, 2010). The two techniques that have been widely applied in the literature to estimate the brittleness is described below. The first method utilizes the mineral composition of the rocks, while the second uses the elastic properties.

The estimation of brittleness from the mineral content depends on the fact that the presence of quartz minerals in a rock increases the brittleness while clay minerals do the opposite. Also, rock physics gives a justification for why these two methods have been relatively accepted. It shows that the mineral composition of rocks determines the elastic properties of rock.

The concept of the estimation of brittleness as the "Brittleness Index" (BI) was first proposed by Jarvie et al. (2007). They defined the BI as the ratio of quartz to the sum of quartz, clay, and calcite (Equation 1). Following the work of Wells (2004) that showed that the presence of dolomite tends to increase the brittleness of shale and organic matter tends to increase ductility, Wang and Gale (2009) included dolomite and TOC in the relationship (Equation 2).

$$\boldsymbol{BI_j = \frac{Qtz}{Qtz + Cal + Cly}} \tag{1}$$

$$\boldsymbol{BI_w = \frac{Qtz + Dol}{Qtz + Dol + Cal + Cly + TOC}} \tag{2}$$

Where BI is the Brittleness Index, Qtz is the quartz content, Cal is the calcite content, Cly is the clay content by weight in the rock, and TOC is the total organic carbon content.

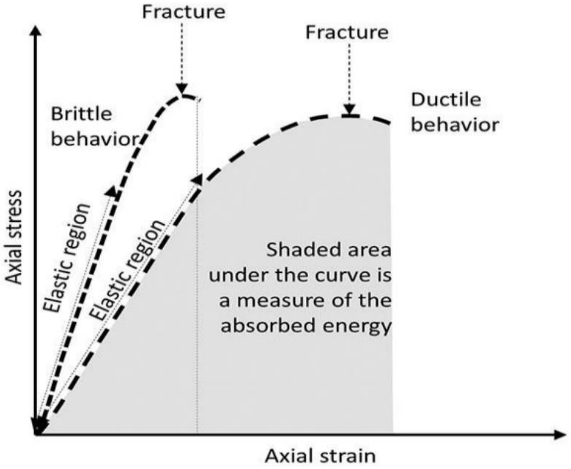

Figure 6. Brittle versus ductile behavior of rock samples as seen on a stress-strain graph (Chopra and Sharma, 2015).

As two important elastic properties, Young's modulus and Poisson's ratio are here used to differentiate ductile rocks from brittle rocks in a relationship called the brittleness average proposed by Rickman et al. (2008). The brittleness average is calculated by first normalizing Young's modulus and Poisson's ratio by their ranges, to produce a scaled elastic parameter, and their average values. It is believed that ductile rocks exhibit a low Young's modulus and high Poisson's ratio, whereas brittle rocks exhibit a moderate to high Young's modulus and low Poisson's ratio (Grieser and Bray, 2007; Rickman et al., 2008).

Young's modulus and Poisson ratio are estimated from sonic and density log measurement. It follows from the wave equation (Equations 3 & 4) that the speed of wave propagation is related to the moduli of the material.

$$\boldsymbol{V_p = \sqrt{\frac{M}{\rho}}} \quad (3)$$

$$\boldsymbol{V_s = \sqrt{\frac{G}{\rho}}} \quad (4)$$

Where V_p is the P-wave velocity (m/s), V_s is the S-wave velocity (m/s), M is the Compressional modulus (Pascals), G is the shear modulus (Pascals) and ρ is the bulk density (kg/m^3).

These wave velocities are used to estimate Young's modulus and Poisson's ratio using equations 5 & 6:

$$\boldsymbol{E = \rho V_s^2 \left[\frac{\left(3V_p^2 - 4V_s^2\right)}{\left(V_p^2 - V_s^2\right)}\right]} \quad (5)$$

$$\boldsymbol{v = 0.5 \left[\frac{\left(V_p^2 - 2V_s^2\right)}{\left(V_p^2 - V_s^2\right)}\right]} \quad (6)$$

Where E is Young's Modulus and v is Poisson's ratio

The calculated E and v are then normalized by a technique previously described using equation 7 & 8 before applying it to estimate the brittleness with equation 9:

$$\boldsymbol{E_{brittleness} = \frac{E - E_{min}}{E_{max} - E_{min}}} \qquad (7)$$

$$\boldsymbol{v_{brittleness} = \frac{v - v_{min}}{v_{max} - v_{min}}} \qquad (8)$$

$$\boldsymbol{BA = \frac{E_{brittleness} + v_{brittleness}}{2}} \qquad (9)$$

Where E_{min} is the minimum Young's Modulus, E_{max} is the maximum Young's Modulus, v_{min} is the minimum Poisson's ratio, v_{max} is the maximum Poisson's ratio and BA is the brittleness average

This research focuses on the brittleness estimated from the elastic properties. One drawback of this technique is that the P-wave and S-wave velocities are required to estimate the elastic properties. These velocities are obtained from a Full-waveform and dipole logging tool. Unfortunately, these tools are not always used in boreholes and when they are used it is generally deployed in the intervals of interest. There always seems to be ample P-wave sonic and density logs, but not S-wave sonic logs. To bridge this gap, three machine learning algorithms are selected and utilized to build the most effective predictive models on a comparative basis.

2.3 MODEL BUILDING

Due to the unavailability of data, the brittleness can only be estimated for one well in the seismic survey. This is greatly due to the cost of acquisition of full-waveform sonic log among other reasons. Therefore, a template needs to be created to predict the brittleness from other readily available logs in wells without a full-waveform sonic log. This, previously, was a challenge due to the non-linear relationships between these logs. Not forgetting the number of mathematical simplification and assumptions needed to properly model the predicted property. With the proliferation of several machine learning techniques, this has become a problem of the past. The model in the research is developed using a support vector regression (SVR), gradient boosting (XGBoosting), and artificial neural network (ANN) architecture.

2.3.1 *Support Vector Regression*

The support vector regressor uses the same principles as the support vector machine (SVM) with the only difference being that the SVR is used as an estimator while the SVM is a classifier. The principle behind SVM is that in an N-dimensional space, a hyperplane can be drawn to distinguish the different classes.

Given a training input *x* with corresponding target values *y*, the support vector regression looks for a function $\boldsymbol{f(x) = (w, \Phi(x))}$, with w and $\Phi(x)$ being vectors that minimize the regularized risk (Cortes and Vapnik, 1995):

$$\boldsymbol{R[f] = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} L\big(y, f(x)\big)}, \qquad (10)$$

The constant C determines the trade-off between the flatness of *f* and the amount up to which deviations larger than ϵ are tolerated. $\boldsymbol{L\big(y, f(x)\big)}$ is either the linear ϵ-insensitive loss (L1-SVR) function defined by equation 11

$$\boldsymbol{L\big(y, f(x)\big) = |y - f(x)|_{\epsilon} = \max(0, |y - f(x)| - \epsilon)}, \qquad (11)$$

Or quadratic ϵ-insensitive loss (L2-SVR) defined by equation 12.

$$\boldsymbol{L\big(y, f(x)\big) = |y - f(x)|_{\epsilon}^{2}}. \qquad (12)$$

The optimization problem of minimizing the loss function is solved by setting up a dual problem and using Lagrange multipliers. The formulation is beyond the scope of this research, but the method will be adopted in the algorithm used in training the SVR.

2.3.2 *Gradient Boosting*

This is a machine learning technique that produces a predictive model by combining several weak decision trees in a process called an ensemble. It builds the model in a sequential, stage-wise fashion allowing for the optimization of a suitable cost function. The decision trees go from input features of the data, represented in the branches, to the target/output feature, represented in the leaves. In Gradient Boosting, the subsequent decision tree learns from the mistake of the previous branch, combining the results with an appropriate weight. This model is made up of three components; a loss function, a weak prediction model, and an additive model to combine the weak models (Brownlee, 2016).

Assume that a weak model F_m exists at the first iteration of training which can predict the mean of the target, the gradient boosting algorithm improves on F_m by constructing a new model that adds an estimator *h* to create a better model F_{m+1}. *h* is referred to as the residual of the previous model.

To describe what the gradient boosting algorithm does, consider a training set $\{x_i, y_i\}_{i=1}^{n}$, a differentiable loss function $L\big(y, F(x)\big)$, according to Hastie et al. (2009);

Initialize model with a constant value:

$$\boldsymbol{F_0(x) = arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)} \quad (13)$$

For m=1 to M where M is the number of iterations:

Compute the pseudo-residuals:

$$\boldsymbol{r_{im}} = -\left[\frac{\partial \boldsymbol{L}(\boldsymbol{y_i}, \boldsymbol{F}(\boldsymbol{x_i}))}{\partial \boldsymbol{F}(\boldsymbol{x_i})}\right]_{\boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{F_{m-1}}(\boldsymbol{x})} \quad for\ \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{n}.$$

Fit a weak learner $\boldsymbol{h_m(x)}$ to pseudo-residuals

Compute multiplier $\boldsymbol{\gamma_m}$ by solving $\boldsymbol{F_0}(\boldsymbol{x}) = \boldsymbol{arg} \min_{\boldsymbol{\gamma}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{L}(\boldsymbol{y_i}, \boldsymbol{F_{m-1}}(\boldsymbol{x_i}) + \boldsymbol{\gamma} \boldsymbol{h_m}(\boldsymbol{x_i}))$

Update the model

$$\boldsymbol{F_m}(\boldsymbol{x}) = \boldsymbol{F_{m-1}}(\boldsymbol{x}) + \boldsymbol{\gamma_m} \boldsymbol{h_m}(\boldsymbol{x})$$

Output the trained model $\boldsymbol{F_M}(\boldsymbol{x})$

In this research, a popular implementation of the gradient boosting, called Extreme Gradient Boosting (XGBoost) will be used. The XGBoost, developed by Chen and Guestrin (2016), is widely adopted because of its scalability, handling of missing data, and clever penalization of trees.

2.3.3 *Artificial Neural Network*

The concept behind the ANN was inspired by the neurons in the human body. These neurons are interconnected and transfer information from one cell body to another through the synaptic connections at the end of the axon (Figure 7). Each neuron consists of hundreds of synapses which results in the complex, highly connected nature of the neural network system of the human body (Graupe, 2007).

In the 1940s, Warren McCulloch and Walter Pitts proposed a simplified mathematical model of the biological neurons and showed that, in principle, they can compute any arithmetic or logical function (McCulloch & Pitts, 1943). In the field of neural networks, they are generally regarded as pioneers as the future advancement was motivated by their proposal. This led to the development of the perceptron algorithm by Rosenblatt (1957). Rosenblatt proved that a perceptron can separate the input into classes provided they are linearly separable. A limitation of this single layer perceptrons was published in a book called “Perceptrons” by Marvin Minsky and Seymour Papert (Minsky and Papert, 1969). They showed that elementary calculations such as simple “exclusive or” (XOR) problems cannot be solved by single layer Perceptrons. These claims by Minsky and Papert resulted in a waning effect on the acceptance and adoption of neural networks. It was not until Hopefield’s work on new learning algorithms such as back-propagation, neural networks hit the ground running again. The growth of neural network research and applications has been phenomenal since this revival.

Artificial Neural networks, as described above, mimics the biological nervous system. They are made up of several elements working together interconnectedly (Figure 7). These neurons receive inputs from other sources, perform a generally nonlinear operation on them, and then output the results.

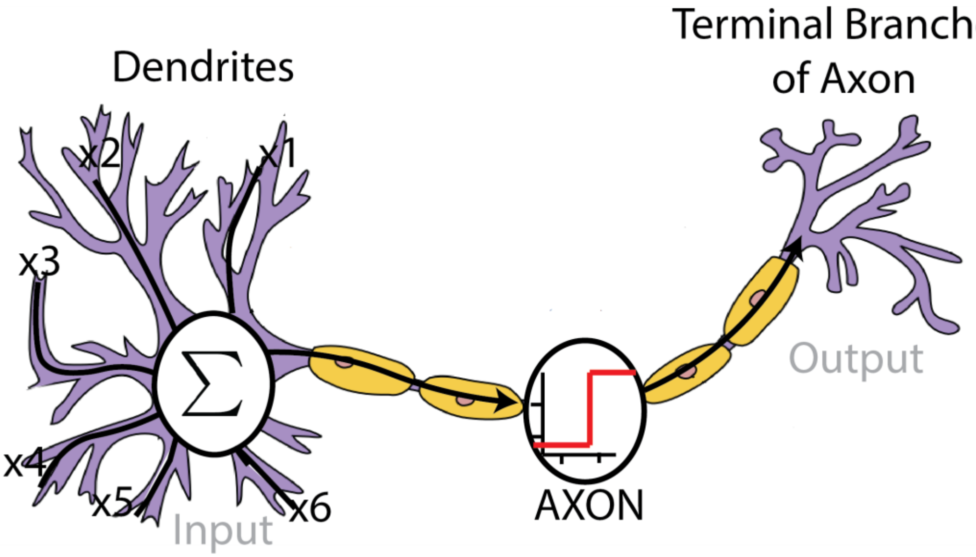

Figure 7. General representation of a single neuron of an ANN and the comparison with a biological nerve.

A typical ANN consists of three main parts: an input layer, a hidden layer(s), and an output layer (Figure 8). The input layer in this network is the leftmost layer, and the neuron with the layer is called input neurons. The rightmost layer is called the output layer (with the output neuron(s)) and the middle layer is called the hidden layer(s).

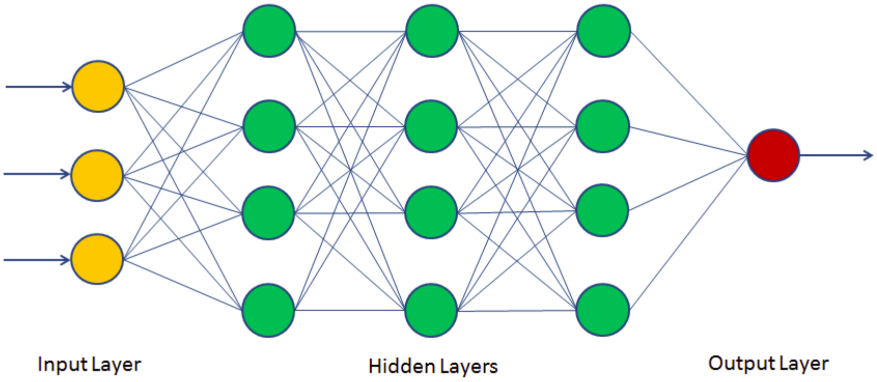

Figure 8. Generalized ANN architecture.

These parts are connected by a set of synapses that have individual weights. A signal $\boldsymbol{x_j}$ at the input of synapse $\boldsymbol{j}$ connected to neuron $\boldsymbol{k}$ is multiplied by the synaptic weight $\boldsymbol{w_{kj}}$. The weighted signals are then summed and a bias ($\boldsymbol{b_k}$) is introduced before operating by an activation function to introduce non-linearity into the output of the neuron.

Mathematically, the neuron can be expressed as:

$$\boldsymbol{z_k = \sum_{j=1}^{n} w_{kj}x_j + b_k} \qquad (14)$$

$$\boldsymbol{y_k = \sigma(z_k)} \qquad (15)$$

Where x_j is the input signal, n is the number of input, w_{kj} is the weight of neuron k, z_k is the linear output of the input signal, b_k is the bias, $\sigma(\mathrm{x})$ is the activation function, and y_k is the output signal of the neuron.

In literature, these neurons have been combined in various ways resulting in several types of architecture. Some of these architectures are recurrent neural networks, multilayer feedforward neural networks, convolutional neural networks, and generative adversarial networks. These algorithms have been applied to a wide range of problems, and the selection of the architecture to adopt for a project depends on the nature of the data, the objective of the project and experience.

In this research, the multilayer feedforward neural networks (MLP) is adopted because of the wide acceptance of its capability to solve similar problems, not to mention the extensive application of MLP in the field of geosciences and petroleum engineering.

The feedforward systems operate in two steps. The first step involves training the neural network with the input. Here the weights for each synapse are randomly set and the input from the input layer is transferred to the hidden layer(s) then to the output layer. The second step is the application of a learning algorithm called backpropagation. In this step, the desired output is compared with the predicted output by the network and an error is estimated. This error is propagated backward from the output layer to the input layer and the weights of each synapse are adjusted. This general process is iterative, and it stops when the error generated becomes considerably small, thus referring to the system as a trained neural network. Feedforward networks have no connections that loop making them relatively easy to implement.

2.4 CORRELATION BETWEEN FEATURES

The strength of association between the input features and the output feature is analyzed to observe any linear or non-linear relationships. This is carried out by utilizing statistical techniques for bivariate analysis. The value of correlation ranges from +1 to -1, where a value of ± 1 indicates a perfect degree of association between the two variables and the strength of the relationship between the variables reduces as the correlation coefficient tends to 0. Also, the sign of the coefficient signifies the direction of the relationship; a "+ sign" indicates a positive relationship, and a "– sign" indicates a negative relationship. In this research, two types of correlation will be measured; Pearson correlation and Spearman correlation.

Pearson correlation is the most widely used correlation statistics to measure the degree of the relationship between linearly related variables. It is based on the method of covariance and gives information about the magnitude of the association as well as the direction of the relationship. Equation 16 is used to calculate the Pearson correlation:

$$\boldsymbol{r_{xy}} = \frac{\boldsymbol{n\sum x_i y_i - \sum x_i \sum y_i}}{\sqrt{\boldsymbol{n\sum x_i^2 - (\sum x_i)^2}}\sqrt{\boldsymbol{n\sum y_i^2 - (\sum y_i)^2}}} \qquad (16)$$

Where r_{xy} is the Pearson correlation coefficient between x and y, n is the number of observations, x_i is the value of x for the *ith* observation, and y_i is the value of y for the *ith* observation.

Spearman rank correlation is a non-parametric test that is used to measure the degree of monotonic association between two variables. The Spearman rank correlation test does not carry any assumptions about the distribution of the data and is the appropriate correlation analysis when the variables are measured on a scale that is at least ordinal.

Equation 17 is used to calculate the Spearman rank correlation:

$$\boldsymbol{\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}} \qquad (17)$$

Where ρ is the Spearman rank correlation, d_i is the difference between the ranks of corresponding variables, and n is the number of observations.

2.5 DATA PREPROCESSING

It is common practice in the machine learning realm to preprocess the raw data, therefore, transforming it into a format that will produce a more efficient and accurate model. Typically, the data is partitioned into three segments: training, calibration, and validation. The training dataset is the set of data used to train the model and create a relationship between the input and output variables.

The calibration dataset serves as an over-watch for the training process. It determines the quality of the different epochs. After every epoch of training, the weights are saved and the network is tested against the calibration dataset to see if the neural network is learning and getting better. This performance is measured by using one or more metrics such as Mean Square Error, R^2 score, etc. The calibration dataset also serves as a means to prevent overtraining.

The validation or test dataset plays an important role in accessing the generalization capabilities of the trained neural network. They are the subset of the data that the neural network has not seen and was not involved in the training process of the model.

2.6 DATA NORMALIZATION

This is an important step before building and training machine learning models. It involves normalizing the range of independent features of the data. This is important in improving the performance of machine learning algorithms where the range of values of raw data varies widely.

There are several normalization/scaling techniques available to preprocess data. In this research, I utilize the MinMaxScaler which transforms each feature to a specific range. For each value in a feature, the MinMaxScaler subtracts the minimum value in the feature and is then divided by the range. The range is the difference between the original data maximum and the original data minimum (equation 18).

$$\boldsymbol{\hat{x}} = \frac{\boldsymbol{x - min(x)}}{\boldsymbol{max(x) - min(x)}} \qquad (18)$$

Where x is the original value, and $\hat{x}$ is the normalized value.

This normalization technique scales and translates each feature individually such that it is the range of 0 and 1. This scaling doesn't change the shape of the original distribution, therefore, doesn't meaningfully change the information embedded in the original data.

2.7 HYPERPARAMETER TUNING

Hyperparameters are parameters that are not directly learned within the model. They are passed to the model during development and determine the performance of the model. These parameters need to be tuned for several reasons, some of which are:

- Bias and variance trade-off: the complexity and depth of a machine learning model determine how well the model will generalize. A bias model is developed when the model is built in such a way that it has high accuracy on the training set, but poorly on the test set. While the variance is affected by restricting the model complexity to avoid memorization, the model will perform poorly on the development set. A sweet spot exists between these two scenarios and it can be achieved by tuning the hyperparameters, as every hyperparameter affects the bias and variance.

- Convergence and or speedup learning: In the context of machine learning, convergence is the point when the training process stops because no better model can be trained. How fast the training approaches this convergence is controlled by the hyperparameter called the learning rate.
- Saddle points and local minima: saddle points and local minima are bottlenecks when training neural networks. They have a gradient of zero, therefore making the model stuck without room for improvement. Tuning hyperparameters such as learning rate and momentum will prevent this from happening.

Hyperparameters considered in this research:

Support Vector Regression

1. C: This is the regularization parameter and it determines the strength of the regularization as it is inversely proportional to C.
2. Gamma: Kernel coefficient.

Gradient Boosting

1. Max_depth: This is the maximum number of nodes allowed from the root to the farthest leaf of a tress. The ability for the model to learn complex relationship depends on the depth, however, deeper trees are susceptible to overfitting.
2. Min_child_weight: This is the minimum weight required for a new node to be created in the tree. More branches are created in the tree if the min_child_weight is small, but the problem of overfitting can also arise.
3. N_estimator: this is the number of gradients boosted trees which is equivalent to the number of boosting rounds.

4. Learning_rate: this is related to the weights of the nodes and determines how fast the boosting learning will reach a minimum.

Artificial Neural Network

1. Number of hidden layers: The robustness of the data and nature of the problem sought out to be modeled must be considered when selecting the appropriate number of hidden layers. Also, as described above, when too small they lead to high bias and when too large lead to the problem of overfitting.
2. Number of hidden units per layer: These also have to be tuned to find the perfect combination that will result in a network properly trained with output in between high bias and high variance.
3. Solver: This is the algorithm the neural network will use to update the weights of every layer after each epoch. Some popular algorithms which will be tested are; stochastic gradient descent, Adam and lbfgs (Limited-memory Broyden-Fletcher-Goldfarb-Shanno). Describing how these algorithm works are beyond the scope of this research.
4. Activation Function: This function defines the output of the node in the layer of the neural network. The popular choice which will be used are ReLU ($f(x) = max(0, x)$), tahn ($\boldsymbol{f(x) = tanh(x)}$) and logistic sigmoid ($\boldsymbol{f(x) = \frac{1}{(1+e^{-x})}}$).
5. Learning Rate: As discussed in the importance of hyperparameter tuning, the learning rate is a core hyperparameter that must be properly tuned for optimum performance of the model. Values that will be tested are between 10^{-3} and 1 with an increment in the power of 10.
6. Number of iterations: This is a hyperparameter that defines the number of times the learning algorithm will train over the entire training dataset. That is, one iteration/epoch

means that each sample in the training dataset has had a chance to update the internal model parameters (weight). This plays an important role in how well the model fits on the training dataset. This is usually tuned based on computational power and time.

Cross validating the models for different hyperparameters manually is a nightmare as the combinations usually lead to hundreds or thousands of distinct models. Also, it may lead to variance problems where the accuracy obtained on one test is very different to the accuracy obtained on another test set using the same algorithm.

A more efficient way to train the model and tune the hyperparameter is using Grid Search Cross-Validation. The grid search is the process of performing hyperparameter tuning to determine the best hyperparameter combinations to optimize the model. While the cross-validation is a solution to the variance problem, where the data is divided into K folds. Out of the K folds, the first set is used for training while the remaining set is used for testing. This process is repeated K times, with every iteration using a new set for training. Figure 9 shows 5-fold cross-validation where 5 models are fitted on distinct yet partly overlapping training sets and evaluated on non-overlapping validation sets. The performance is computed as the arithmetic mean of all the 5 performance estimates from the validation sets.

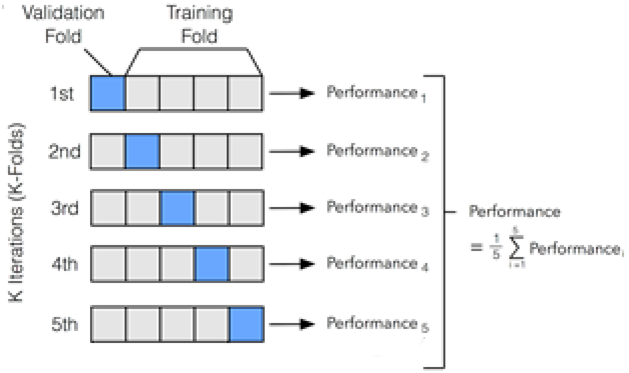

Figure 9. Schematic diagram of the structure of 5-fold cross-validation.

2.8 ESTIMATING MODEL PERFORMANCE

There exist a lot of statistical techniques used to evaluate the performance of a model. They are based on some sort of relationship that measures the error between the observed values and the predicted values. Some of these metrics are: Root mean squared error, R-squared score, Mean squared error, and Mean absolute error. Throughout this research, the R-squared score will be used as the error metrics for model performance.

The R-squared (R^2) or the coefficient of determination is the proportion of the variance in the dependent variable that is predictable from the independent variable. The closer the value of the R^2 is to 1 the more accurate the model is at predicting the variables. The R^2 score is calculated using equation 19:

$$\boldsymbol{R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}} \qquad (19)$$

Where N is the number of observations in the data set, y_i is the observed/measured value, $\hat{y}_i$ is the predicted value, and $\bar{y}_i$ is the mean of the observed value ($\boldsymbol{\bar{y}_i = \frac{1}{N}\sum_{i=1}^{N} y_i}$).

2.9 RESULTS AND DISCUSSION

2.9.1 *Elastic Based Brittleness Estimation*

The brittleness of the interval of interest was estimated using the technique described in the previous section. A display of the result is shown in Figures 10, 11, and 12 along with the plot of the other wells used in the machine learning prediction. As expected, the sonic is related to the mineralogy of the formation and is in turn shown by the inverse relationship with the brittleness.

It can be seen that calcite greatly increases the brittleness and reduces the sonic response of the formation. When compared to the overlying and underlying formations, the Marcellus shale relatively has low brittleness. However, there is also a variation in brittleness within the Marcellus itself. This is due to the heterogeneity of the Marcellus shale and the presence of calcite.

Figures 10, 11, and 12 further show visually how all the selected geophysical logs are related and how they are a good fit to serve as the input features for the prediction of brittleness.

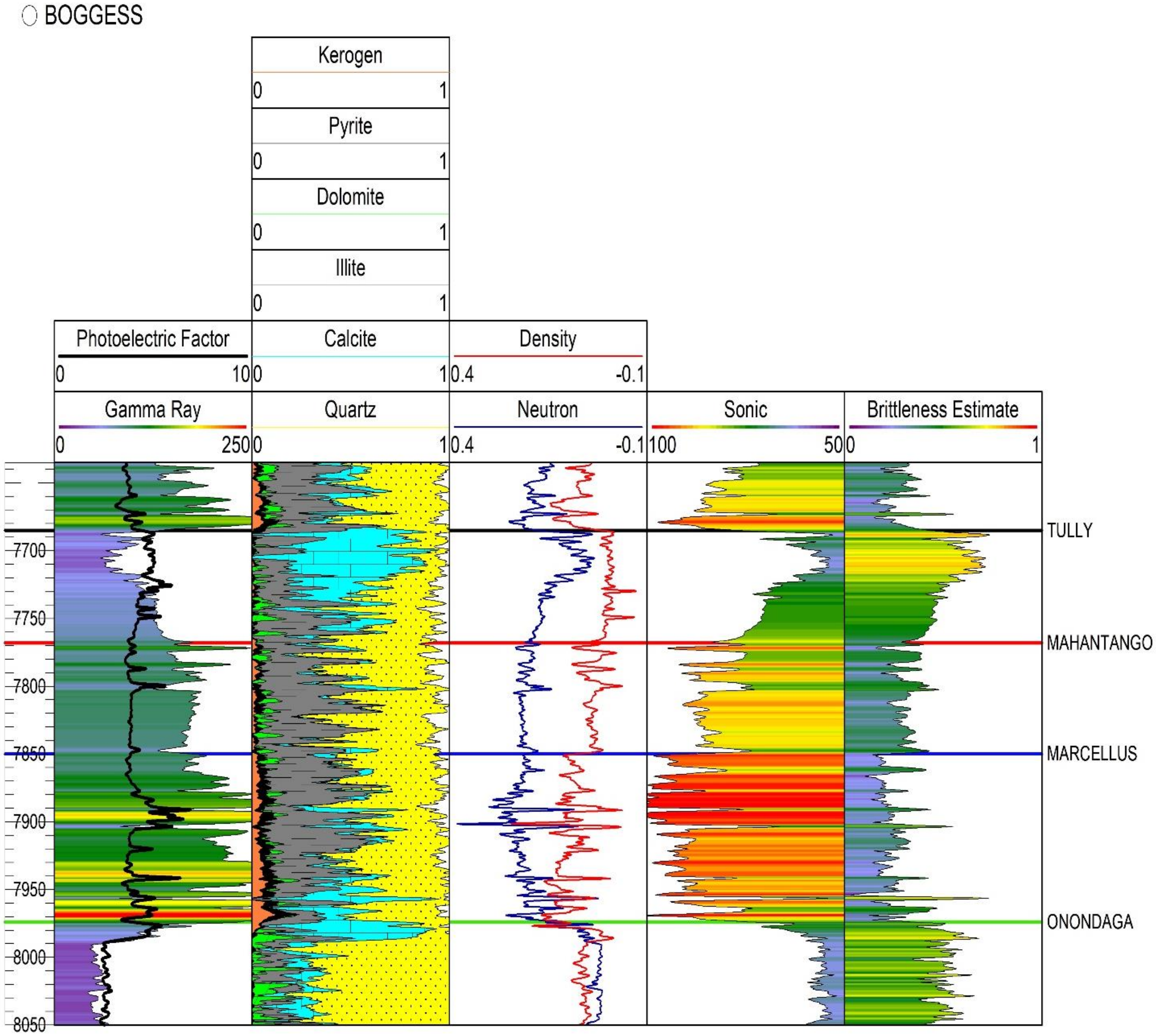

Figure 10. Geophysical logs and estimated brittleness display for the Boggess well.

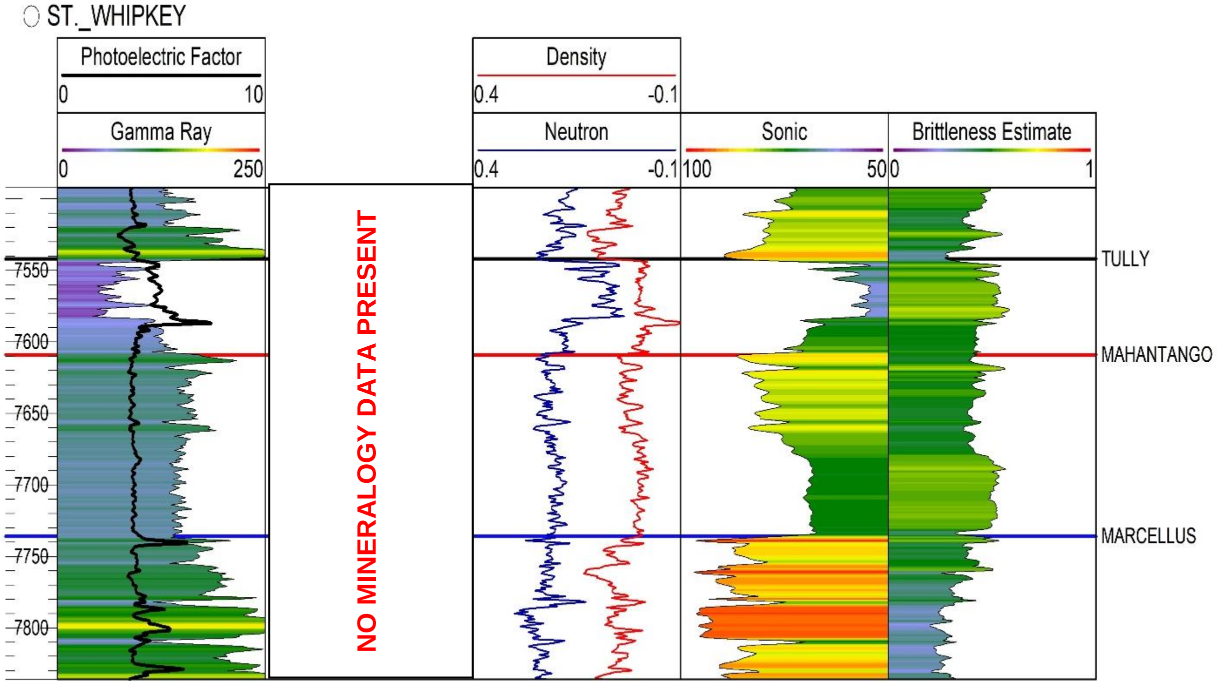

Figure 11. Geophysical logs and estimated brittleness display for the Whipkey well.

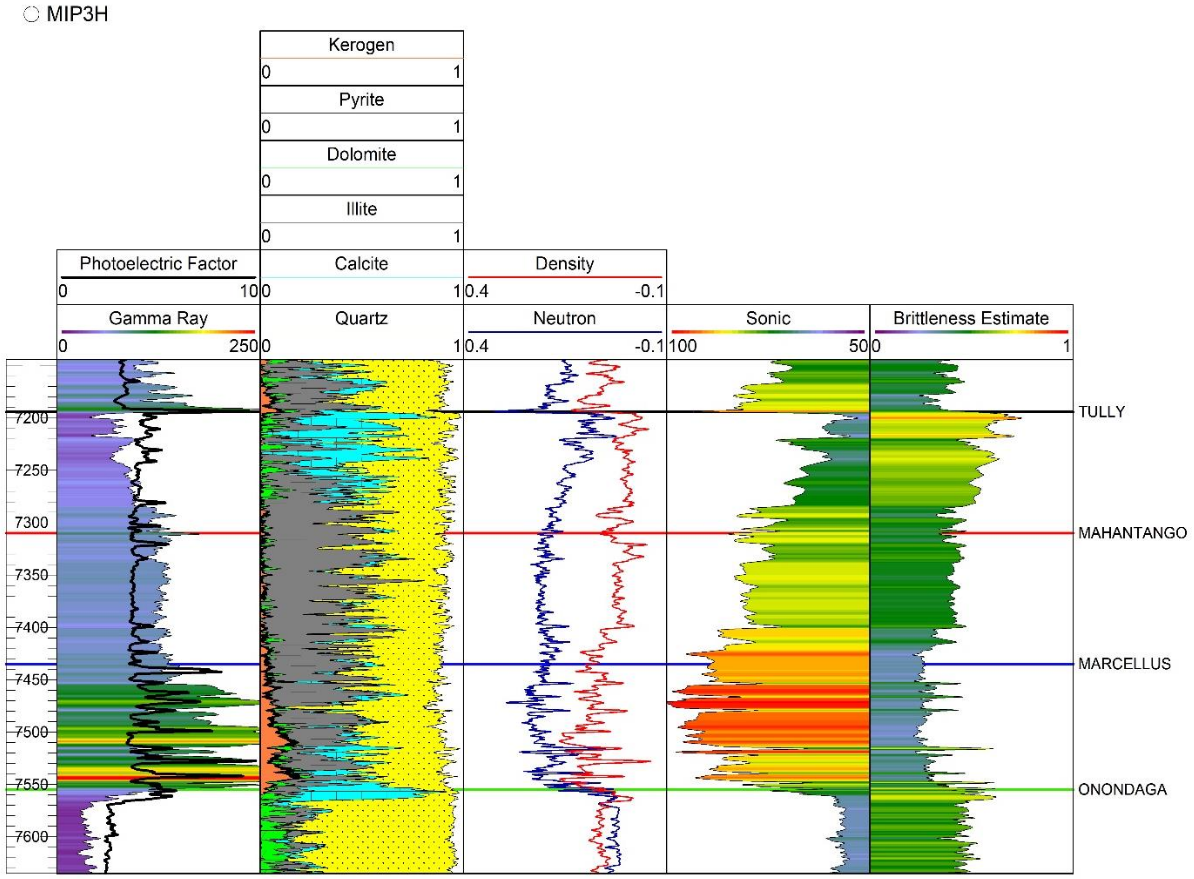

Figure 12. Geophysical logs and estimated brittleness display for the MIP3H well.

2.9.2 *Feature Selection*

The result of the Pearson's correlation shows that the brittleness estimate has a large association with the sonic (-0.7096), Gamma ray (-0.5364) and Neutron (-0.5162) but with the other variables, it had a low correlation (Density (+0.3759) and Photoelectric Factor (+0.0349)). Also, Neutron, Gamma ray, and Sonic showed an intrinsic large linear relationship (Figure 13a).

These features were selected intuitively because of their inherent properties that are related to the brittleness of the formation. However, the Photoelectric Factor in which response is related to mineralogy showed low correlation which is not in agreement with the definition of brittleness, as related to mineralogy. To this end, to further investigate the non-linearity in the relations of the input features and the output, the spearman rank correlation was calculated. This further hinted that there might exist an apparent relationship between the brittleness estimation with Photoelectric Factor (Figure 13b).

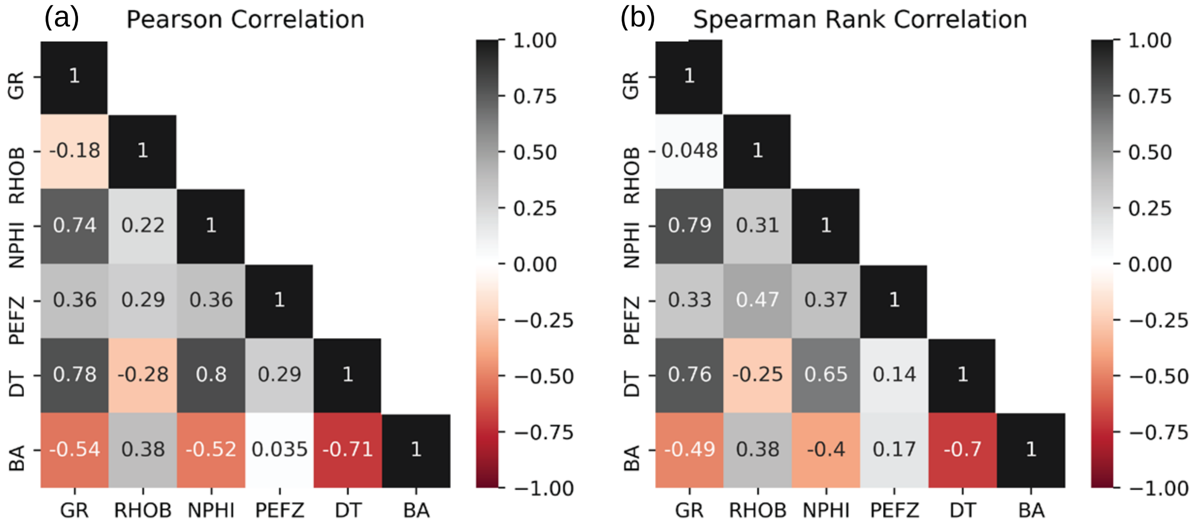

Figure 13. Heat map showing the (a) Pearson correlation and (b) Spearman rank correlation between the input and output features.

To visualize the relationship between these features and brittleness, cross-plotting was utilized. Also, these cross-plots were used to ocularly identify outliers, given the strain it imposes on model performance. Figure 14 shows the trend of how brittleness is changing with respect to other features.

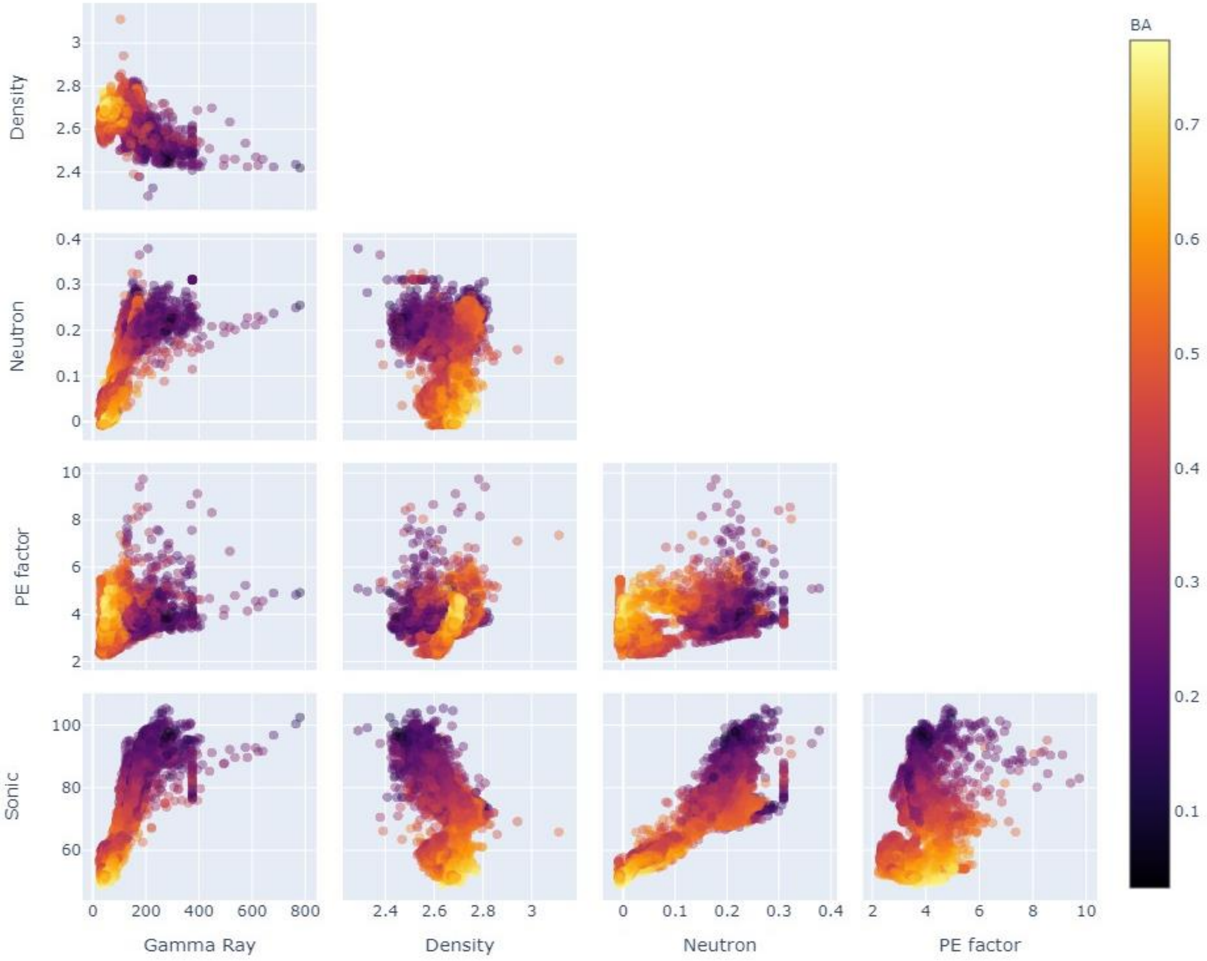

Figure 14. A crossplot of all the geophysical features with an overlay of the estimated brittleness.

2.9.3 *Data Preprocessing*

Comparing the statistics of the features as shown in table 2 shows that the range of the features is of different magnitude; the range of gamma ray is 756, density is 0.82, neutron porosity is 0.38, photoelectric factor is 7.51, and sonic is 105 and brittleness average is 0.74. Also, the standard deviation indicates that using these raw features without normalization will have an adverse effect on the performance of the machine learning model. Scaling input and output variables is a critical step in using machine learning models. In practice, it is nearly always advantageous to apply pre-processing transformations to the input data before it is used to train a machine learning model.

Table 2. Dataset summary before normalization.

	GR	RHOB	NPHI	PEFZ	DT	BA
count	5990	5990	5990	5990	5990	5990
mean	137	2.67	0.17	4	72	0.42
std	63	0.07	0.08	0.69	11	0.12
min	24	2.29	-0.01	2.27	49	0.03
25%	103	2.62	0.14	3.52	68	0.34
50%	143	2.68	0.20	3.79	73	0.44
75%	164	2.73	0.22	4.04	79	0.51
max	781	3.12	0.38	9.78	105	0.77

The MinMaxScaler is used to normalize the range of all the features to 1. Also, this normalization changes the standard deviation of the dataset to the same magnitude, thereby making them more suitable for the model consumption for optimal performance. A summary of the new data statistics is shown in table 3.

Table 3. Dataset summary after normalization.

	GR	RHOB	NPHI	PEFZ	DT	BA
count	5990	5990	5990	5990	5990	5990
mean	0.15	0.46	0.45	0.20	0.42	0.53
std	0.08	0.09	0.20	0.09	0.19	0.16
min	0	0	0	0	0	0
25%	0.10	0.39	0.37	0.17	0.34	0.41
50%	0.16	0.47	0.52	0.20	0.42	0.55
75%	0.18	0.53	0.58	0.24	0.53	0.65
max	1	1	1	1	1	1

Contrary to the conventional partitioning of the data into three subsets; training, calibration, and testing, the dataset was first divided into two (training and testing). This is due to the nature of the training algorithm, which utilizes a cross-validation training scheme. Having an explicit validation/calibration dataset will be counterintuitive.

The dataset is made up of 5,990 records. After splitting into training and test set using a 70%-30% scheme, the training set now contains 4,193 records while the test set has 1,797 records.

2.9.4 *Model Results*

The machine learning models were built in the python programming language utilizing the Scikit-learn library (Appendix I). This library is well-maintained and popular because of its ease of use, and the wide range of already programmed machine learning modules.

The Support Vector Regressor (SVR) exists as an estimator in the Support Vector Machines module of sklearn, the artificial neural network (ANN) was built using the MLPRegressor( ) model in the neural network module, while the gradient boosting (XGBoosting) was built using an efficient, and stable gradient boosting algorithm that exists in the XGBOOST library.

Scenario 1

Support Vector Regression

The SVR was built and trained using gridsearch cross-validation. The parameters which were tuned are the regularization parameter (‘C’) and the kernel coefficient (‘gamma’). After 5-fold cross-validation, it was found that the best parameter combination was {'C': 10, 'gamma': 10}. The best score for the cross-validation is 0.76 with an average fit time of 3.3 seconds. After deployment of the model, the training score was 0.78 and the testing score was 0.77 with the result of the predicted and estimated brittleness shown in figure 15.

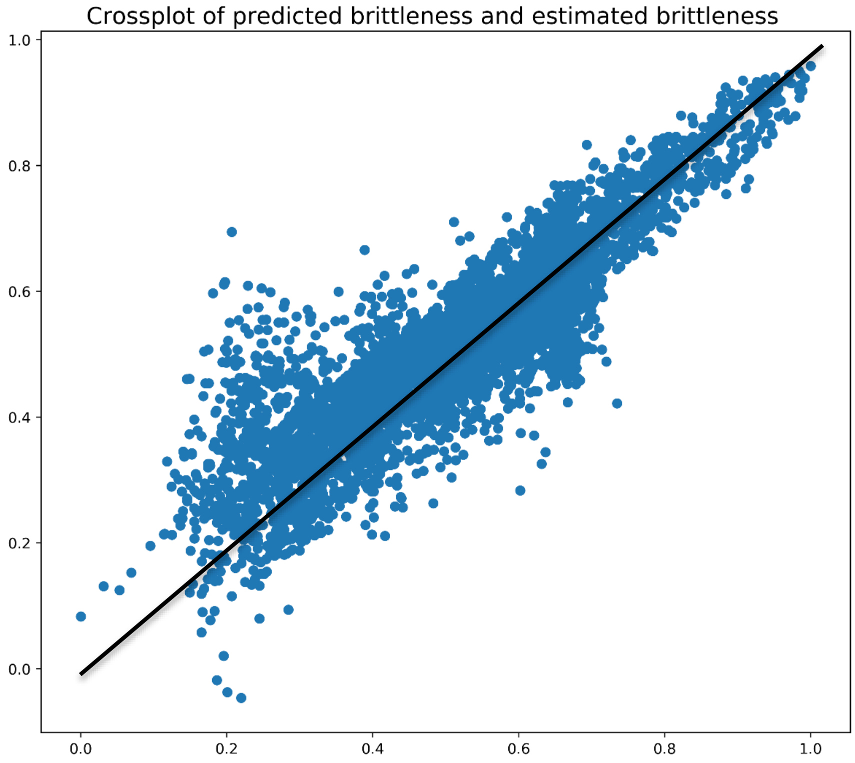

Figure 15. A crossplot showing the relationship between the predicted brittleness using the SVR model and the estimated brittleness from the elastic property

Gradient Boosting

The Gradient Boosting algorithm utilized decision trees as the estimators. The parameters which were tuned are the number of estimators (‘n_estimator’), the learning rate, the minimum child weight (‘min_child_weight’), and the maximum depth (‘max_depth’). After a 3-fold cross-validation, it was found that the best parameter combination was {‘max_depth’: 8, ‘n_estimators’: 100, ‘learning_rate’: 0.07, ‘min_child_weight’: 6}. The training score was 0.95 while after deployment of the model, and the testing score was 0.85 with a crossplot of the predicted and actual brittleness in figure 17.

This model result displayed improved performance in predicting the brittleness as compared to the SVR model. Another added advantage of using gradient boosting is that the decision tree can easily be visualized. Also, the importance of each of the features in building the model can be investigated. This importance provides a scoring metric to show how valuable each feature was in the construction of the model as the more an attribute is used to make key decisions within the decision tree, the higher its relative importance. The importance is calculated for a single decision tree by the amount that each attribute split point improves the performance measure, weighted by the number of observations the node is responsible for, and then averaged across all the decision trees within the model. Figure 16 shows that Sonic has the most influence in deciding how the trees were built. This is expected as the brittleness estimate is closely related to the sonic as the elastic properties, calculated from the sonic, served as the basis for the estimation of the brittleness. It is worthy to note that the photoelectric factor showed little to no correlation with the brittleness estimate in the initial feature correlation. However, it was the second most important feature.

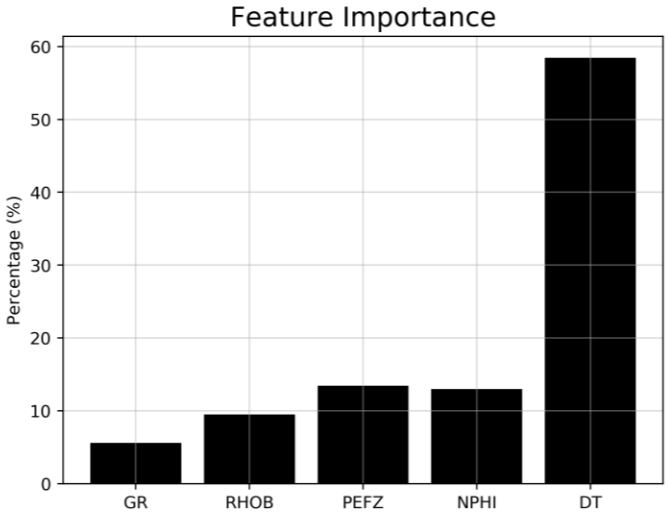

Figure 16.	A histogram showing the significance of each feature in training the decision tree.

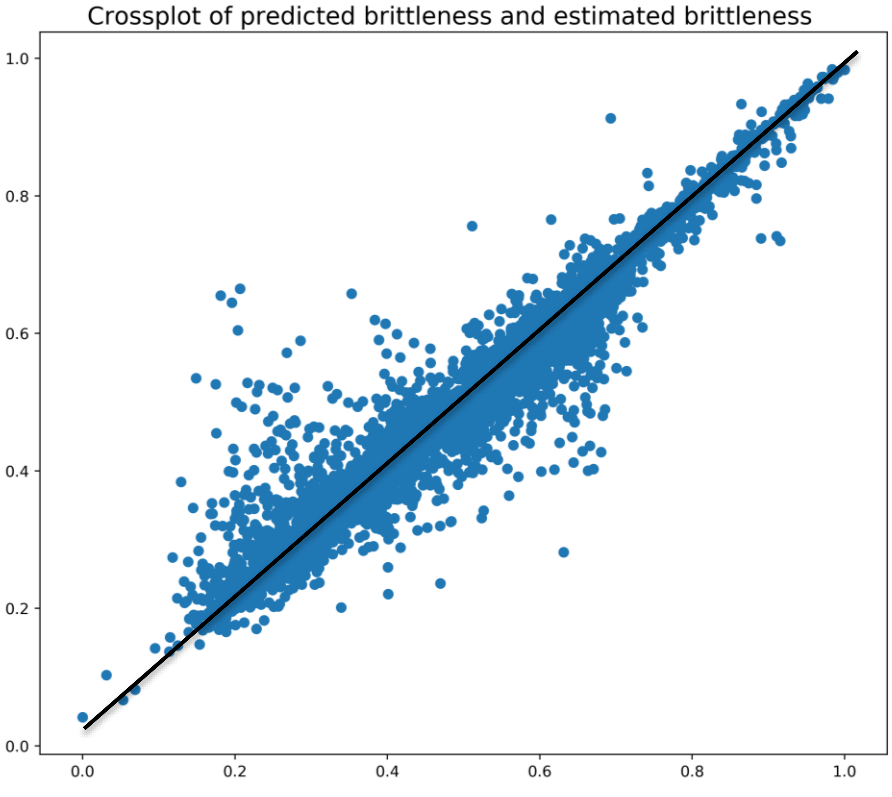

Figure 17.	A crossplot showing the relationship between the predicted brittleness using the XGBoost model and the estimated brittleness from the elastic property.

Artificial Neural Network

The ANN model was built with caution, as they are very susceptible to overfitting. The majority of the publications available in this domain with high accuracy are a product of this effect because, given the right combinations of parameters, a neural network can model any problem. Therefore, the predictive ability of this model depends on whether the model is learning or just memorizing the data. Due to the limited number of features and records, the number of neurons in the hidden layer was trained with several numbers not exceeding 20 where 15 was found to be the best parameter, with a maximum iteration of 500 and a tolerance of 0.0001. The best score from the 3-fold cross-validation of this model is 0.7561 with a test score of 0.7524 on redeployment on the training set and 0.7428 on the blind set. A linear trend can be seen in figure 18 which a crossplot of the predicted and actual brittleness.

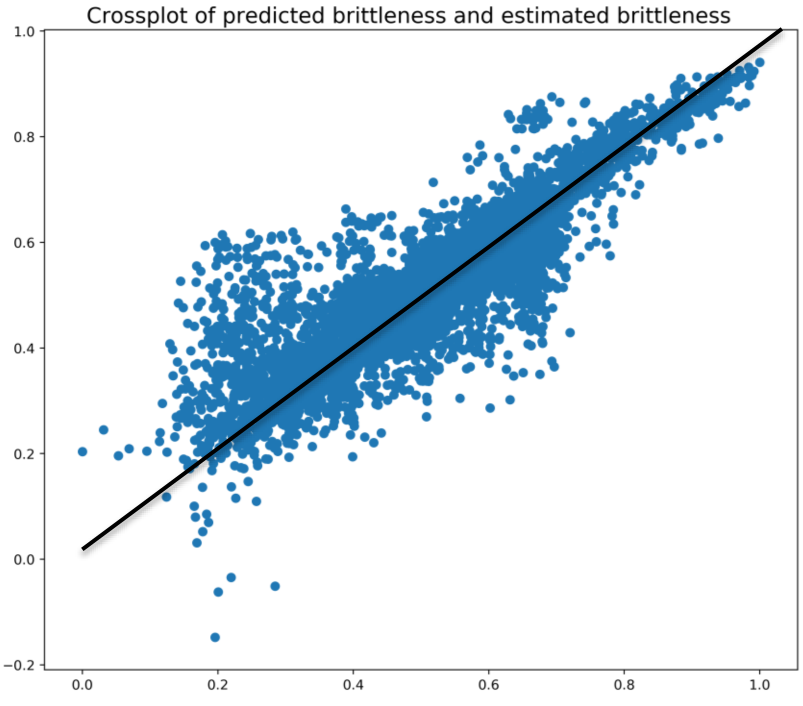

Figure 18. A crossplot showing the relationship between the predicted brittleness using the ANN model and the estimated brittleness from the elastic property.

The SVR, ANN, and XGBoost performed relatively well in modeling the brittleness. However, the models appeared to be affected in regions with low gamma ray. This might be due to training data bias, as the majority of the training sample is representative of formation response with high gamma ray. The effect of the gamma ray with regards to the model performance can be seen in figure 19.

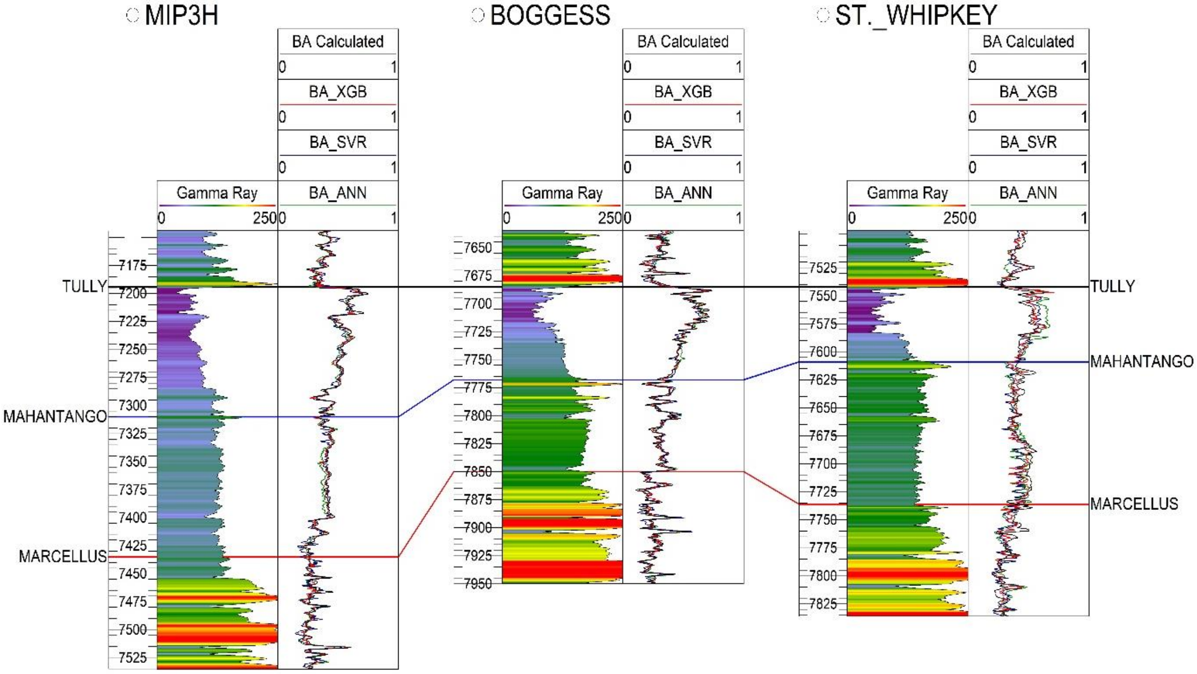

Figure 19. A display of all the different model results and the actual brittleness with reference to the gamma ray.

Scenario 2

After setting a baseline for the models using the available geophysical logs in the training data, a second scenario was modeled with only logs in all the wells. These logs were gamma ray, neutron, and density. The same step as before was carried out, however, deployment of the model cannot be verified as there are no brittleness estimations in all the wells.

The SVR model was trained with the best parameter for ‘C’ and ‘gamma’ being 10 and 100 respectively. The best score for the cross-validation is 0.63 and after deployment, the training score was 0.67 and the testing score was 0.63.

The best parameters for the XGBoosting model were ‘max_depth’ of 5, ‘min_child_weight’ of 6 and ‘n_estimators’ of 100 with the Gamma Ray log being the most influential feature. The cross-validation best score was 0.66 while the training score was 0.75 and the test score was 0.65.

A one hidden layer ANN was trained with 20 neurons in the hidden layer. The cross-validation best score was 0.59, training score 0.60, and test score of 0.59.

When comparing the results, though the three models have relatively close and good R^2 scores, their prediction quality on the other wells not used in the training varies (Figure 20). The evaluation of this result in terms of the prediction quality is sensitive because there are no actual brittleness values for the other wells. Therefore, this is done on a basis of comparison with the other wells with the estimated brittleness and if the prediction makes actual geological sense.

On this basis, the ANN and SVR model appears to fail. Though the SVR model performs well on the 3 wells used for training (Figure 20), on the other wells, it doesn’t make sensible predictions (Figure 21). This maybe is due to, in parts, the algorithm not finding suitable relationships between the features due to limited training data. Also, the ANN model had some negative predicted values. This doesn’t make physical sense as the brittleness estimate is between 0 and 1. Therefore, moving forward, with the availability of more data, the xgboost would be the algorithm of choice to model similar problems.

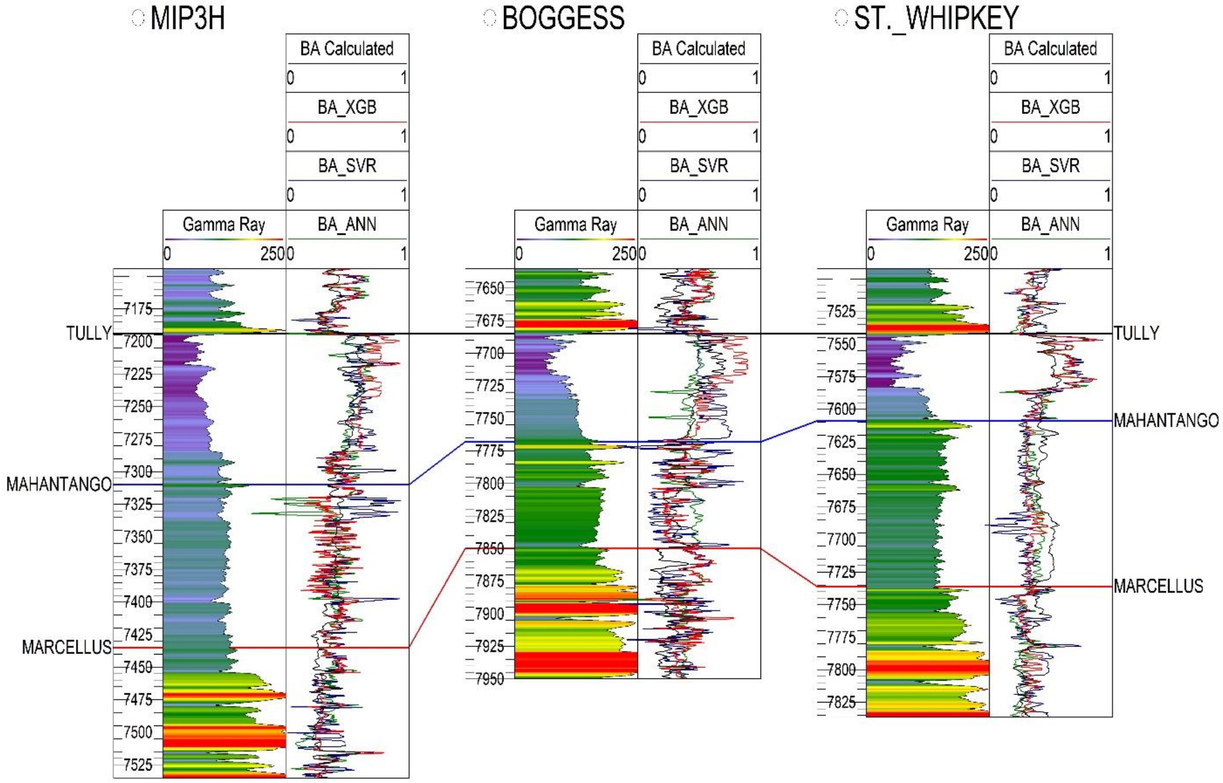

Figure 20. Display of the model prediction for the training wells.

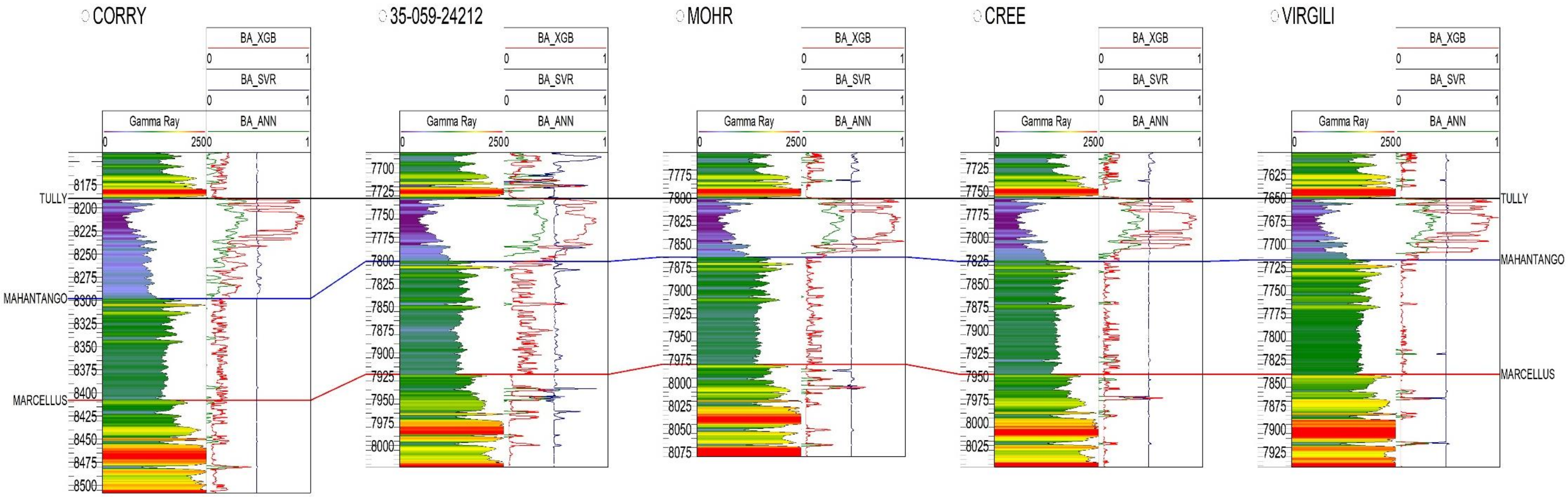

Figure 21. Display of the model prediction when deployed on other wells in the research.

2.10 CONCLUSION

Brittleness–which is a function of mineralogy and elastic properties–is a very useful parameter in unconventional resource exploration as areas with high brittleness can be candidates for hydraulic fracturing. The estimation of the brittleness of rocks is a contentious issue that has been debated over the decade, and several techniques have been proposed. In this research, the elastic property approach is adopted to estimate brittleness as this bears a more physical relationship.

Support Vector Regression, Gradient Boosting, and Artificial Neural Network are applied to predict brittleness for wells without elastic logs. These algorithms present an opportunity for a more robust characterization of brittleness from geophysical logs. Gamma ray, density, neutron, photoelectric, and sonic logs are the selected logs used for the building and training of the predictive models based on preliminary feature selection which showed their correlation and relationship with brittleness. To extend this research to calibrating the 3D seismic data for a spatial brittleness estimation, the brittleness of the other wells in the seismic block was estimated using the three algorithms. However, the input features were affected by the availability of logs. The input features used were gamma ray, density, and neutron.

The Gradient Boosting outperformed the Support Vector Regressor and Artificial Neural Network in terms of accuracy on predicting the blind/test set. Also in the second scenario, the Gradient Boosting was the best algorithm when comparing the quality of the prediction. The result of this research further emphasizes that neural networks are not a panacea of all modeling problems, and other algorithms should be explored before accepting the results.

CHAPTER 3 – WORKFLOW FOR INVERTING BRITTLENESS FROM SEISMIC TEXTURES

3.1 INTRODUCTION

3D seismic data have proven to be an effective tool in the exploration and development of oil and gas as it has been used to understand subsurface geology, and also recognize various stratigraphic and structural features. These features have been generally enhanced by the use of attributes. Over the past years, several workers have developed attribute extraction algorithms that have proven to be useful and widely applied by geophysicists.

Seismic responses are affected by the properties of the rock in the subsurface, such as porosity, fluid saturation, and texture. Also, fractures in rocks significantly affect the seismic response as they facilitate compressibility which lowers the velocity and impedance. Therefore, brittleness is physically linked to the seismic signal texture. Seismic texture analysis has been proven to be useful in enhancing interpretation capabilities and in extracting quantitative information in subsurface reservoirs (Chopra, 2005; Gao, 2004, 2008).

Gao (2011) described an algorithm that carries out a complex trace analysis for reservoir property prediction, thereby, treating the seismic amplitudes as analytic signals and classifying the seismic signal texture for each sample of the seismic data. This seismic texture has conceptually been evolved and extended from the texture analysis of seismic data first introduced by Love and Simaan (1984) where the seismic texture is referred to as the lateral and vertical changes in amplitude and waveform. The algorithm, referred to as the Texture Model Regression (TMR), is utilized in this research to predict the brittleness of the Marcellus shale reservoir interval in untested areas away from the well location (Gao, 2011; Gao et al., 2020).

3.2 DATASET

The dataset utilized is a 3D seismic data and the brittleness estimation at the well location from the previous chapter. The 3D post-stack seismic data is approximately 25 mi^2 in Greene County, southwest Pennsylvania acquired with a lateral resolution (bin size) of 110 ft, dominant frequency of 25 Hz, and a vertical sampling rate of 2 ms. There are 332 inlines and 296 crosslines in the survey. The survey encompasses a group of wells variably spaced (Figure 22).

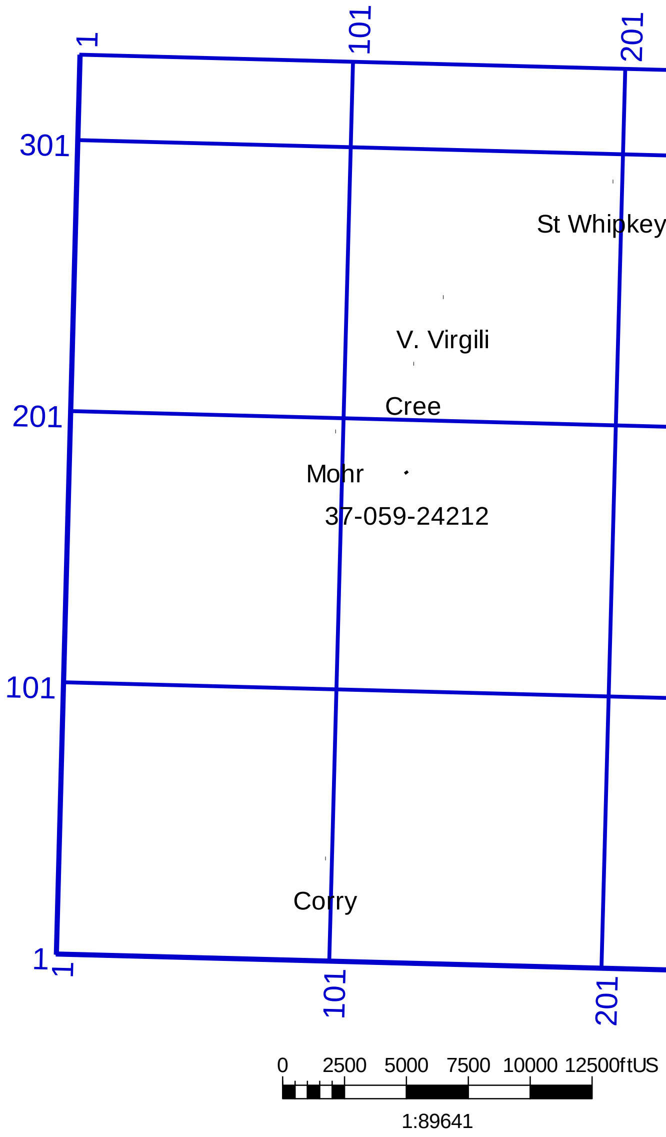

Figure 22. Basemap of the seismic survey with well locations.

3.3 METHODOLOGY

3.3.1 *Preliminary Structural Interpretation*

The preliminary structural interpretation of seismic data involves basic evaluation by mapping some subsurface reflectors and faults. This gives a general indication of the tectonic regime in the area and serves as a basis for more detailed studies. Typically, the major steps involve the correlation of horizons, picking the faults, and generation of several types of maps for describing the general geology.

Horizon correlation involves interlinking a particular event from one seismic line to another, making sure the stratigraphic surface followed is the same over the area. This correlation is based on the reflection character, which includes amplitude, phase (peak, through or zero-crossing), frequency, and waveform. Reflections that appear to be continuous and extensive, known as markers, are usually the first candidate to be mapped. Also, the horizons are first picked on dip lines where the lateral continuity of the events is more visible.

Faults are identified on seismic data as breaks in the continuity of seismic horizons. Depending on the scale of the faults and the quality of the seismic data, care has to be taken in the interpretation of subtle faults. A gradational grayscale is utilized to enhance the terminations of low amplitude events as suggested by Brown (2011). To further investigate the structural grains and fluid flow pathways of the reservoir interval of the study area, a seismic attribute called ant tracking is utilized. They are used for subsurface structural analysis to detect faults and fractures and their directional (lateral and vertical) connectivity.

3.3.2 *Well log conditioning*

Well to seismic tie

This a critical and important step in seismic interpretation. It is used to correlate the well logs to the 3D seismic lines which will enable comparison between the two data types. The density and sonic logs are combined into the acoustic impedance which is then used to calculate the reflectivity using the relationship in equation 20 & 21:

$$\boldsymbol{Z = \rho V} \qquad (20)$$

$$\boldsymbol{RC = \frac{Z_2 - Z_1}{Z_2 + Z_1}} \qquad (21)$$

where Z is the acoustic impedance, Z_1 is the acoustic impedance for layer 1, Z_2 is the acoustic impedance for layer 2 and RC is the reflectivity. This reflectivity is convolved with a zero-phase wavelet, which in this research, a Ricker wavelet is used. The result of this is a synthetic seismogram for the well. This synthetic seismogram is compared with the extracted seismic trace along the well path and the matching quality is accessed. The synthetic seismogram can be aligned to match the seismic trace either by shifting the synthetic seismogram up or down or by selecting specific zones to stretch or squeeze.

Upscaling of Well logs

Well log data are acquired at a higher resolution compared to the seismic data. This disparity in resolution brings about a major challenge when correlating the two datasets. Also, the high-frequency noise from the logs can obscure variations in the property at the seismic scale. One way to bridge this gap is to upscale the high-resolution data.

In this research, a running weighted average of the brittleness is estimated based on the pixel of the seismic data. The result of this step is a smooth brittleness log with a resolution comparable to that of the seismic data.

3.3.3 *Seismic Brittleness Prediction*

In the area of reservoir property prediction from seismic data, several techniques have been adopted. The most widely used being statistical inversion methods and multi-attribute supervised classification. While these methods have been successful in some cases, they are computationally expensive and typically produce results that are biased and have little to no relevance to the prediction of the reservoir property.

In this research, a novel technique proposed by Gao (2011) is utilized. This method entails directly extracting the waveform at the well location that is directly related to the reservoir property and then calculating the correlation between each of the seismic waveform and the model waveform extracted at the well location. This results in a 3D cube intrinsically related to the property that is sought to be predicted.

This method, referred to as the Texture Model Regression (TMR), is based on a dynamic model with variable amplitude, frequency, phase, and dimension. This is a data-driven, trial-and-error process that keeps fitting different model combinations until an optimum model is achieved that has a high correlation with the reservoir property. This algorithm uses a predefined texture model as a reference to evaluate the similarity of seismic texture/waveform at each data location relative to the model via linear least-squares regression analysis. The similarity is evaluated from the perspective that is defined by a specific model composed of n amplitude samples. The reference model ($\boldsymbol{M_i}$) is defined by a full wavelength of the trigonometric sine function with a variable phase using equation 22:

$$\boldsymbol{M_i} = \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{1} + \boldsymbol{A}\,\mathbf{sin}(\mathbf{2}\boldsymbol{\pi\omega i/n} + \emptyset)) \qquad (22)$$

where A is an amplitude scaling constant, ω is the frequency in hertz, n is the number of samples defining the size of the model, $\emptyset$ is the phase variable and *i* is a natural variable denoting the *i*th sample ranging from 1 to *n*.

The model $\boldsymbol{M_i}$ is initialized with an amplitude equal to that of the seismic data, the frequency same as the dominant frequency of the seismic data in the interval of interest and a full wavelength with a specific phase. At the well locations, the algorithm retrieves a segment of data-trace defined by n amplitude samples, which in this case is a 1D wavelet that is equivalent to 4n ms at a 4-ms sampling rate. Then, the algorithm performs least-squares linear regression using the two sets of amplitude samples, reference waveform samples $x_{i,}$ and data trace samples y_i, on the x-y crossplot.

Before applying this algorithm, a reliable time-depth conversion needs to exist and the top and base of the interval of interest be defined. Then, at all the well locations in the zone of interest, the algorithm iteratively updates the model by changing the parameters defined in equation 3 as necessary until an optimal match is realized for all the wells in the reservoir interval. At that point, the final model is used to create a thematic volume that is optimally calibrated to a specific reservoir property observed from all the wells.

Generally, the workflow consists of 5 steps.

I) Construction of a mini-volume model $\boldsymbol{M_i}(i = 1, 2, \dots, n)$ consisting of *n* amplitude samples.

II) Retrieval of mini-volume data (texture element) $\boldsymbol{D_i}(x, y, z)(i = 1, 2, \dots, n)$ consisting of *n* amplitude samples centered at a location *(x, y, z)* in 3D amplitude volume.

III) Linear least-squares regression between the mini-volume model $\boldsymbol{M_i}$ and the mini-volume data (texture element) $\boldsymbol{D_i}(x, y, z)$.

IV) Output the regression gradient at each location $g(x, y, z)$, correlation coefficient $r(x, y, z)$, and intercept $i(x, y, z)$.

V) Repeat steps 2 through 4 at all the sample locations until a high correlation with well data.

This algorithm calculates the regression gradient (g), correlation coefficient (r), and intercept (i) by performing linear-squares regression between the model and amplitude data using equations 23, 24, and 25:

$$\boldsymbol{g = \frac{\sum_{i=1}^{n}(M_i - \bar{M})(D_i - \bar{D})}{\sum_{i=1}^{n}\left((M_i - \bar{M})\right)^2}} \tag{23}$$

$$\boldsymbol{r = \frac{1}{n}\sum_{i=1}^{n}\left[D_i - \bar{D} + g\left((M_i - \bar{M})\right)^2\right]} \tag{24}$$

$$\boldsymbol{i = \bar{D} - g\bar{M}} \tag{25}$$

where $\boldsymbol{\bar{D}}$ and $\boldsymbol{\bar{M}}$ denote the mean values of the element $\boldsymbol{D_i}$ and $\boldsymbol{M_i}$, respectively, and n is the number of amplitude samples of the model.

3.4 RESULTS

3.4.1 *Well to seismic tie*

The synthetic seismogram generated from the forward modeling process of the sonic and density log shows a good match with the seismic data. This indicates that the well data are located accurately on the seismic section and can be confidently extrapolated outwards. Also, they provide a means of correctly identifying the horizons to pick in the seismic data (Figure 23).

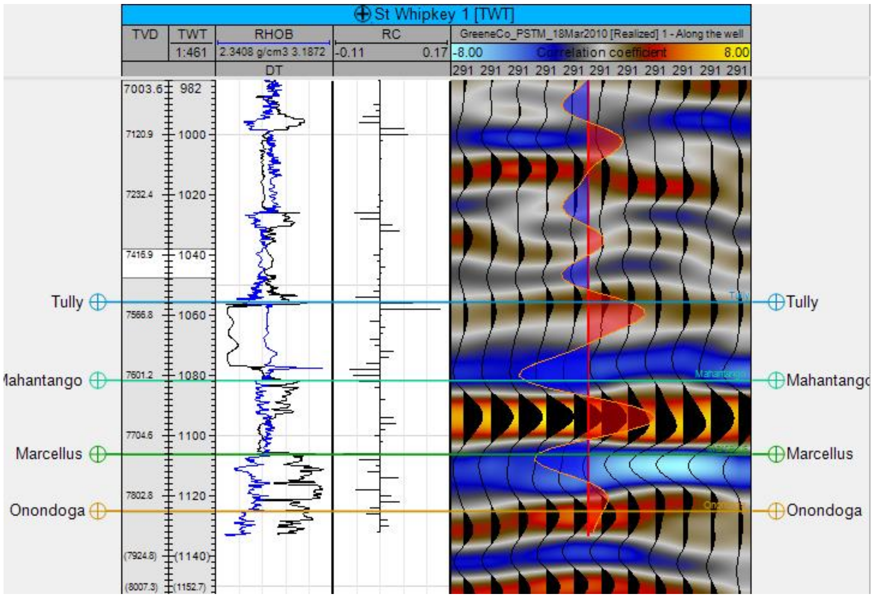

Figure 23. Seismic to well tie display with the well cross-referenced with inline 291.

3.4.2 *Log Upscaling*

Upscaling of the brittleness log reduced the frequency and also resulted in a smooth log that averaged the measurement for each pixel, considering a total of 21 pixels with the Marcellus shale at the center (Figure 24). This upscaled brittleness estimate is used in the seismic inversion of brittleness.

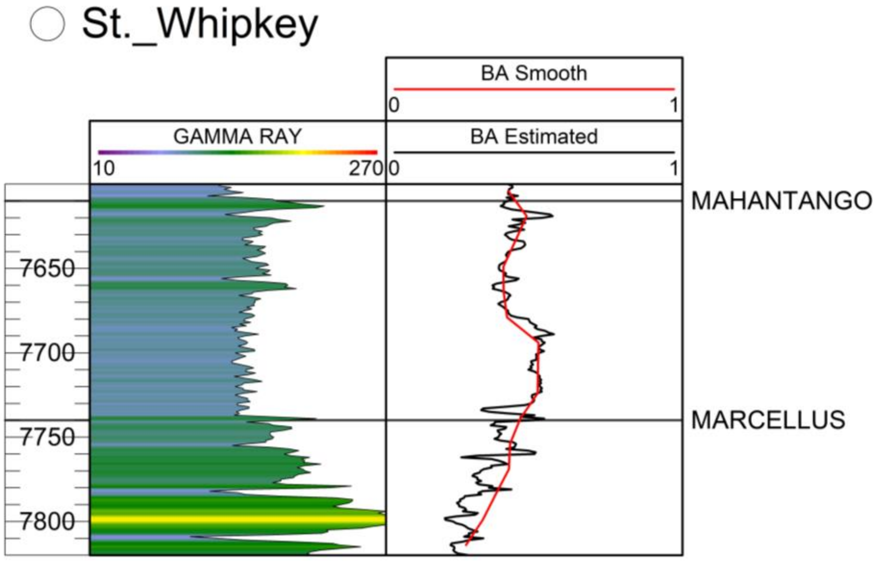

Figure 24. A plot of the actual and upscaled brittleness estimate.

3.4.3 *Surface maps*

Preliminary studies of the structure maps and the seismic cross-sections indicate that the folds in the Devonian section are dominantly east verging bounded reverse faults (Figure 25). A more detailed structural analysis was carried out by Donahoe (2011). Three major northeast-trending reverse faults were interpreted to cut through the Marcellus formation. However, several similar types of faults occur in the sequence below and above. This fault development and reactivation are linked to the Rome Trough.

The structural complexity of this area will adversely affect the prediction of brittleness from the 3D seismic data, as the algorithm relies on comparison on the adjacent seismic traces to establish a meaningful relationship, not forgetting the limited well control available in the area. As such, the initial result from this research will be of low confidence and more of a theoretical display of the algorithm.

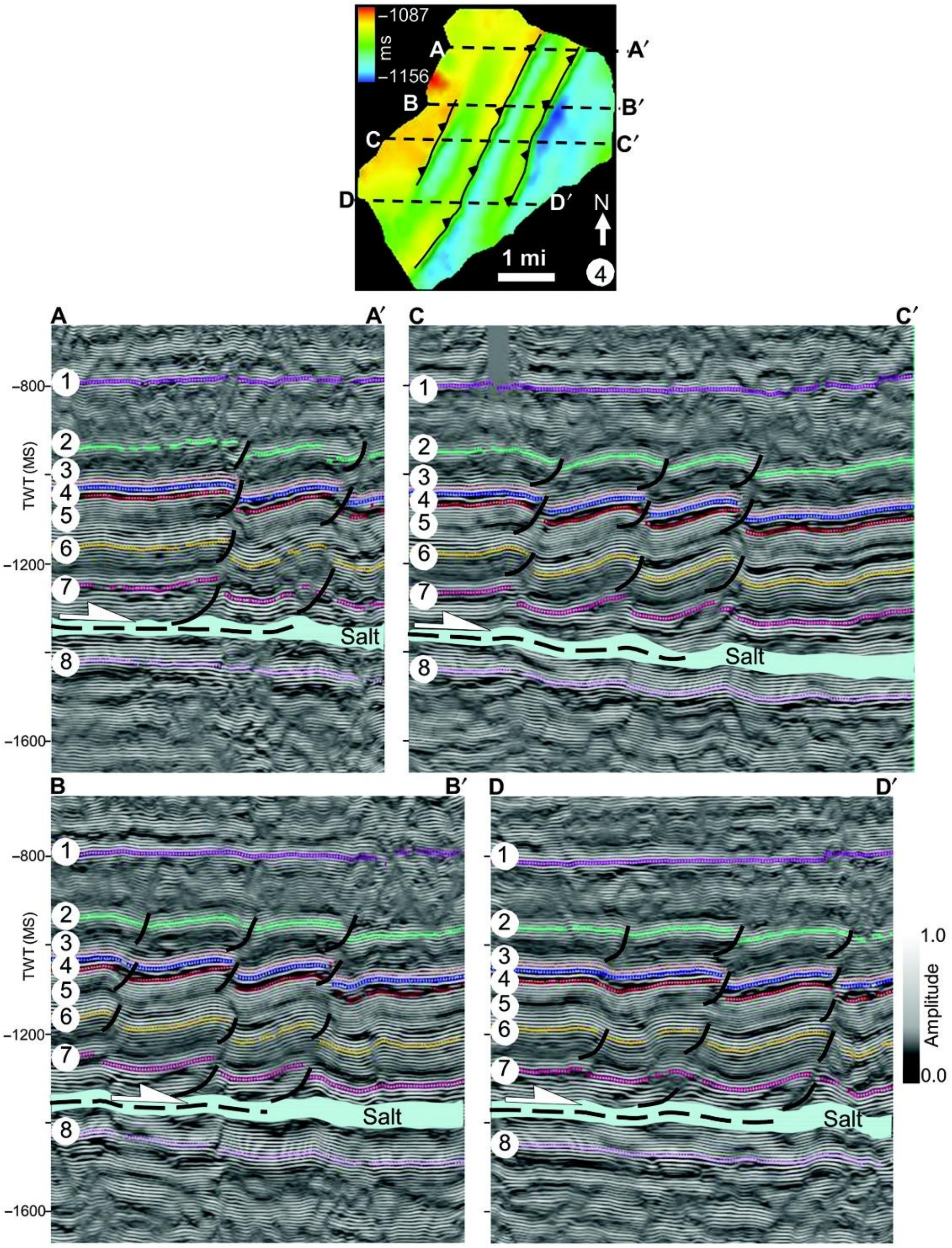

Figure 25. Sequential cross-sections of processed seismic volume with interpretations. The black lines are listric detachment faults. The Salina salt is highlighted as the primary basal detachment horizon. Color lines are eight mapped stratigraphic horizons of interest. (1) Elk, (2) Tully, (3) Mahantango, (4) Marcellus, (5) Onondaga, (6) Oriskany, (7) post-Salina, and (8) pre-Salina (Gao et al., 2018).

3.4.4 *Comparison of seismic amplitude with geophysical logs*

An investigation of how the seismic amplitude is related to the geomechanical estimations was carried out. The reference horizon shown in figures 26, 27, and 28 is the Marcellus shale. A closer look at these logs shows a distinct trend when compared to the seismic amplitudes. This is true because the seismic events are direct representations of the subsurface lithology which in turn has varying geomechanical properties.

The rationale behind using the 3D seismic data to predict geomechanical properties is more intuitive than physical. 3D seismic data have been successfully used in the estimation of properties such as porosity, fluid saturation, and lithology. This is possible because seismic amplitude which is a measure of acoustic contrast between layers is directly affected by these properties. Also, the geomechanical property of the layer plays a role in the variation of the seismic amplitude, thus making it suitable for the spatial prediction of this property.

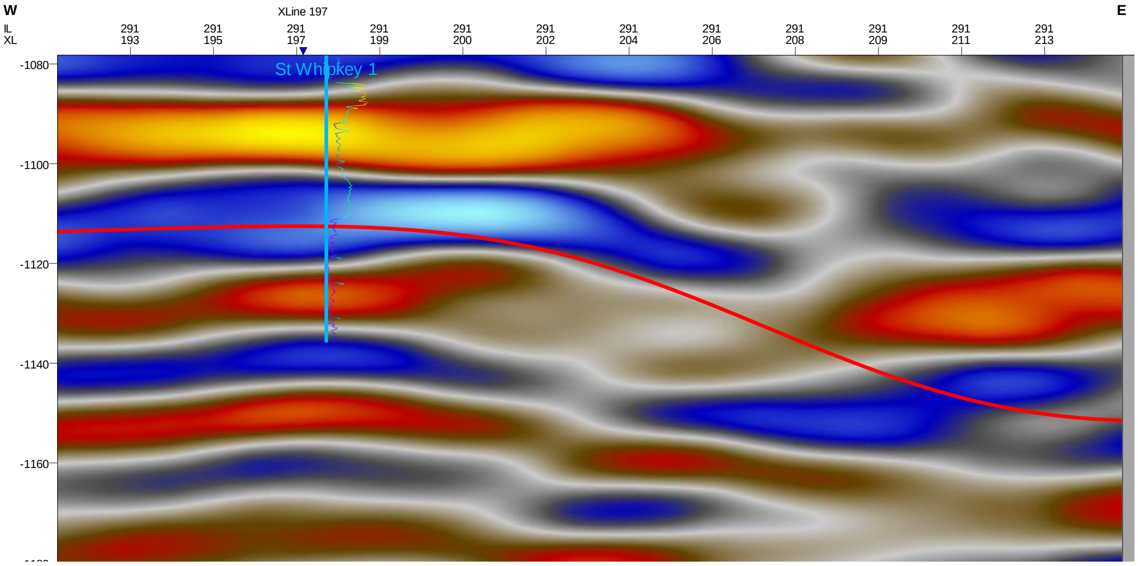

Figure 26. Young's Modulus estimated brittleness juxtapose on the seismic amplitude.

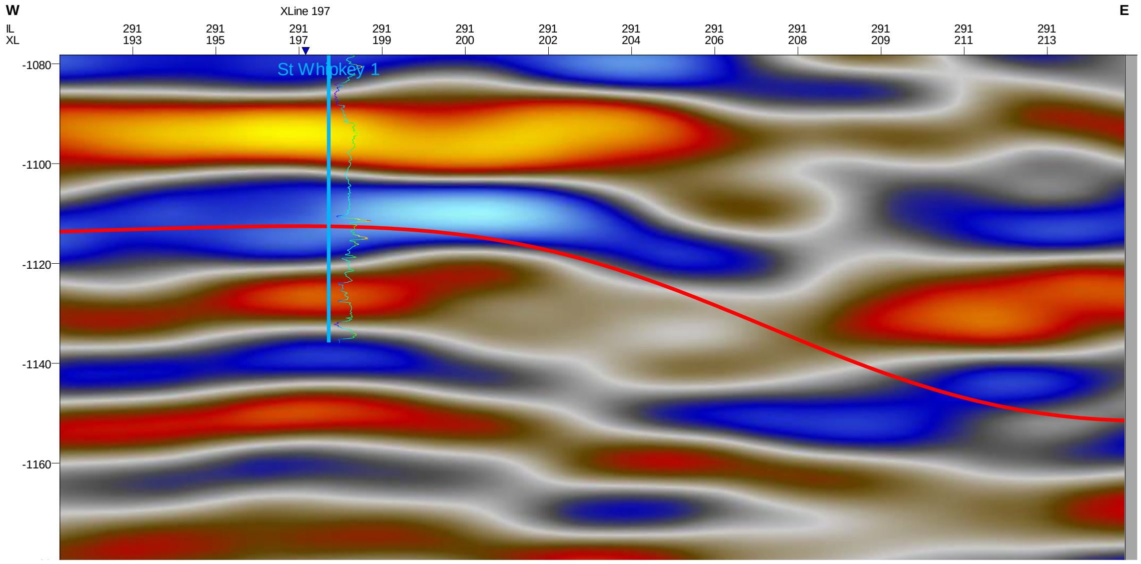

Figure 27. Poisson Ratio estimated brittleness juxtaposed with the seismic amplitude.

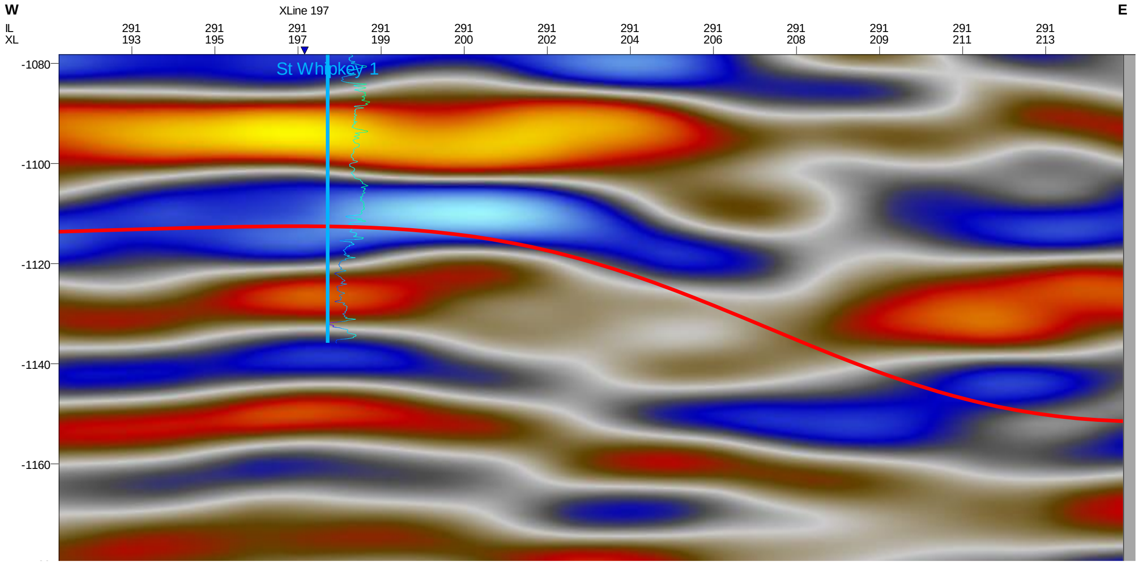

Figure 28. Estimated brittleness juxtaposed with the seismic amplitude.

3.4.5 *Spectral Texture Analysis*

The post stacked time migrated 3D seismic data contains numerous spectral information which is utilized in this research. As described in the methodology, the algorithm extracts the texture information for a variety of frequency and phase and matches it to the log information. The starting point of this is the dominant frequency of the seismic data which is 25 Hz as seen in figure 29.

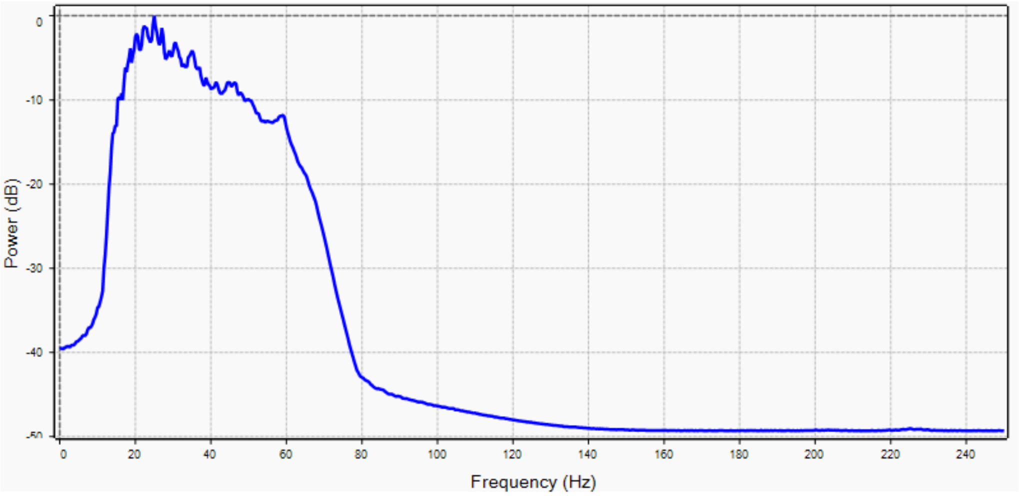

Figure 29. Spectral frequency analysis of the seismic data.

3.4.6 *Inversion result*

Implementation of the techniques described in previous sections to invert brittleness from seismic texture yielded a positive result (Figures 30 & 31). These results are preliminary, as the verification has not been carried out with the other wells in the seismic survey. Also, the inversion result will be used qualitatively as the seismic response produced is related to the correlation with the brittleness of the rocks and not an actual measure of brittleness.

The result of the data-driven seismic inversion is, therefore, qualitatively compared with the seismic amplitude and ant tracking (Figure 32). This is to evaluate the suitability of this approach before carrying out a more constrained inversion with the well control in the area, which will increase the accuracy and confidence of the result.

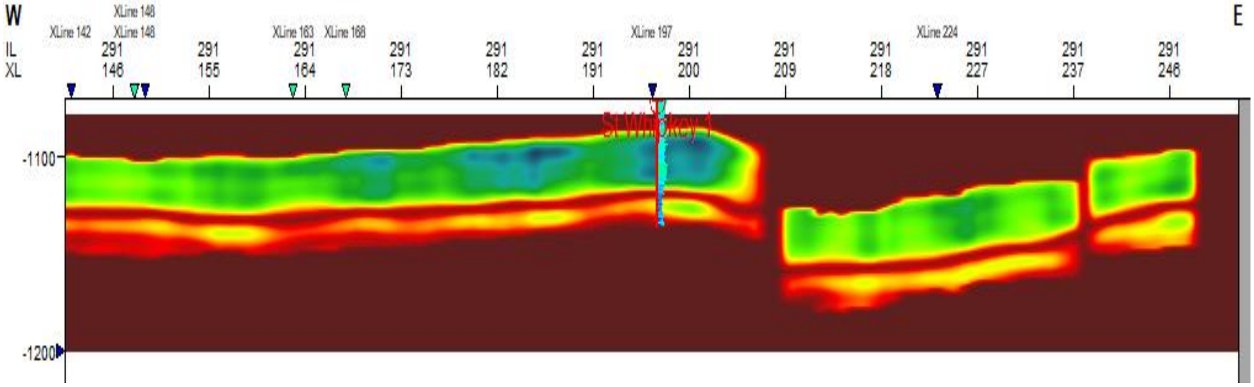

Figure 30. A cross-section of inline 291 showing the seismic brittleness inversion result.

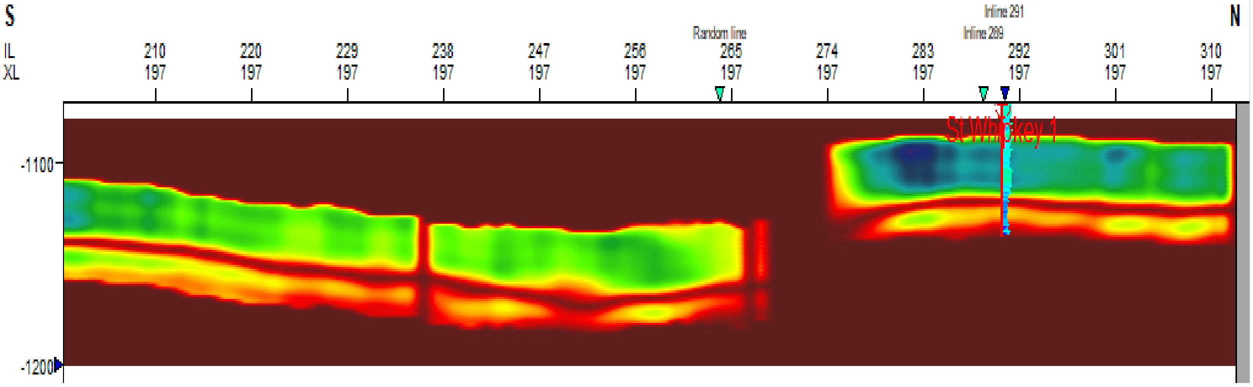

Figure 31. A cross-section of crossline 197 showing the seismic brittleness inversion result.

The first observation of the algorithm is how the presence of large scale faults affects the results. This adversely affects the result as the inversion depends on the seismic signal texture. Therefore, the application is limited to areas with limited faulting with adequate well control. A possible solution to this problem is to calibrate the seismic on both sides of the faults. This will increase the confidence level of the algorithm, and therefore perform better in regions with major faults.

Figure 32b shows the ant tracking image for a time slice at 1114ms close to the Marcellus shale horizon. This shows the fault and fracture network which is used as a qualitative measure of brittleness. Brittle regions will exhibit a higher degree of faulting and fracturing and vice versa. Comparison of figures 32b and 32c, the black ellipse shows a region with low brittleness while the red square shows a region with high brittleness.

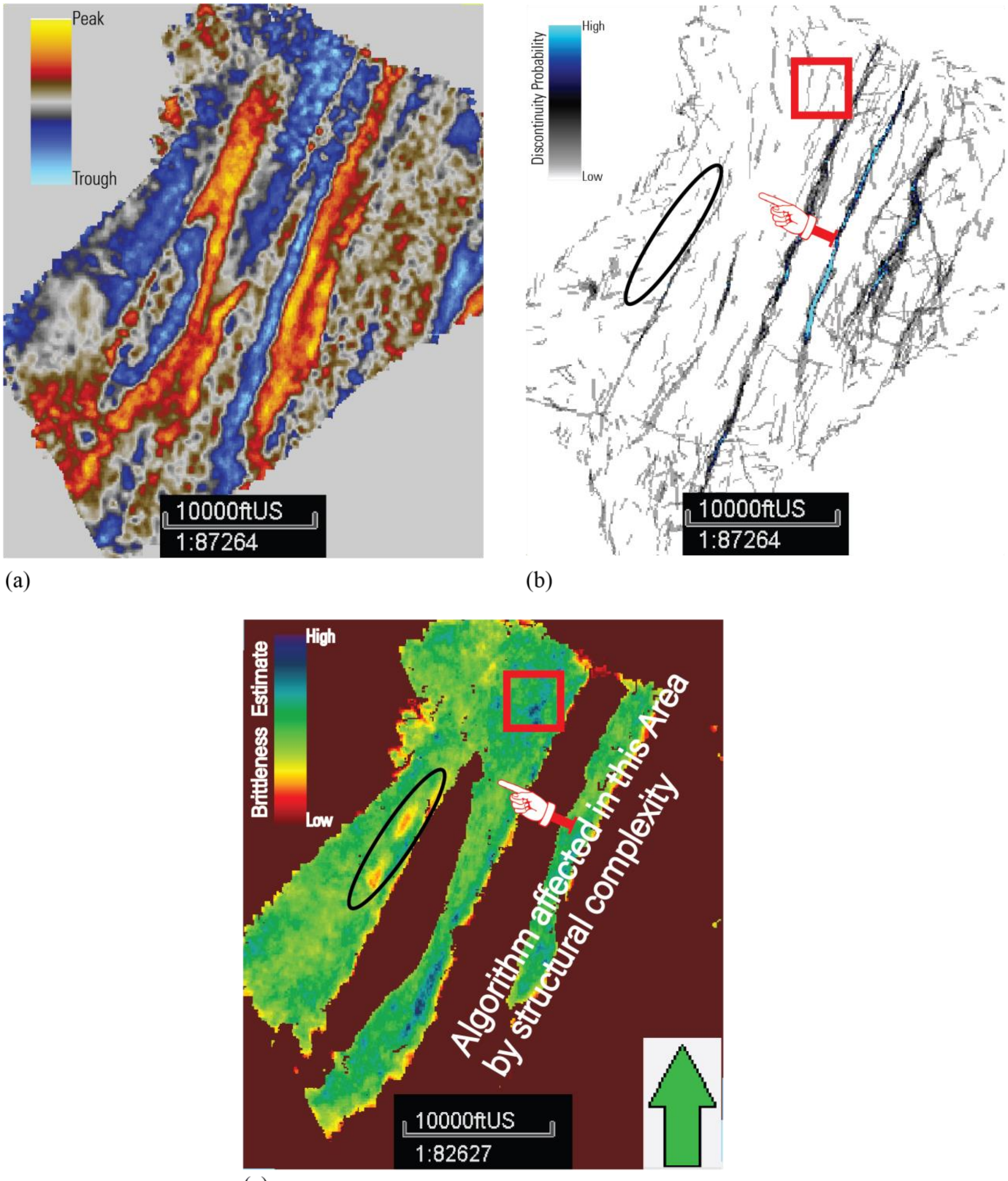

Figure 32. A time slice at 1114 ms of the (a) seismic amplitude, (b) ant tracking, and (c) seismic brittleness inversion.

3.5 CONCLUSION

It is important to integrate 3D seismic data with geophysical logs in the characterization of unconventional reservoirs. 3D seismic data was used in the characterization of brittleness, which is an important reservoir property for sweet spot identification in unconventional reservoirs. A novel approach (Gao, 2011) that utilizes the spectral component of the seismic traces provides a seismic texture analysis tool for a machine learning-based, data-driven approach to reservoir brittleness characterization from seismic texture analysis (Gao et. al., 2020).

Preliminary results from the texture-based data-driven inversion process show a promising future for using the seismic texture as an indicator for brittleness and for using texture spectral analysis algorithm to invert brittleness spatially. More wells are required to laterally sample and calibrate the seismic responses to the log responses. The algorithm offers a workaround to the paucity of wells with relevant geophysical logs by vertically sampling and calibrating the seismic response to the log responses, making it possible to invert brittleness even with one well only (Gao et al., 2020). Such a workaround can apply to horizontal wells given its dense lateral sampling along the reservoir. The workflow used to execute the prediction of brittleness from the seismic data can be extended to predict other reservoir properties, such as porosity, that has a physical or sensible relationship with seismic amplitude responses.

3.6 FUTURE WORK

To properly demonstrate the application of the algorithm, it will be applied to seismic data that have more well control. This will reduce the bias in the machine learning process, and produce a more robust, well-trained inversion result. Also, it is suggested that the brittleness estimated using the elastic properties be calibrated by using cores in the laboratory.

CHAPTER 4 – REFERENCES AND APPENDIX

4.1 REFERENCES

Altindag, R., 2003, Correlation of specific energy with rock brittleness concepts on rock cutting. The Journal of the South African Institute of Mining and Metallurgy. April 2003, p 163ff.

Altindag, R., and A. Guney, 2010. Predicting the relationships between brittleness and mechanical properties (UCS, TS and SH) of rocks: Scientific Research and Essays, 5, 2107–2118.

Avary, K. L., and Lewis, J. E., 2008. New interest in cores taken thirty years ago: the Devonian Marcellus Shale in northern West Virginia, AAPG: Eastern Section Meeting. Poster.

Blakey, R., 2010. Paleogeography. http://jan.ucc.nau.edu/~rcb7/index.html

Brown, A.R., 2011. Interpretation of three-dimensional seismic data. Society of Exploration Geophysicists and American Association of Petroleum Geologists.

Brownlee, J., 2016. A gentle introduction to the gradient boosting algorithm for machine learning. Machine Learning Mastery. Nov, 9.

Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

Chopra, S. and Sharma, R.K., 2015. Determining brittleness from seismic data. Search and Discovery Article, 41725.

Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), pp.273-297.

Davis, D., and S. J. Reynolds, 1996, Structural geology of rocks and regions, 2nd ed.: Wiley.

Donahoe, T.B. 2011, 3D Seismic, Attribute-Assisted, Structural Interpretation for Hydrocarbon Exploration and Production: Southwest Pennsylvania, Central Appalachian Basin, West Virginia University.

Donahoe, T., and D. Gao, 2016. Application of 3D seismic attribute analysis to structure interpretation and hydrocarbon exploration southwest Pennsylvania, Central Appalachian Basin: A case study: Interpretation, 4, no. 3, T291–T302.

EIA, 2017. Updates to the Marcellus Shale Play Maps. United States Department of Energy, Energy Information Administration

Engelder, T., 2008. Structural geology of the Marcellus and other Devonian gas shales: geological conundrums involving joints, layer-parallel shortening strain, and the contemporary tectonic stress field. Pittsburgh Association of Petroleum Geologists Field Trip (Sept. 12–13, 2008), 96 p

Ettensohn, F. R., 1985. The Catskill delta complex and the Acadian orogeny: A model: GSA Special Papers 201, 39–50.

Gale, J.F., Reed, R.M. and Holder, J., 2007. Natural fractures in the Barnett Shale and their importance for hydraulic fracture treatments. AAPG bulletin, 91(4), pp.603-622.

Gao, D., 2004. Texture model regression for effective feature discrimination: Application to seismic facies visualization and interpretation. Geophysics, 69(4), pp.958-967.

Gao, D., 2008. Application of seismic texture model regression to seismic facies characterization and interpretation. The Leading Edge, 27(3), pp.394-397.

Gao, D., 2011. Latest developments in seismic texture analysis for subsurface structure, facies, and reservoir characterization: A review: Geophysics, 76, no. 2, W1–W13

Gao, D., Donahoe, T., Duan, T. and Sullivan, P., 2018. Acadian hinterland-vergent detachment structures in southwestern Appalachian Plateau: Implications for Marcellus Shale gas exploration and production. Interpretation, 6(4), pp.SN85-SN99.

Gao, D., Ore, T. and Zulkapeli, N. I., 2020, Reservoir brittleness prediction using seismic texture analysis: Application to the Marcellus Shale in the central Appalachian Basin: AAPG Annual Conference, Houston, TX.

Goodway, B. Perez, M., Varsek, J., and Abaco, C., 2010. Seismic petrophysics and isotropic-anisotropic AVO methods for unconventional gas exploration, The Leading Edge, 1500-1508.

Graupe, D., 2007. Principles of Artificial Neural Networks. 2nd ed. Hackensack: World Scientific Publishing Co. Pte. Ltd.

Grieser, W.V. and Bray, J.M., 2007. Identification of production potential in unconventional reservoirs. In Production and Operations Symposium. Society of Petroleum Engineers.

Handin, J., and R. V. Hager, 1957, Experimental deformation of sedimentary rocks under confining pressure: Tests at room temperature on dry samples: AAPG Bulletin, 41, 1–50.

Handin, J., and R. V. Hager, 1958, Experimental deformation of sedimentary rocks under confining pressure: Tests at high temperature on dry samples: AAPG Bulletin, 42, 2892–2934.

Handin, J., R. V. Hager, M. Friedman, and J. N. Feather, 1963, Experimental deformation of sedimentary rocks under confining pressure: Pore pressure tests: AAPG Bulletin, 47, 717–755.

Hastie, T., Tibshirani, R. and Friedman, J., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hermana, M., Ghosh, D.P. and Sum, C.W., 2019. Elastic-Based Brittleness Estimation from Seismic Inversion. In Exploitation of Unconventional Oil and Gas Resources-Hydraulic Fracturing and Other Recovery and Assessment Techniques. IntechOpen.

Jarvie, D. M., Hill, R. J., Ruble, T. E., and Pollastro, R. M., 2007. Unconventional shale-gas systems: the Mississippian Barnett Shale of North-Central Texas as one model for thermogenic shale-gas assessment: AAPG Bulletin, v. 91, pp. 475 – 499.

Kulander, C.S. and Ryder, R.T., 2005. Regional seismic lines across the Rome Trough and Allegheny Plateau of northern West Virginia, western Maryland, and southwestern Pennsylvania (p. 9). US Geological Survey.

Lash, G. G., and T. Engelder, 2011. Thickness trends and sequence stratigraphy of the Middle Devonian Marcellus Formation, Appalachian Basin: Implications for Acadian foreland basin evolution: AAPG Bulletin, 95, 61–103.

Love, P. L. and M. Simaan, 1984. Segmentation of stacked seismic data by the classification of image texture: 54th Annual International Meeting, SEG, Expanded Abstracts, 480–482.

McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys., v. 5, pp. 115-133.(Рус. In Автоматы: Сб.-М., 1956. С. 363-384.).

Milici, R.C. and Swezey, C., 2006. Assessment of appalachian basin oil and gas resources: Devonian shale-middle and upper paleozoic total petroleum system (p. 54). US Department of the Interior, US Geological Survey.

Minsky, M.P. and Papert, S.A., 1969. Perceptron: an introduction to computational geometry, Cambridge.

Negara, A., Ali, S.S., Al Dhamen, A., Kesserwan, H. and Jin, G., 2017. Data-Driven Brittleness Index Prediction from Elemental Spectroscopy and Petrophysical Properties Using Support-Vector Regression. In SPWLA 58th Annual Logging Symposium. Society of Petrophysicists and Well-Log Analysts.

Park, H., D. L. Barbeau, Jr., A. Rickenbaker, D. Bachmann-Krug, and G. Gehrels, 2010. Application of foreland basin detrital-zircon geochronology to the reconstruction of the southern and central Appalachian orogeny: The Journal of Geology, 118, 23–44.

Perez, R. and Marfurt, K., 2013, January. Brittleness estimation from seismic measurements in unconventional reservoirs: Application to the Barnett Shale. In 2013 SEG Annual Meeting. Society of Exploration Geophysicists.

Perez, R. and Marfurt, K., 2014. Mineralogy-based brittleness prediction from surface seismic data: Application to the Barnett Shale. Interpretation, 2(4), pp.T1-T17.

Rickman, R., Mullen, M.J., Petre, J.E., Grieser, W.V. and Kundert, D., 2008, January. A practical use of shale petrophysics for stimulation design optimization: All shale plays are not clones of the Barnett Shale. In SPE annual technical conference and exhibition. Society of Petroleum Engineers.

Roen, J.B., 1984. Geology of the Devonian black shales of the Appalachian Basin. Organic Geochemistry, 5(4), pp.241-254.

Rolon, L., Mohaghegh, S.D., Ameri, S., Gaskari, R. and McDaniel, B., 2009. Using artificial neural networks to generate synthetic well logs. Journal of Natural Gas Science and Engineering, 1(4-5), pp.118-133.

Rosenblatt, F., 1957. The perceptron: A probabilistic model for Visual Perception. In Proceedings of the 15th International Congress of Psychology, North Holland (pp. 290-297).

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

Shumaker, R. C., and T. H. Wilson, 1996. Basement structure of the Appalachian foreland in West Virginia: Its style and affect on sedimentation, in B. A. van der Pluijm and P. A. Catacosinos, eds., Basement and basins of eastern North America: Geological Society of America Special Paper 308, 141–155.

Sumi, L., 2008. Shale gas: Focus on the Marcellus shale, Report, The Oil and gas accountability project/Earthworks, Durango, Colo.

Tan, M., Song, X., Yang, X. and Wu, Q., 2015. Support-vector-regression machine technology for total organic carbon content prediction from wireline logs in organic shale: A comparative study. Journal of Natural Gas Science and Engineering, 26, pp.792-802.

Wang, F.P., and Gale, J. F. W., 2009. Screening criteria for shale-gas systems: Gulf Coast Association of Geological Societies Transactions, v. 59, pp. 779-793.

Wells, F., 2004. A new method to help identify unconventional targets for exploration and development through integrative analysis of clastic rock properties: Houston Geological Society Bulletin, 47, 34–49.

Woodrow, D.L., 1985. Paleogeography, paleoclimate, and sedimentary processes of the Late Devonian Catskill Delta: in Woodrow, D.L. and Sevon, W.D. eds., The Catskill Delta: Geological Society of America Special Paper 201, .P· 51-64.

Wrightstone, G., 2009. Marcellus Shale–Geologic Controls on Production. In AAPG Annual Convention, Denver, CO, Jun 7-10.

Zagorski, W. A., G. R. Wrightstone, and D. C. Bowman, 2012. The Appalachian Basin Marcellus gas play: Its history of development, geologic controls on production, and future potential as a world-class reservoir, in J. A. Breyer, ed., Shale reservoirs — Giant resources for the 21st century: AAPG Memoir 97, 172–200.

Zhang, B., Zhao, T., Jin, X. and Marfurt, K.J., 2015, Brittleness evaluation of resource plays by integrating petrophysical and seismic data analysis. Interpretation, 3(2), pp.T81-T92.

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

4.2 APPENDIX

Python Code

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar  1 12:43:15 2020
@author: tmo0005
"""
import numpy as np    #numpy package
import pandas as pd   #panda package
import matplotlib.pyplot as plt   #package to make plots
import plotly.graph_objects as go
import pickle
from sklearn import metrics
from sklearn.svm import SVR
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import GridSearchCV, train_test_split
from xgboost import XGBRegressor
import seaborn as sns
from sklearn.neural_network import MLPRegressor
#%% #load the training and testing data used to build the model
directory = r'C:\Users\Tobi\Desktop\Thesis work\Wells'  #the path of the directory containing the log data in csv format
```

```
df1 = pd.read_csv('{}\{}.csv'.format(directory,"whipkey_edited"))
df2 = pd.read_csv('{}\{}.csv'.format(directory,"MIP3H_edited"))
df3 = pd.read_csv('{}\{}.csv'.format(directory,"boggess_edited"))
data = df1.append([df2,df3])
data.dropna(inplace=True)

#%% A general summary of the data
summary = data.describe()
summary.to_csv('{}\{}.csv'.format(directory,"basic_statistics"))   #save the summary

#%% Feature Selection
data_corr = data.loc[:,['GR','RHOB','NPHI','PEFZ','DT','BA']]  #select the input features
correlation = data_corr.corr(method ='pearson')
correlation1 = data_corr.corr(method ='spearman')

fig, ax = plt.subplots(nrows=1, ncols=2,figsize=(10,4))
sns.heatmap(correlation, vmin=-1, vmax=1, cmap='RdGy', annot=True, ax = ax[0])
ax[0].set_title('Pearson Correlation')

sns.heatmap(correlation1, vmin=-1, vmax=1, cmap='RdGy', annot=True, ax = ax[1])
ax[1].set_title('Spearman Rank Correlation')

fig.savefig('{}\{}.png'.format(directory,"feature_correlation"), dpi=400) #save the image
```

```
#%% Matrix Scatter Plot

fig = go.Figure(data=go.Splom(

                dimensions=[dict(label='Gamma Ray',

                                 values=data['GR']),

                            dict(label='Density',

                                 values=data['RHOB']),

                            dict(label='Neutron',

                                 values=data['NPHI']),

                            dict(label='PE factor',

                                 values=data['PEFZ']),

                            dict(label='Sonic',

                                 values=data['DT'])],

                showupperhalf=False,  # remove plots on diagonal,

                diagonal_visible=False,

                text=data['BA'],

                marker=dict(

                    opacity=0.4,

                    color=data['BA'],

                    showscale=True,

                    colorscale='Inferno',

                    colorbar=dict(

                        title='BA'

                    ),
        ),
        ))

fig.update_layout(
  title='Matrix Scatter Plot',
  width=950,
  height=900,
)

fig.write_image('{}\{}.jpeg'.format(directory,"matrix_scatter_plot"))
#%% #splitting the data into training and test set
scaler = MinMaxScaler()

X = data.loc[:,['GR','RHOB','NPHI']]
y = data.loc[:,['BA']]

X_norm =  scaler.fit_transform(X)
y_norm = scaler.fit_transform(y)

X_train, X_test, y_train, y_test = train_test_split(X_norm,y_norm, test_size=0.3, random_state =
42)

#%%
```

```
data_norm = scaler.fit_transform(data)
summary = pd.DataFrame(data_norm).describe()

summary.to_csv('{}\{}.csv'.format(directory,"basic_statistics_after_norm"))


#%% training the SVR
#hyper_parameters =  {'C': np.logspace(-3,10, num=13),
#                       'gamma': np.logspace(-9,3, num=13)}
hyper_parameters =  {'C': [10,100],
                      'gamma': [ 1,10,100]}

gs = GridSearchCV(SVR(), param_grid=hyper_parameters,
                   cv=5, scoring='r2', verbose=1, n_jobs=-1)

gs.fit(X_train,y_train)

#%%
y_train_pred = gs.predict(X_train)
R2_train = metrics.r2_score(y_train,y_train_pred)

y_blind_pred = gs.predict(X_test)
R2_blind = metrics.r2_score(y_test,y_blind_pred)
```

```
y_total_pred = gs.predict(X_norm)

R2_total = metrics.r2_score(y_norm,y_total_pred)


gs.best_score_

#%%

y_pred = scaler.inverse_transform(y_total_pred.reshape(-1, 1))

data['BA_svr']=y_pred

#%% saving the model

filename = '{}\{}.sav'.format(directory,"svr_scenario2")

pickle.dump(gs, open(filename, 'wb'))


#%%

fig, ax = plt.subplots(figsize=(10,8))

fig.suptitle("Crossplot of predicted brittleness and estimated brittleness", fontsize=16)

fig.subplots_adjust(top=0.95)

ax.scatter(y_norm, y_total_pred)

fig.savefig('{}\{}.png'.format(directory,"svr_scenario2_cross_plot"), dpi=400)


#%% training the XGB

hyper_parameters =  {'n_estimators' : [100, 300, 500],

                    'min_child_weight': [6],

                    'max_depth' : [5, 8, 15]

                    }
```

```
gs = GridSearchCV(XGBRegressor(), param_grid=hyper_parameters,
                  cv=3, scoring='r2', verbose=1, n_jobs=-1)

gs.fit(X_train,y_train)
#%%
y_train_pred = gs.predict(X_train)
R2_train = metrics.r2_score(y_train,y_train_pred)

y_blind_pred = gs.predict(X_test)
R2_blind = metrics.r2_score(y_test,y_blind_pred)

y_total_pred = gs.predict(X_norm)
R2_total = metrics.r2_score(y_norm,y_total_pred)

#gs.best_score_
#%%
y_pred = scaler.inverse_transform(y_total_pred.reshape(-1, 1))
data['BA_xgb']=y_pred

#%%
fig, ax = plt.subplots(figsize=(10,8))
fig.suptitle("Crossplot of predicted brittleness and estimated brittleness", fontsize=16)
fig.subplots_adjust(top=0.95)
ax.scatter(y_norm, y_total_pred)
fig.savefig('{}\{}.png'.format(directory,"xgb_scenario2_cross_plot"), dpi=400)


#%% saving the model
filename = '{}\{}.dat'.format(directory,"xgb_scenario2")
pickle.dump(gs, open(filename, 'wb'))


#%% feature importance
fig, ax = plt.subplots()
ax.bar(['GR','RHOB','NPHI'], (gs.feature_importances_)*100, color='black')
ax.set_ylabel('Percentage (%)')
ax.set_title('Feature Importance', fontsize=16)
ax.grid(True,alpha=0.5)
fig.savefig('{}\{}.png'.format(directory,"xgboost_feature_importance_scenario2"), dpi=400)


#%% training the Neural Network
hyper_parameters =  {'hidden_layer_sizes': [(10,),(15,),(20,),(25,)],
                  'tol': [0.0001,0.00001,0.001]}
gs = GridSearchCV(MLPRegressor(max_iter=1200,solver = 'lbfgs'),
param_grid=hyper_parameters,
               cv=3, scoring='r2', verbose=1, n_jobs=-1)
```

```
gs.fit(X_train,y_train)

#%%
fig, ax = plt.subplots(figsize=(10,8))
fig.suptitle("Crossplot of predicted brittleness and estimated brittleness", fontsize=16)
fig.subplots_adjust(top=0.95)
ax.scatter(y_norm, y_total_pred)
fig.savefig('{}\{}.png'.format(directory,"ann_scenario2_cross_plot"), dpi=400)

#%% saving the model
filename = '{}\{}.dat'.format(directory,"ann_scenario2")
pickle.dump(gs, open(filename, 'wb'))

#%%
y_train_pred = gs.predict(X_train)
R2_train = metrics.r2_score(y_train,y_train_pred)

y_blind_pred = gs.predict(X_test)
R2_blind = metrics.r2_score(y_test,y_blind_pred)

y_total_pred = gs.predict(X_norm)
R2_total = metrics.r2_score(y_norm,y_total_pred)
```

```
gs.best_score_
#%%
y_pred = scaler.inverse_transform(y_total_pred.reshape(-1, 1))
data['BA_ann']=y_pred
#%%
data.to_csv('{}\{}.csv'.format(directory,"data_with_predicted_values_scenario2"))

#%%
#%% load the models to deploy it
filename = '{}\{}.dat'.format(directory,"svr_scenario2")
svr_model = pickle.load(open(filename,'rb'))

filename = '{}\{}.dat'.format(directory,"xgb_scenario2")
xgb_model = pickle.load(open(filename,'rb'))

filename = '{}\{}.dat'.format(directory,"ann_scenario2")
ann_model = pickle.load(open(filename,'rb'))

#%% load all the wells in the other wells
df1 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis
work\Wells\{}.csv'.format("whipkey_edited"))
df2 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis work\Wells\{}.csv'.format("MIP3H_edited"))
df3 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis work\Wells\{}.csv'.format("boggess_edited"))
```

```
df4 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis work\Thesis\Wells\{}.csv'.format("37-059-
24212"))
df5 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis work\Thesis\Wells\{}.csv'.format("375924716
- Cree"))
df6 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis work\Thesis\Wells\{}.csv'.format("Corry"))
df7 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis work\Thesis\Wells\{}.csv'.format("Mohr"))
df8 = pd.read_csv(r'C:\Users\Tobi\Desktop\Thesis
work\Thesis\Wells\{}.csv'.format("V_Virgili"))

#%% select the traget zone in the wells
df4 = df4[df4['DEPT']>=6000].loc[:,['DEPT','GRGC','DEN','NPHI']]
df5 = df5[df5['DEPT']>=6000].loc[:,['DEPT','GRGC','DEN','NPHI']]
df6 = df6[df6['DEPT']>=6000].loc[:,['DEPT','GRGC','DEN','NPHI']]
df7 = df7[df7['DEPT']>=6000].loc[:,['DEPT','GRGC','DEN','NPHI']]
df8 = df8[df8['DEPT']>=6000].loc[:,['DEPT','GR','RHOB','NPHI']]
#%%
df4.rename(columns={'GRGC':'GR','DEN':'RHOB'}, inplace=True)
df5.rename(columns={'GRGC':'GR','DEN':'RHOB'}, inplace=True)
df6.rename(columns={'GRGC':'GR','DEN':'RHOB'}, inplace=True)
df7.rename(columns={'GRGC':'GR','DEN':'RHOB'}, inplace=True)

df1.dropna(inplace=True)
df2.dropna(inplace=True)
```

```
df3.dropna(inplace=True)
df4.dropna(inplace=True)
df5.dropna(inplace=True)
df6.dropna(inplace=True)
df7.dropna(inplace=True)
df8.dropna(inplace=True)
#%% deploy models
scaler = MinMaxScaler()

X1 = df1.loc[:,['GR','RHOB','NPHI']]
X1_norm =  scaler.fit_transform(X1)

X2 = df2.loc[:,['GR','RHOB','NPHI']]
X2_norm =  scaler.fit_transform(X2)

X3 = df3.loc[:,['GR','RHOB','NPHI']]
X3_norm =  scaler.fit_transform(X3)

X4 = df4.loc[:,['GR','RHOB','NPHI']]
X4_norm =  scaler.fit_transform(X4)

X5 = df5.loc[:,['GR','RHOB','NPHI']]
X5_norm =  scaler.fit_transform(X5)
```

```
X6 = df6.loc[:,['GR','RHOB','NPHI']]
X6_norm =  scaler.fit_transform(X6)

X7 = df7.loc[:,['GR','RHOB','NPHI']]
X7_norm =  scaler.fit_transform(X7)

X8 = df8.loc[:,['GR','RHOB','NPHI']]
X8_norm =  scaler.fit_transform(X8)
#%%
y1_svr = svr_model.predict(X1_norm)
y2_svr = svr_model.predict(X2_norm)
y3_svr = svr_model.predict(X3_norm)
y4_svr = svr_model.predict(X4_norm)
y5_svr = svr_model.predict(X5_norm)
y6_svr = svr_model.predict(X6_norm)
y7_svr = svr_model.predict(X7_norm)
y8_svr = svr_model.predict(X8_norm)

y1_xgb = xgb_model.predict(X1_norm)
y2_xgb = xgb_model.predict(X2_norm)
y3_xgb = xgb_model.predict(X3_norm)
y4_xgb = xgb_model.predict(X4_norm)
```

```
y5_xgb = xgb_model.predict(X5_norm)
y6_xgb = xgb_model.predict(X6_norm)
y7_xgb = xgb_model.predict(X7_norm)
y8_xgb = xgb_model.predict(X8_norm)

y1_ann = ann_model.predict(X1_norm)
y2_ann = ann_model.predict(X2_norm)
y3_ann = ann_model.predict(X3_norm)
y4_ann = ann_model.predict(X4_norm)
y5_ann = ann_model.predict(X5_norm)
y6_ann = ann_model.predict(X6_norm)
y7_ann = ann_model.predict(X7_norm)
y8_ann = ann_model.predict(X8_norm)
#%%
df1['BA_svr']=y1_svr
df1['BA_xgb']=y1_xgb
df1['BA_ann']=y1_ann

df2['BA_svr']=y2_svr
df2['BA_xgb']=y2_xgb
df2['BA_ann']=y2_ann

df3['BA_svr']=y3_svr
```

```
df3['BA_xgb']=y3_xgb
df3['BA_ann']=y3_ann

df4['BA_svr']=y4_svr
df4['BA_xgb']=y4_xgb
df4['BA_ann']=y4_ann

df5['BA_svr']=y5_svr
df5['BA_xgb']=y5_xgb
df5['BA_ann']=y5_ann

df6['BA_svr']=y6_svr
df6['BA_xgb']=y6_xgb
df6['BA_ann']=y6_ann

df7['BA_svr']=y7_svr
df7['BA_xgb']=y7_xgb
df7['BA_ann']=y7_ann

df8['BA_svr']=y8_svr
df8['BA_xgb']=y8_xgb
df8['BA_ann']=y8_ann
#%% save the result
```

```
df1.to_csv('{}\{}.csv'.format(directory,"whipkey_predicted"))
df2.to_csv('{}\{}.csv'.format(directory,"MIP3H_predicted"))
df3.to_csv('{}\{}.csv'.format(directory,"boggess_predicted"))
df4.to_csv('{}\{}.csv'.format(directory,"37-059-24212_predicted"))
df5.to_csv('{}\{}.csv'.format(directory,"375924716 - Cree_predicted"))
df6.to_csv('{}\{}.csv'.format(directory,"Corry_predicted"))
df7.to_csv('{}\{}.csv'.format(directory,"Mohr_predicted"))
df8.to_csv('{}\{}.csv'.format(directory,"V_Virgili_predicted"))
```