USE OF IMAGE-BASED MACHINE LEARNING AND QUANTUM LEARNING FOR SUBSURFACE CHARACTERIZATION

A Thesis

by

MATTEO CAPONI

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Siddharth Misra
Committee Members,	Hadi Nasrabadi
	David W. Bapst
Head of Department,	Jeff Spath

December 2022

Major Subject: Petroleum Engineering

ABSTRACT

The proposed thesis aims to explore novel applications of machine learning for subsurface characterization. In the first chapter, an image-based data-driven workflow is proposed to characterize oil viscosity from side-wall rock sample images. Informative features are extracted from the rock sample images deploying several image-based filters and statistical models. Both regression and classification tasks are performed on the preprocessed data. The proposed workflow shows promising results for viscosity classification whereas future work is needed to improve the regression performance. The second and third chapters explore the application of quantum-enhanced machine learning models for lithology classification and the resulting comparison with classical machine learning models. The second chapter compares a quantum support vector machine with a traditional support vector classifier for lithology classification from well log data. Different sample sizes are tested to understand if a quantum advantage is obtained when the available data is limited. The third chapter investigates the application of both quantum support vector and variational quantum classifier for binary lithology classification. The score distribution obtained from testing the models with multiple iterations gives more insight on the current performance capabilities of quantum-enhanced machine models when compared to artificial networks. Overall, although a quantum advantage is not observed in both chapters, this work opens the door to future applications of quantum-enhanced machine learning for subsurface characterization.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Misra, and my committee members, Dr. Nasrabadi, and Dr. Bapst, for their guidance and support throughout the course of this research. In addition, I would like to acknowledge Adam Cox and the entire team at Berry Petroleum for their guidance and responsiveness in addition to my research teammates Rui Liu and Yusuf Falola for improving my knowledge of data science. Thanks also go to my classmates, the department faculty, and staff for making my time at Texas A&M University a great experience. Finally, thanks to my mother and father for providing me constant encouragement and the tools to achieve my dreams.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Professor Dr. Siddharth Misra and Dr. Hadi Nasrabadi of the Department of Petroleum Engineering and Professor Dr. David W. Bapst of the Department of Geology & Geophysics.

Funding Sources

Graduate study was supported by a fellowship and research assistantship from the DICE research group led by Dr. Misra and the Department of Petroleum Engineering at Texas A&M University. The contents and results of this work are solely the responsibility of the student and his advisory committee, and do not necessarily represent the official views of the funding sources cited above.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

IMAGE PREPROCESSING AND SUPERVISED LEARNING FOR VISCOSITY PREDICTION IN HEAVY OIL RESERVOIRS

This chapter illustrates a hybrid approach of image preprocessing and supervised machine learning for viscosity prediction on rock samples. In the image preprocessing segment, the application of multiple image-based filters and pixel intensities provides the extraction of meaningful feature for supervised learning model deployment. With the resulting features, supervised models are trained and evaluated for both regression and classification of target fluid viscosity.

Fundamental Questions

- Can color emitted from rock sample images provide information on the fluid viscosity that they contain?
- Do textural features extracted from RGB rock sample images provide useful information for fluid viscosity prediction of the stored hydrocarbon?
- With the advancements in machine learning, can data-driven methods provide an alternative from laboratory measurements for fluid property prediction in subsurface reservoirs?

Novelty and Scientific Impact

- Sato, LBP, and Multi-Otsu filters can describe the textural features of rock samples under white-light and UV-light.

- White-light images show greater association of target viscosity with pixel intensities as compared to image-based filters. UV-images have a strongest association with image-based filters.
- The color emitted from white-light and UV-light images have a direct relationship with oil viscosity based on the chemical structure of the hydrocarbon present in the rock samples. Textural information observed in white -light and UV-light images is related to fluid viscosity based on changes in rock matrix and organic matter.
- Histogram-based parametrization can translate the information from rock images to numerical features.
- Application of statistical-based methods can drastically reduce the feature space dimension while retaining information.
- The proposed image-based supervised learning model can accurately classify oil viscosity based on the rock sample image under white-light and UV-light.

Introduction

Literature Review

Table 1 and Table 2 show a literature review of the work being done on viscosity estimation and property estimation through image analysis respectively. As described in Table 1, the current literature is limited in regards of using data-driven methods for viscosity prediction. Al-Amoudi explores the use of artificial neural networks for viscosity regression using PVT information with positive results (Al-Amoudi et al, 2019). However, the sue of image-base machine learning workflows has not been explored yet. As shown in Table 2, the use of image preprocessing paired with machine learning techniques for subsurface characterization is widely

present in the industry. Wu and Misra show a successful application of organic-rich shales characterization using scanning electron microscopy images (Wu et al, 2019). Lastly, Table 3 represents examples of the research led by Dr. Misra in subsurface characterization using data-driven workflows. The results from these studies confirm the ability of machine learning techniques to accurately characterize the subsurface.

Table 1: Literature review of viscosity prediction methods in the Oil and Gas industry. There is no current model that predicts fluid viscosity based on image analysis. The use of rock images has not been explored yet for viscosity prediction.

Authors	Method	Input	Hydrocarbon Considered	Conclusions
Amir et al, 2018	Modified Mehrotra, Svrcek's, and Ghaderi's equation	Pressure and temperature	Super lightweight completion fluid	It is statistically possible to predict the variation of fluid's density and viscosity over the wide range of pressure and temperature
Al-Amoudi et al, 2019	Artificial Neural Network	PVT data	Yemeni crude	Neural network shows greater accuracy than conventional prediction through PVT data
Ahmed et al, 2014	NMR relaxation	NMR T_2	Kuwait Crudes	T_2-Viscosity correlation shows better accuracy than traditional correlations in a Kuwait Oil Field
De la Porte et al, 2009	Mole-average power law	Experimental oil viscosity	U.S light crude	Fitting the power law on experimental data is problematic and extrapolation in temperature is highly unreliable
Vesovic, 2007	Vesovic-Wakeham	Experimental gas viscosity	Natural Gas	The VW method can predict viscosity of natural gas within +- 2% in the temperature region of 260-400 K
Deumah et al, 2021	KNN, Random Forest, Decision Tree	N_2, CO_2, H_2S	Yemeni Gas	Decision Tree model predicts gas viscosity with a 0.042% MAPE

Table 2: Literature review of image-based models for physical properties prediction. The papers are focused on images derived from CT scans, X-Rays, and logs. There is no current model that looks at traditional RGB images.

Author	Method	Goal	Results
Wu et al, 2019	Random forest classifier	Locating organic matter and pores in shales	Validation dataset achieves an overall F1 score higher than 0.9
Diaz et al, 2020	Multi-energy radiographic image analysis	Unconventional rock typing	Carbonates and shales were properly discriminated with and average predictive error ranging from 1% to 14%
De Figueiredo et al, 2019	Convolutional neural network on CT scanned images	Estimate distribution of porosity and P-wave velocity	The approach is computationally efficient to estimate rock properties
Jobe et al, 2018	Convolutional neural networks on optical microscopy thin-section images	Classify lithofacies, reservoir zone, porosity, and permeability	Key uncertainties still exist
Cook et al, 2016	Correlation of volumetric attributes from seismic data with image logs	Calibrate seismic fracture prediction	Highly fractured diagenetically altered chert results in anomalously low envelope and low density.
Hossain et al, 2011	Numerical simulation and image analysis	Estimate relative permeability	Results are consisted with lab-based measurements

Table 3: Examples of work performed on subsurface characterization using data-driven models. The positive results confirm the validity of machine learning techniques for subsurface applications.

Author	Method	Goal	Results
Li et al, 2019	Long short -term and convolutional neural networks	Generating NMR T2 distributions	The neural networks models are robust to noise of the conventional input logs and achieve a R2 score of 0.75
He et al, 2019	Multiple machine learning models	In-situ sonic log synthesis in shale reservoirs	Artificial neural networks show the greatest performance with a R2 score of 0.87 and 0.85
He et al, 2019	Artificial neural networks	Generate synthetic dielectric dispersion logs in organic-rich shales	The best performance is achieved in generating permittivity dispersion logs with a 0.089 root-mean squared error
Liu et al, 2018	Discrete wavelet transform, neural networks	Visualize mechanical discontinuity	A double layered neural network achieves the best performance with an accuracy of 0.95
Foster et al, 2021	K-nearest neighbor, support vector machine, and logistic regression	Detection of high-water cut wells in Delaware basin	The three classifiers achieve a high median Matthew's correlation score of 0.9
Hossain et al, 2011	Deep reinforcement learning	Control mixed-mode fatigue crack growth	The reinforcement learning framework can successfully control the fatigue crack growth in a material despite its complexity

Background

Viscosity plays a crucial role in the development of heavy oil reservoirs around the globe. High viscosity impairs the flow of oil from the reservoir to the surface and therefore, limiting the field's production potential. Several enhanced oil recovery (EOR) methods such as CO2 and steam injection have been studied and applied to heavy oil reservoirs increase reservoir temperature with the intent of lowering fluid viscosity. EOR methods can be an extreme economic burden for operators due to the vast number of resources and manpower required. As a result, identifying

subsurface fluid viscosity becomes very useful to predict potential expenses associated with EOR methods. Side-wall core samples are taken to the lab for direct viscosity measurements under constant temperature. From the lab, the rock fluid is extracted and purified to remove any residual solids and water. There are three categories of viscometers widely used in the industry: capillary viscometer, electromagnetic viscometer, and rheometer. These methods work on different operating principles dictated by the desired temperature and pressure of the reservoir. Accuracy and repeatability of heavy oil viscosity measurements are not only affected by sample handling, storage, and cleaning procedures, but they are also affected by the selection of viscometers and the experimental procedures followed by different operators (Zhao et al, 2016). In addition to accuracy and repeatability concerns, lab viscosity measurement provides an additional economic burden due to transportation, handling and processing associated with viscometer measurements. This paper aims to bypass lab measurements by predicting hydrocarbon viscosity utilizing side-wall core images and machine learning models. This novel approach could severely reduce time and costs for heavy oil operators while potentially provide more accurate viscosity readings when compared to lab measurements.

Machine Learning Models

Elastic Net

Elastic Net performs an ordinary least squares method to identify a linear combination of features that best predict the target based on alpha and l1_ratio. Alpha controls regularization. The larger the alpha, the less features the model considers. A small value in alpha results in the model using all the original features. L1_ratio determines the ratio between l1 and l2 norm penalty. An Elastic Net model with only l1 penalty represents a Lasso regression whereas a module with only l2 penalty represents a Ridge regression.

K Nearest Neighbors

K Nearest neighbors (KNN) performs a feature similarity analysis to predict the target values of new datapoints. The number of neighbors represents the groups present in the dataset needed to perform a prediction on a new sample. Possibility of overfitting increases as the number of neighbors gets closer to 0. On the contrary, a large number of neighbors has the potential to underfit the data. P value dictates the formulation used to determine the distance between the new sample and neighboring samples. A p value of 1 represents Manhattan distance whereas a value of 2 represents Euclidian distance. Euclidian distance is effective in two and three-dimensional spaces. As p values get larger, the effect of large separations is enhanced compared to smaller separation. In higher dimensional spaces, p values over 1 tend to improve model accuracy.

Random Forest

Random forest creates a group of decision trees trained on different portions of the dataset. The trees are trained using the bootstrap aggregation in which the training is perform in parallel between different data samples. The result is then determined by the average of each decision tree. Random forest can be used for both regression and classification problems. The number of estimators represents the number of trees in the model. A small number of trees is conducive to overfitting. Criterion represents the decision function deployed on the feature space. In Random Forest, criterion dictates the feature threshold required to perform the split on a random feature. This hyperparameter varies depending if the model is used for regression or classification. Next, max depth represents the complexity of each tree. The deeper the tree, the larger the number of splits performed on the features. Increasing tree depth leads to potential

overfitting and thus, a decrease in model testing performance. Lastly, max features dictate the largest number of features given to each tree. The larger this threshold is, the more information each tree possesses. A small threshold can cause underfitting since the model trees are only exposed to a small subset of all the features. On the contrary, a large threshold causes information to be similar between each tree, causing the model to overfit.

Gradient Boosting

Gradient Boosting predicts target values by building a sequence of trees which improves during each step. Similar to Random Forest, Gradient Boosting is a tree-based model. However, while Random Forest build trees in parallel, Gradient Boosting builds a sequence of trees in a vertical structure also known as boosting. Due to the high model complexity, hyperparameter tuning is essential to ensure that the model doesn't overfit the training data. Learning rate controls how fast the model learns with each sequential tree. Lower learning rate improves model performance by reducing overfitting. However, this comes at the cost of increasing computational time. Like Random Forest, the number of estimators represents the number of trees in the model. Since the trees are not placed in parallel, a larger number of trees causes overfitting. The loss refers to the loss function that the model uses to compare the predicted with the actual values. Huber loss is chosen as the loss function. Huber loss combines both MAE and MSE. While MAE weighs larger errors the same as smaller errors, MSE focuses on larger outlier errors. A combination of the two given by the Huber loss provides a suitable balance. Like Random Forest, max depth and max features control the threshold for the maximum depth and the maximum number of features within a tree respectively.

Performance Metrics

Regression

- Mean absolute error (MAE) represents the average of the absolute differences between actual and predicted value. For MAE, the error increases as the value gets larger. MAE has a value of 0 if actual and predicted value are the same. MAE is never lower than 0. MAE weights large errors coming from outlier predictions the same as lower errors. This can lead to misinterpretations of model performance when solely looking at MAE.
- Mean absolute percentage error (MAPE) calculates the absolute difference of actual and predicted divided by the actual value. By scaling the error to a relative percentage, MAPE gives a more detail representation of the machine learning model performance through a wide range of values. The biggest limitation of this metric is his tendency to infinity when the actual values are close to zero.

Classification

- F1 Score calculates the harmonic mean of recall and precision which can be derived in terms of confusion matrix parameters.

$$\mathbf{F1\ Score} = \frac{\mathbf{TP}}{\mathbf{TP} + \frac{\mathbf{1}}{\mathbf{2}} * (\mathbf{FP} + \mathbf{FN})} \qquad \textbf{Equation 1}$$

Where:

- TP: true positives, both actual and prediction belong to the same class which can be either 0 or 1.
- FN: false negative, the actual belongs to class 1, but the predicted class is 0.
- FP: false positive, the actual belongs to class 0, but the predicted class is 1.

The metric ranges from 0 to 1. A value of 0 represents a total incapacity for an accurate prediction whereas a value of 1 represents a perfect prediction. By itself, the F1 score does not account for sample balance throughout the classes. F1 weighted score overcomes this limit by calculating a weighted average of each class F1 score based on the frequency of each class in the dataset.

- Matthew's Correlation Score represents a correlation coefficient between the actual and predicted classes. MCC can only be used for binary classification and can be expressed in terms of a confusion matrix parameters.

$$\mathbf{MCC} = \frac{\mathbf{TP} * \mathbf{TN} - \mathbf{FP} * \mathbf{FN}}{\sqrt{(\mathbf{TP} + \mathbf{FP})(\mathbf{TP} + \mathbf{FN})(\mathbf{TN} + \mathbf{FP})(\mathbf{TN} + \mathbf{FN})}} \qquad \textbf{Equation 2}$$

The MCC score ranges from -1 to 1. A value of one represents a perfect classification whereas a value of -1 a totally opposite classification. A value of 0 shows complete randomness. Due to its mathematical computation, MCC provides a better score for datasets with unbalanced classes.

Data Gathering

The proposed model is catered to heavy oil reservoirs. In particular, the data utilized during this study comes from a heavy oil reservoir located in the western United States. The reservoir is characterized by high oil content with predictable steady decline. The composition of the target reservoir is primarily described by the two prevalent lithologies of sandstone and diatomite. Samples of side-wall core from various reservoir depths were extracted for viscosity lab testing. In the process, images of such samples were captured under white-light and UV-light before fluid extraction. The images provided contain 300X300 pixels in RGB format. Samples

underwent testing procedures to accurately measure oil viscosity at a constant temperature of 180 degF. Overall, 684 samples of core rock were collected. Each sample provides a white-light and UV-light image as well as a corresponding fluid viscosity measurement from the lab. From the data provided, fluid viscosity ranges from 5 cP to as high as 600 cP. The samples are categorized in 2 classes, the low viscosity class containing samples with viscosity < 50 cP, and the high viscosity class containing samples with viscosity > 300 cP. For both classes, randomized samples are collected and compared against each other to obtain a greater understating of the differences in color and texture. Figure 2 represents one of the comparisons obtained from high and low viscosity samples for both white-light and UV-light images. As shown in Figure 1, from the resulting comparison, there are clear differences in both color and texture of the rock samples between the low and high viscosity classes. When comparing samples under white-light, low viscosity samples show a darker color than high viscosity samples. In contrast, low viscosity rock samples under UV-light display a brighter golden color compared to the dark purple color of high viscosity samples. When focusing on the texture of the rock samples, the low viscosity group shows a much smoother surface when compared to the more granular texture of the high viscosity group displaying larger pores.

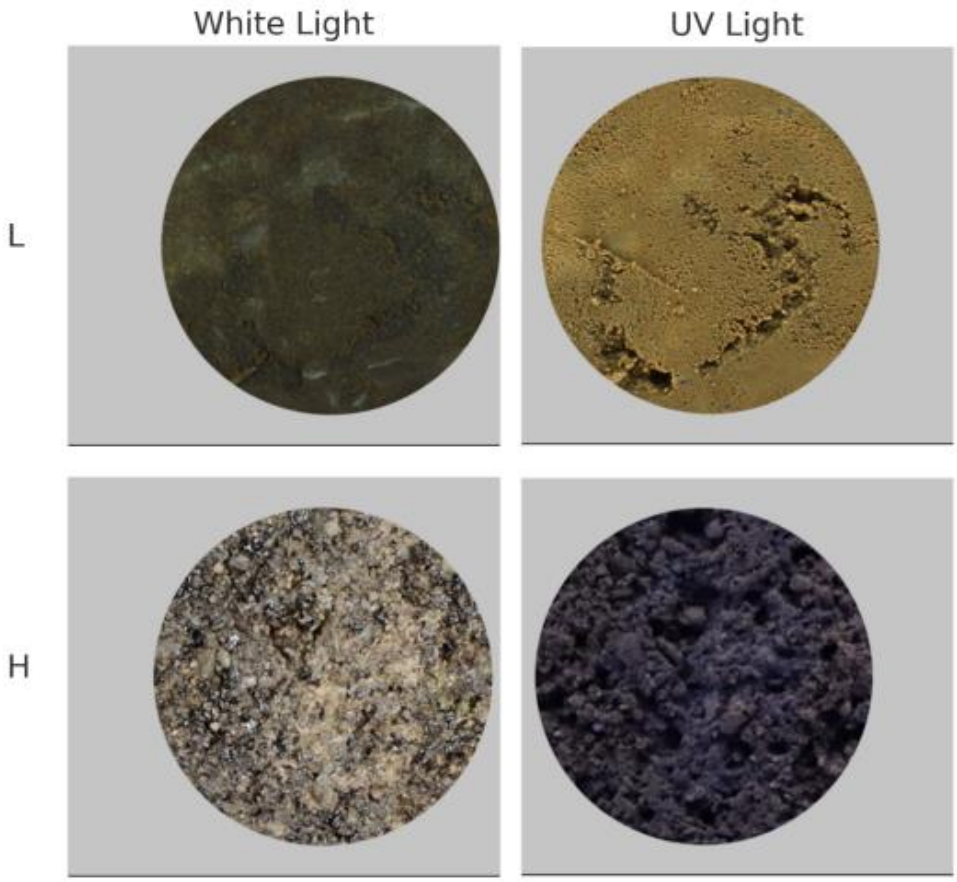

Figure 1: Visual comparison of rock samples containing low (<50 cP) fluid viscosity and high (>300 cP) fluid viscosity under UV-light and white-light. Significant changes in color and texture can be observed between the two viscosity classes.

Methodology

The data driven viscosity prediction model follows a clear workflow. The workflow implemented in the study is represented as a flowchart in Figure 2.

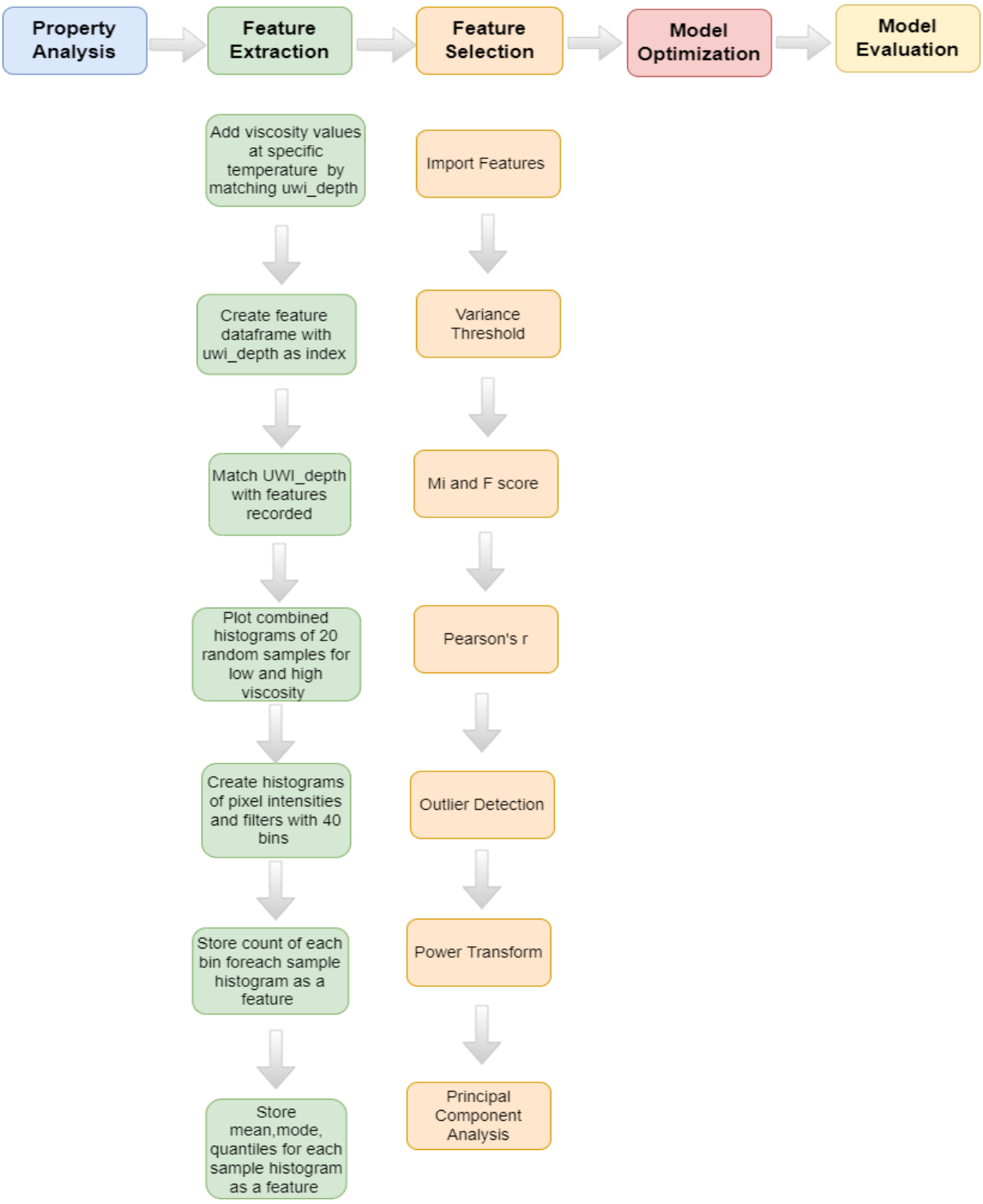

Figure 2: Flowchart of the workflow applied to create an image-based supervised learning model for viscosity prediction. The flowchart outlines the main steps taken as well as a more detailed explanation of feature extraction and feature selection.

The workflow is structured in 5 levels and begins with property analysis in which the raw image and viscosity data is visually analyzed and properly formatted for the next step. Feature extraction contains all the image preprocessing steps required to transform raw white-light and UV-images into meaningful numerical features by analyzing pixel intensities paired with image-based filters. In feature selection, further filtering is applied to the resulting histogram-based features to optimize performance and improve computational time. Several statistical-based models are applied to reduce the corresponding number of features. In model optimization, the model is initialized and trained on the respective dataset. Optimization algorithms are implemented to test thousands of possible combinations of model hyperparameters that give the most optimal performance metric. In model evaluation, the previously trained supervised model is tested to assess performance on both regression and classification tasks.

In the development of the classification model, four tasks are studied based on thresholds imposed on the target viscosity. Figure 3 shows a visual representation of the selected classes based on viscosity distribution on a natural logarithmic scale.

- High vs Low: Samples with extreme low viscosity are compared to samples with extreme high viscosity. Samples with viscosity lower than 57 cP are labeled low viscosity, whereas the samples labelled high have values higher than 330 cP.
- High vs Rest: Samples with extreme high viscosity are compared with the rest. the high viscosity category has samples higher than 330 cP while the rest category has samples lower than 122 cP.

- Low vs Rest: Samples with extreme low viscosity are compared with the rest. samples lower than 57 cP are labelled as low, whereas samples greater than 148 cP are labeled as the rest.
- End Points vs Middle Points: End members of the viscosity distribution are compared with the most frequent members. The end points are defined with viscosity less than 45 cP and greater than 403 cP, whereas middle points have viscosity ranging from 77 cP to 245 cP.

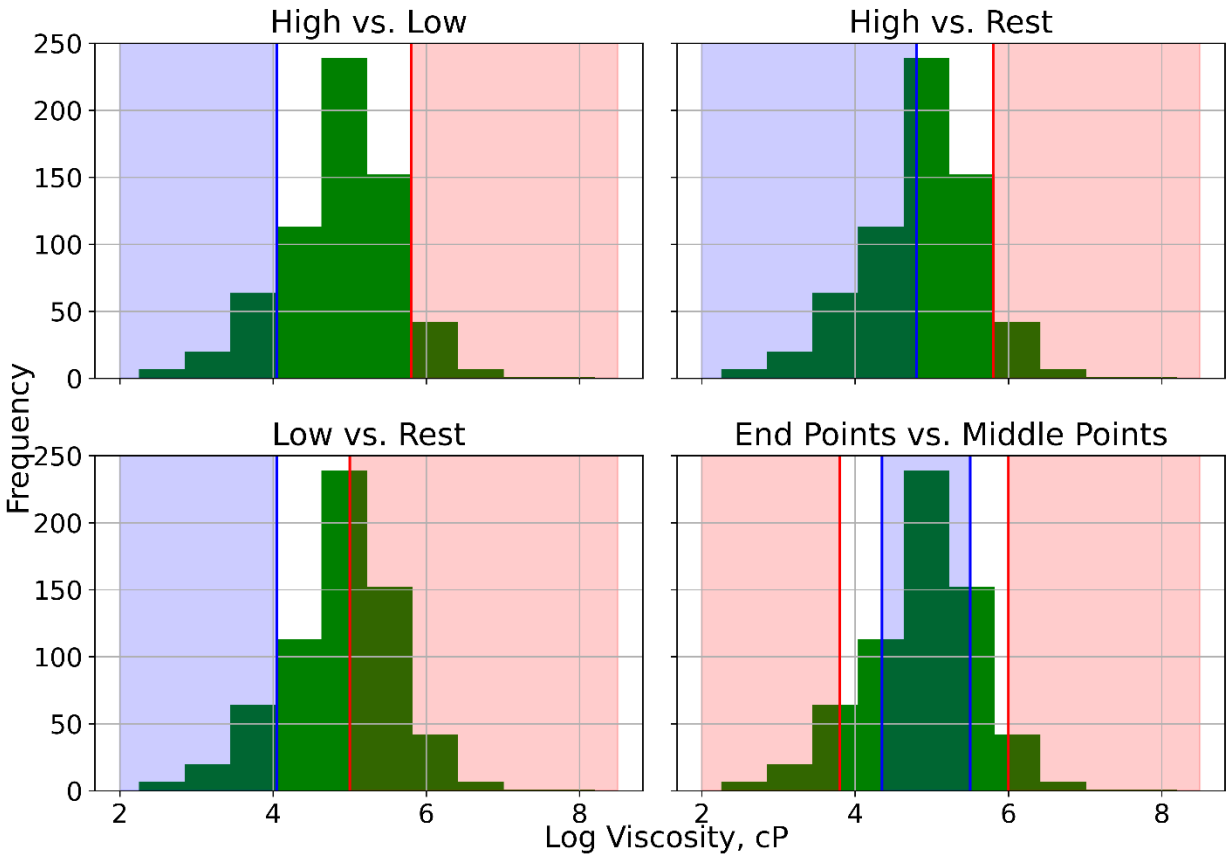

Figure 3: Thresholds applied to viscosity distribution on natural log scale to identify the binary classes of each classification task.

Lastly, the results are visualized to get a clear understanding of how the model is performing when compared to actual viscosity data.

Feature Extraction

Background Elimination

The white-light and UV-light rock sample images contains 300X300 pixels with a RGB format. As shown in Figure 1, the rock sample images are contained within a circular perimeter. Outside this perimeter, the background shows a grey color. After detail analysis, the background color is determined to be the same between all sample images in the provided dataset. Before applying image preprocessing techniques such as pixel intensities and image-based filters, the background is eliminated from the samples. This process ensures that the pixels contained within the background do not contaminate the feature extraction process. Since the background color is consistent across all samples, the portion of the 3D RGB matrix containing the specific background color is removed for each sample.

Pixel Intensities

The first image preprocessing technique deployed for feature extraction is pixel intensities. Both white-light and UV-light images follow an RGB format. Unlike 2D grayscale images, RGB images are represented as a 3D matrix with the third dimension representing the red, green, and blue channel respectively. The color of each pixel is therefore described as a one-dimensional array containing a combination of red, green, and blue intensities. The original image is broken down into three 2x2 matrices each representing a color channel. This process allows for the creation of numerical features that describe the differences of each pixel RGB intensities when compared to those of the adjacent pixels. Pixel intensities alone is not sufficient to produce meaningful features for accurately predict viscosity values through the use of a supervised learning model. Additional features extracted from textural information can increase

the machine learning model ability to distinguish between low and high viscosity samples and thus, improving the model accuracy.

Image-based filters

A total of 5 image-based filters were tested in both white-light and UV-light images. The goal of image-based filters is to accurately describe differences in texture between low and high viscosity samples.

- Sobel edge detector calculates the absolute non-directional gradient value at each pixel by applying a 3x3 edge-detection matrix. Sobel detects odd edges. This process allows the Sobel filter to derive both horizontal and vertical edges (Han et al., 2020).
- Sato is referred to as a tubeness filter. It is commonly used in the medical field to detect continuous ridges by calculating the eigenvectors of the Hessian matrix and computing the similarity of an image to tubes (Sato et al., 1998).
- Hessian filter describes the second order local image intensity variations. Hessian filter detects even edges. Based on the calculated eigenvalues, the filter can detect edge and corner regions (Rudzki, 2011). Hessian filters are good at detecting local structures, such as blobs.
- Local Binary Pattern (LBP) describes the local texture patterns of an image. LBP works by selecting a center pixel a neighborhood of pixels around a specified radius. The center pixel is used as a threshold for the neighboring pixels to quantify textural information (Sairamya et al., 2019).

- Multi-Otsu is a thresholding algorithm used to separate the pixels of an input image into multiple classes based on the grayscale pixel intensities.

Figure 4 shows a visual representation of these filters applied to a random low viscosity rock sample (<50 cP) and a random high viscosity rock sample (>300 cP). The filters highlight presence of channels and granularity of the samples.

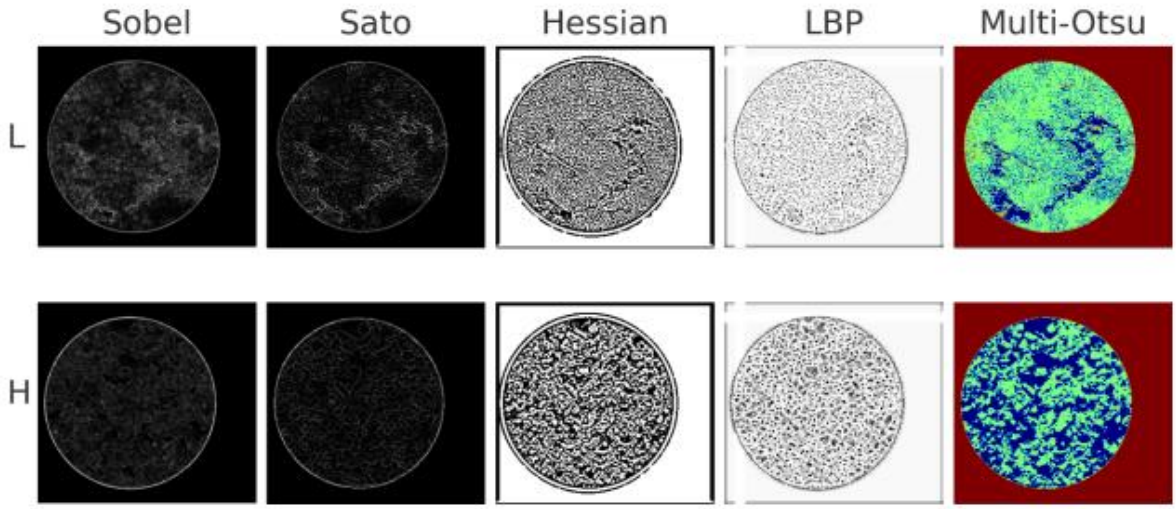

Figure 4: Visual representation of Sobel, Sato, Hessian, LBP. And Multi-Otsu image-based filters applied on white-light images in both low (L) and high (H) viscosity groups. The use of image-based filters provides more information on the texture of the rock samples which significantly differ between the low and high viscosity class.

The five filters as well as pixel intensities are evaluated on how impactfully describe changes in fluid viscosity. The comparison is achieved by analyzing the differences in the histograms between low and high viscosity samples. A histogram is generated for each sample image for the corresponding image filtering technique. To reduce bias, 10 random samples are selected from both the low and high viscosity class. For each technique, the 10 samples histograms are combined into one to better represent the changes in value distribution between high and low viscosity samples. Figure 5 compares the differences in histogram distribution

between rock image samples with oil viscosity lower than 50 cP and samples with viscosity higher than 300 cP. From this comparison, 3 image-based filters (Sato, LBP, and Multi-Otsu) paired with pixel intensities are identified to be the most informative in describing differences in viscosity between low and high viscosity samples. A 40-bin histogram is generated from each filtered/raw white-light and UV-light image. A total of 40 features per sample image are generated from the count of each bin. Additional features were extracted from the resulting histograms to add more information. Statistical parameters namely total count, minimum, maximum, mean, median, 25th percentile, 75th percentile, skewness, kurtosis, and mode are stored as feature for each filtered/raw image. In summary, there are a total of 50 features per sample image for each image-based filter and pixel intensity (red, green, blue). This amount to 300 features extracted from white-light and UV-light image of each rock sample totaling to 600 features per sample.

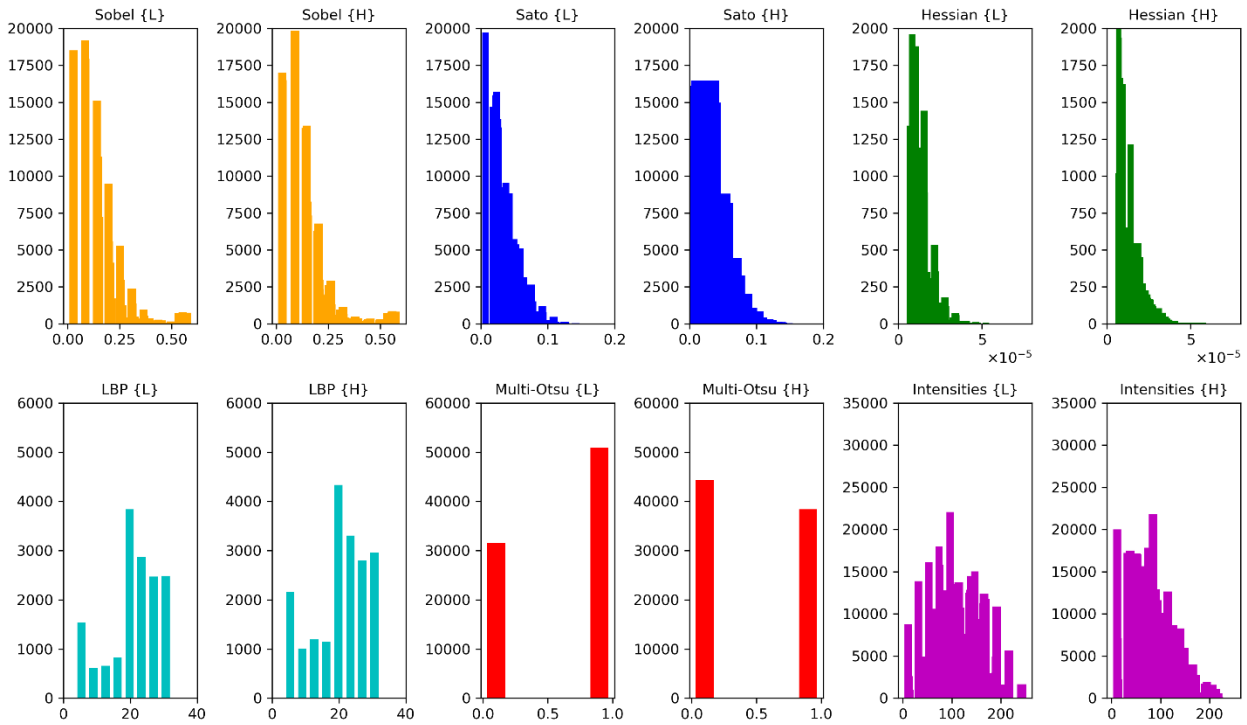

Figure 5 : Value distributions of the image pixels after filter application. The filters analyzed are Sobel (orange), Sato(blue), Hessian(green), LBP (cyan), Multi-Otsu (red), and the average pixel intensity between the three color channels (pink). For each technique, the combined histogram of 10 randomized low viscosity samples is compared to the one generated from the high viscosity class

Feature Selection

A total of 600 histogram-based features are generated for each rock sample from the applied image-based filters and pixel intensities. Due to this large number, feature selection represents a crucial step to assure optimal performance and computational time of the supervised learning model. In feature selection, multiple statistical measurements are applied to reduce the dimensionality of the feature set. For each statistical method, thresholds are applied based on the observed data to ensure the selection of the most informative feature. The outlined steps for feature selection are presented on the flowchart in Figure 2.

Robust Scaler

The first step required before the applications of feature selection method is to scale the features. The process of feature scaling is necessary in most machine learning models to optimize their performance. By scaling the features, the bias is removed between features that display different magnitudes. Robust scaler is used for this dataset. Other scaling methods such as Standard scaler and MinMax scaler are sensitive to outliers due to their inherit mathematical formula. Robust scaler overcomes this problem by scaling the features based on the first and third quartile.

Variance threshold

For optimal development of the supervised learning model, features associated with low, and zero-variance need to be removed from the feature set. Low-variance features do not provide sufficient information for the supervised model to differentiate changes in target viscosity. Zero-variance viscosity features provide no information since they remain constant for each rock samples. In addition, very high-variance features can be detrimental to model development due to their high noise. The Variance Threshold technique eliminates the features outside of a user-defined lower and upper variance range. For this application, a lower threshold is established at 0.05 and the upper threshold at 150.

Mutual Information and F-Test Score

While the Variance Threshold method only looks at information within each feature, Mutual Information and F-Test captures interactions between each feature and the target. Both methods are applied in conjunction to exclude features depicting weak association with the target viscosity. MI describes the association between feature and target both linearly and non-linearly. MI values ranges between 0 and 1. A feature with low MI score shows a weak with the target. As a result, what the model learns from this particular feature does not contribute to the prediction of the target. F-Test measures the linearity between feature and target. For each feature, F-Test compares the sum of squares of errors (SSE) between 2 models. The first model is generated with the feature converted to a constant. The second model combines both the feature and the constant. The score is determined by the ratio of difference in SSEs between the first and second model and the SSE of the second model divided by the degrees of freedom of the second model, which equals to the number of samples minus 2. F-Test becomes larger as the

difference in SSE between model 2 and model 1 becomes smaller indicating greater feature linearity to the target. In summary, a feature is linearly related to the target when the F-Test score becomes large. After calculating MI and F-Test scores for the entire dataset, thresholds are applied to both methods to exclude features with both low MI and low F-Test. A threshold of 0.01 and 5 is applied on the dataset for MI and F-Test score respectively. Features that presented a low MI paired with a higher F-Test value and vice-versa are not removed. In addition, the highest 10 features for MI and F-Test scores are shown in Figure 6. For UV-light images, features derived from the Sato image-based filter show the strongest relationship with target. On the contrary, for white-light images, red, green, and blue pixel intensity features show the greatest association with target.

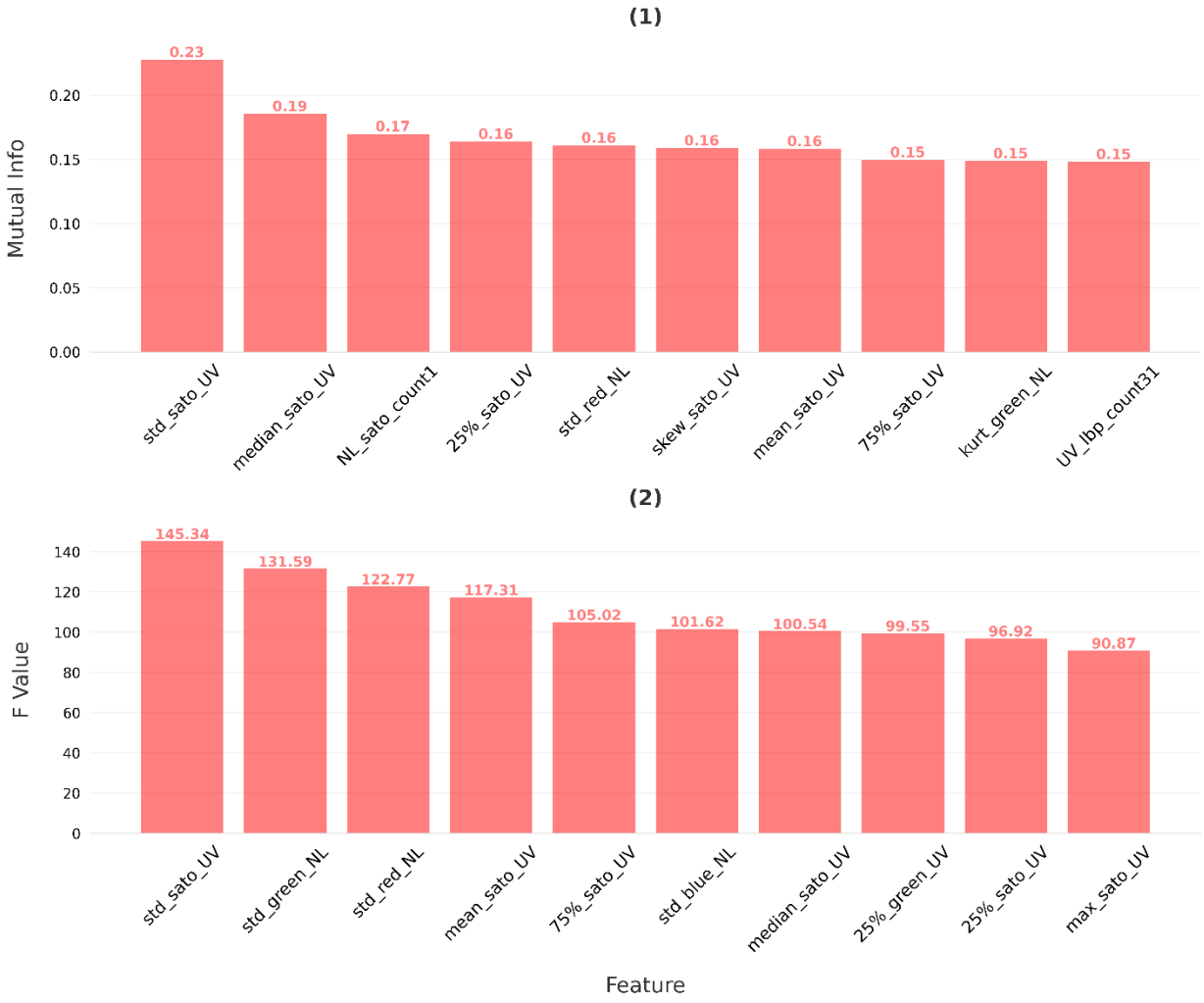

Figure 6: Top ten highest MI (1) and F score (2) observed during feature selection. White-light images show greater relationship with target viscosity on pixel intensities derived features. UV-light images have the strongest relationship to target with image-based filters derived features.

Pearson's r Correlation Coefficient

High collinearity between features can potentially decrease the machine learning model performance. Pearson's r is used to evaluate collinearity between features in the dataset. The r value, which captures the degree of linear relationship between two features, is calculated for each possible pair. Pearson's r ranges from -1 to 1. The greater the absolute value, the stronger

the collinearity between two variables. A r value of 1 represents identical variables whereas a value of -1 represents opposite variables. A value of zero shows no relationship between the variables. After deriving the r value for each possible pair of features, the resulting scores are converted to absolute values. A threshold of 0.9 is imposed to eliminate all features that show high collinearity using Pearson's correlation coefficient.

Isolation Forest

Presence of outliers can greatly affect the regression or classification performance of a machine learning model. Identifying outliers can be really challenging in the presence of a large dataset. Thus, outlier detection techniques namely Isolation Forest is deployed to locate and exclude outliers in the data. Isolation Forest is a tree-based model for outlier detection. The model randomly selects a feature and a split value across the feature range in a recursive manner. The model then implements a decision function based on the number of splitting, referred to as depth, required to isolate a sample. Shallower samples are identified as outliers based on a determined contamination value. Isolation Forest detects 31 out of the 684 samples as outliers.

Yeo-Johnson Transformation

Due to their mathematical nature, lognormal distribution of features can decrease model performance of machine learning models. Normal distribution of feature is an underlying assumption in many machine learning algorithms and statistical techniques. Statistical tests, namely t-test and ANOVA perform better with normally distributed features. Several machine learning algorithms perform better when the features have an underlying Gaussian distribution (Brownlee, 2020), e.g., KMeans and Logistic Regression. After the application of multiple feature selection techniques, a large portion of the remaining features present a highly skewed

distribution. Yeo-Johnson transformation is utilized to convert lognormally distributed features into a normal distribution. Yeo-Johnson, as well as Box-Cox transformation, refers to a group of power transform methods. These methods allow reduction in skewness by stabilizing the variance. Particularly, Yeo-Johnson is chosen for its ability to operate on positive and negative values in the features distributions.

Principal Component Analysis

Optimization of machine learning models is depended on feature dimensionality. Principal Component Analysis is implemented to condense the number of features into a smaller dimension. Principal components are defined as the linear combinations of features that explain most of data variance. PCA recursively decomposes the dataset into multiple set of orthogonal components that explain the greatest amount of variance in the data along the directions of the components. The initial number of principal components equals the original feature space dimension. From there, PCA calculates the variance explained as the number of principal components decreases. Reduction of principal components is inversely proportional to the variance explained. A lower number of principal components can improve model computational time and training speed. However, with a reduction in principal components more information is lost by a lesser percentage of the variance being explained from the data. Therefore, a threshold of 95% variance explained is applied to ensure a small loss of information. Figure 7 shows the relationship between the number of principal components and the percentage of variance explained. Based on these results, 6 principal components explain a least 95% of the variance in the data.

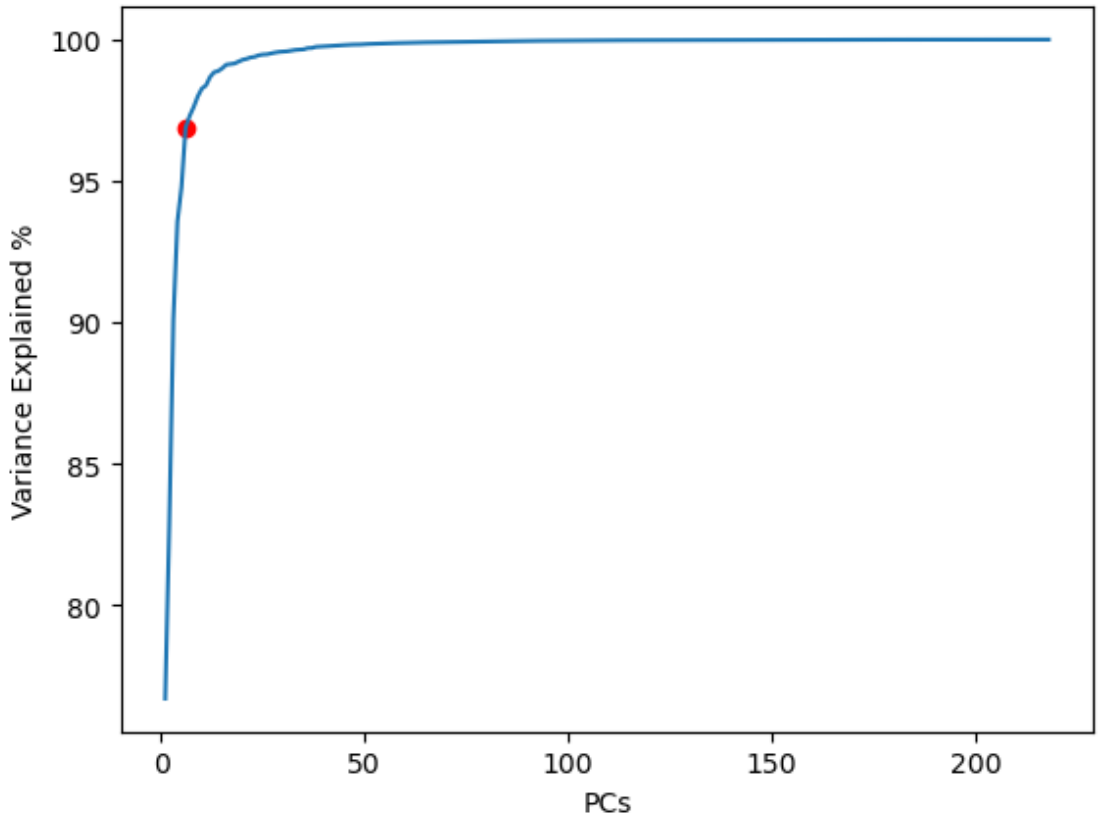

Figure 7: Percentage of variance explained by the data as a function of number of principal components during PCA. This chart illustrates the percentage of information retained (y axis) resulting from reducing the data to a certain number of principal components (x axis). Using 6 principal components results in over 95% of variance explained from the reduce data after PCA.

Summary of Feature Selection

In summary, 6 techniques are used to reduce feature dimensionality. The original dataset begins with 600 features extracted from white-light and UV-light core sample images. Feature selection is applied to both regression and classification supervised learning models. Table 4 shows the progressive reduction of feature dimensionality from left to right as each technique is applied.

Table 4: Reduction in feature dimensionality for each technique applied in the feature selection workflow. For both regression and classification models, the total number of features is reduced from 600 to 6. The final 6 principal components explain at least 95% of the variance in the data.

Class	Initial	Variance Threshold	Mi and F-Test	Pearson's r	PCA	
Regression	600	496	427	211	6	96.9
Classification	600	496	443	224	6	96.5

Model Optimization

In this project, both regression and classification task are performed for viscosity prediction. For regression, 4 supervised learning techniques are analyzed and compared: random forest, gradient boosting, k-nearest neighbor, and elastic net. The models are optimized based on MAE and MAPE. For classification, random forest classifier is utilized. For each of the four classification tasks, the model is initialized and optimized based on F1 weighted and MCC score.

K-fold Cross Validation

Due to the small sample size of 684, K-fold cross validation is performed to ensure that the models are trained a tested on every portion of the data. The k-fold cross validation divides the dataset into k equally sizes portions. Each model is then trained on the k-1 portions and tested on the excluded portion. The process of splitting, training, and testing the models is repeated k times to ensure that every portion of the dataset has been trained and tested by the machine learning models.

Hyperparameter tuning

Hyperparameter tuning is performed to obtain a highly generalizable supervised learning model. Tuning is performed on both regression and classification models. Each model is associated with different parameters that control the learning. The wrong selection of

hyperparameters can lead to decrease performance in the model. Due to the high number of possible combinations of hyperparameters, a grid-search model is used to locate the best combination based on performance. Each combination of hyperparameters is tested across the k splits and the best set is selected based on the best performance metric. Hyperparameter tuning is performed for each model on several parameters. Table 5 represents a list of these parameters with their initial range and the resulting optimal value for regression and classification models.

Table 5: Range and optimal values of hyperparameters of regression and classification techniques. The ranges were predetermined while the grid-search localized the optimal set of hyperparameters for each model.

Task	Technique	Hyperparameters	Range	Optimal Value
Regression	Random Forest	n_estimators	500	500
		criterion	MAE	MAE
		max_depth	6-9	7
		max_features	3-6	3
	KNN	n_neighbors	3-11	11
		p	1-4	2
	Gradient Boosting	learning_rate	0.001-0.1	0.1
		n_estimators	30-90	30
		loss	Huber	Huber
		max_depth	2-4	4
		max_features	3-6	3
	Elastic Net	alpha	0.01-2	0.011
		l1_ratio	0-1	0.5
Classification	Random Forest	n_estimators	500	500
		criterion	entropy/gini	gini
		max_depth	6-9	6
		max_features	3-6	3

Discussion of Results

Regression

As shown in Figure 8, Random Forest performs the best when looking at MAE and MAPE scores. Additionally, when looking at the error distribution, Random Forest and Gradient Boosting show the greatest standard deviation as described in the error bar on Figure 6. The average MAE and MAPE of the highest performing model are 27 cP and 19% respectively.

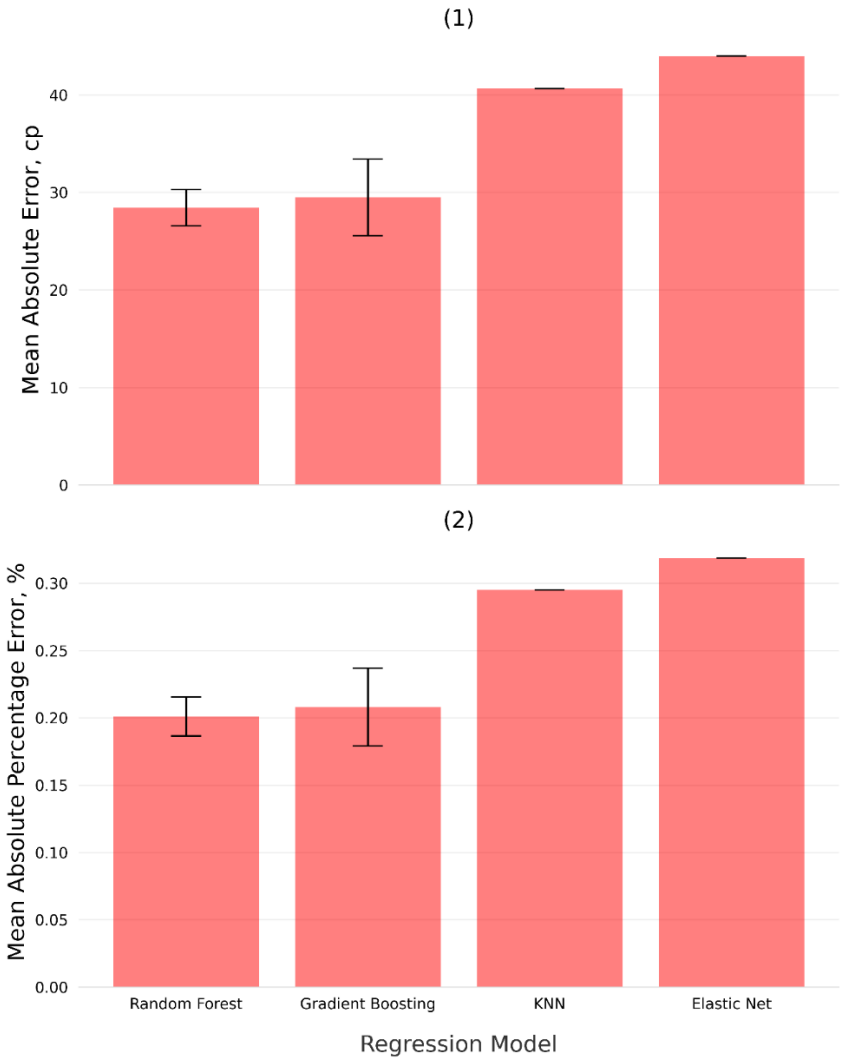

Figure 8: Performances of regression models developed using random forest, gradient boosting, KNN, and Elastic Net. Top figure (1) is MAE in cP and the bottom figure (2) is MAPE in % of

the optimized models. Random Forest performed the best for both error metrics. Values plotted on the bar chart represents the average error metric and error bars represent their standard deviation.

A scatter plot is shown in Figure 9 compares the actual vs predicted values of the best performing model which includes a unit slope line. The closest the point are on the unit slope line, the lowest the difference in viscosity between actual and predicted.

The regression model fails to accurately predict lower viscosity values under 110 cP and higher viscosity values over 160 cP. The deviation from the unit slope line becomes more apparent as the viscosity values deviates from the average. When compared to the unit slope line, the model tends to overestimate low viscosity value and underestimate high viscosity values.

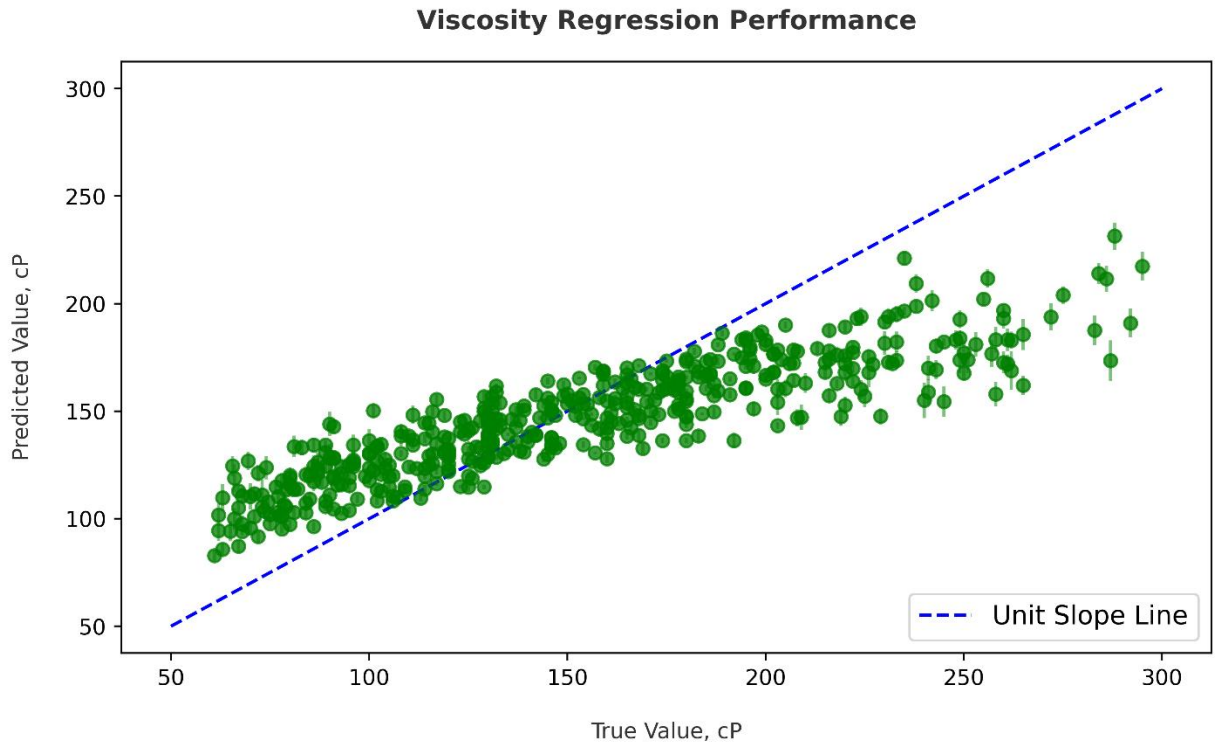

Figure 9: Performance of the best model developed using Random Forest with 7-fold cross validation. The model tends to overestimate low viscosity values and underestimate high viscosity values.

This outcome can be attributed to the original low sample size of viscosity measurements for high and low viscosity. As shown in Figure 10, the dataset does not have sufficient samples that represent high and low viscosity samples.

The regression model is not provided with enough information to accurately learn differences between high and low viscosity. Further studies can improve these results by providing new image-based filters and feature extraction methods.

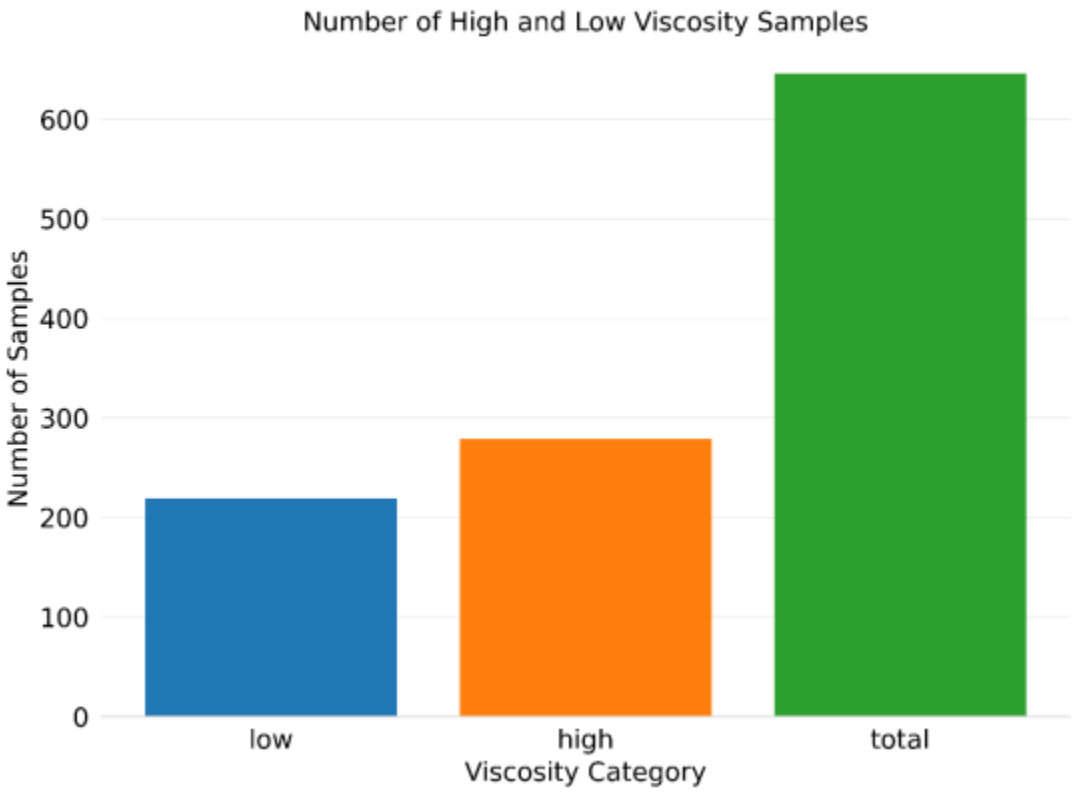

Figure 10: Viscosity samples by category. The low class represents samples lower than 110 cP. The high class represents samples greater than 160 cP. The number of samples is not sufficient for accurate deployment of supervised learning regression model.

Classification

Figure 11 shows mean and variance of f1 weighed score and Matthew's correlation score for each of the 4 classification tasks. The highest performance is achieved by classifying high viscosity samples against low viscosity samples. The task with the lowest performing metrics is detecting end points due to the greater difference in class size between end points and middle points. However, when looking at each class performance based on average error metrics, the classification model shows significant results .Standard deviations of f1 weighted and Matthew's correlation score indicate the highest variability in high vs rest classification task.

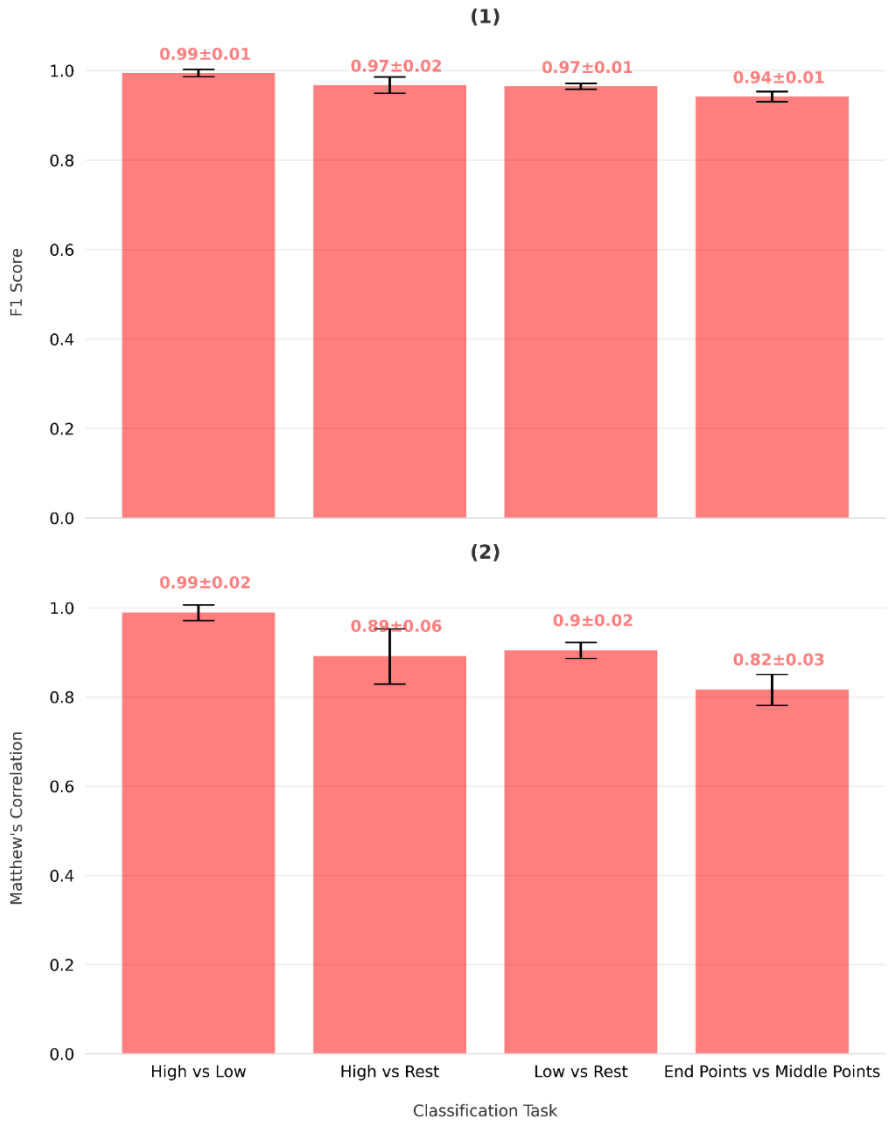

Figure 11: Random Forest classifier performance on F1 weighted score (1) and Matthew's correlation score (2). The values in the bar chart represents the mean error whereas the error bars show the standard deviation. The model performs the best in high vs low viscosity classification and the worst in the end points vs middle points classification.

Answer to Fundamental Questions

- Based on the results obtained from both regression and classification, a significant relationship with fluid viscosity is observed with color and textural features extracted from the 300 pairs of RGB core sample images under UV and white light.
- The proposed feature extraction workflow allows to identify which features show the greatest non-linear and linear relationship with target viscosity through implementation of MI and F-Test scores as shown in Figure 6. White light images show the strongest relationship with color-based features whereas UV light images show the strongest relationship with texture-based features.
- Although promising results are obtained through classification, the model regression shows significant errors for low and high viscosity. The proposed model should not be used as a standalone method but instead in conjunction with reservoir simulators for enhanced subsurface characterization.

Conclusions

The proposed workflow combines image preprocessing techniques and supervised learning models to predict fluid viscosity from side-wall core images under white-light and UV-light. Both regression and classification tasks are performed on the provided data and the results are evaluated. For viscosity regression, the machine learning model fails to accurately predict samples with viscosity <110 cP and >160 cP. Due to the small sample size of viscosity measurements, future work can improve upon the described workflow once more data becomes accessible. Classification model show promising results based on a binary selection between 2 viscosity classes. In particular, the proposed model accurately classifies high viscosity samples

>330 cP with low viscosity samples <57cP. The proposed work aims to reduce the cost and accuracy of viscosity measurements and potentially aid laboratory measurements.

CHAPTER II

COMPARISON OF KERNEL-BASED QUANTUM AND CLASSICAL MACHINE LEARNING MODELS FOR LITHOLOGY CLASSIFICATION FROM PETROPHYSICAL LOGS

This chapter illustrates the application of a quantum-enhanced machine learning algorithm to classify rock lithology based on well-log data. Quantum-enhanced machine learning represents a hybrid class of models that utilizes both quantum and classical computers to apply classification and regression tasks. Readings from Gamma Ray, Resistivity, Neutron, and Density logs are fed into the quantum algorithm to classify the lithology of the side-wall core samples at that corresponding depth. Both binary and multi-class classification are performed to understand the behavior of a machine learning model operating in the quantum space. Furthermore, an analog classical machine learning model is developed on the same dataset. Comparison between classical and quantum models gives more insights on future application of quantum-enhanced machine learning for lithology classification.

Terminology

- Quantum-enhanced machine learning**:** application of quantum calculations on classical machine learning models through the use of quantum simulator or devices with the goal of performance and computational time improvements.
- Qubit: unit of information for quantum computer which is the analogue of bit for classical machines. n qubits can contain 2^n bits of information. For machine learning applications, the number of qubits represent the dimension of the feature space.

- Superposition: qubits can represent a state value of 0,1, or any proportion of the two as probability of occurrence. Classical bits can only assume a value of 0 (on) and a value of 1 (off). This property allows quantum computers to process data much faster.
- Entanglement: qubits can be entangled with one another meaning the state of one qubit directly affects the state of another qubit. If two qubits are entangled, the state of the second qubit can be derived by measuring the state of the first qubit.
- Quantum simulator: classical device built with traditional hardware that simulates the behavior of a quantum computer. Due to the limitation of the classical hardware, the computational time becomes slower as the number of qubits increases.
- Quantum computer: machine that operates based on the law of quantum physics which holds and process information using sub-atomic particles (qubits) as compared to classical machine which use transistors (bits). The premise of quantum computers is the ability to perform immense amount of calculations in a short period of time which otherwise would be impossible to be done by a classical computer.
- Kernel Matrix: matrix used in kernel-based methods to compute the similarity between each pair of points in the feature space by calculating their distances. The matrix is NxN dimensions where N is the number of samples. The matrix is used to identify support vectors and generate a decision boundary for classification.
- Radial Basis Function (RBF): kernel function widely deployed in support vector machine methods which computes the similarity between two points in the feature space with the use of Euclidian distance (l2 norm) and the hyperparameter gamma.

Fundamental Questions

- Can kernel-based quantum models provide a greater accuracy compared to classical models in classifying rock lithology based on well log data?
- Can quantum advantage outperform classical machine learning when limited samples are available?
- Can quantum advantage provide faster results?

Novelty and Scientific Impact

- Tuning the feature map, repetitions, and entanglement type improves the accuracy of quantum support vector machine.
- A kernel-based quantum model provides similar performance in binary classification of rock lithology to a classical model
- When more than two lithologies are considered, a quantum approach does not provide an advantage to classical models.

Literature review

In the recent years, several studies have been conducted on the application of kernel-based quantum-enhanced machine learning models. Table 6 shows the work performed on the application of quantum-enhanced machine learning. The current literature is still at the early stages with few examples testing the validity of quantum-enhanced machine learning models on real-world applications. As shown in the research papers below, majority of applications are being performed on synthetic datasets or simple open-source data available on the python scikit-learn library such as the breast cancer dataset. When comparing classical to quantum-enhanced

machine learning models, as shown in Table 7, there are conflicting opinions. Park shows a greater classification performance of quantum models in complex datasets (Park et al, 2020). On the other hand, Chelli arrives to the opposite conclusion stating that classical machine learning models outperform quantum applications (Chelli et al, 2019). Table 8 shows the studies comparing the two to better understand what types of tasks can achieve a quantum advantage There is very limited research of applications of quantum models on petroleum engineering tasks. Liu presents a practical application of lithology classification using a hybrid neural network with a quantum layer which results in similar performance when compared to a classical convolutional neural network (Liu et al, 2021).

Table 6: Recent work performed on the applications of kernel-based quantum-enhanced machine learning models. The current literature is skeptical on real-world applications of QSVM models.

Author	**Goal**	**Results**
Havlicek et al, 2020	Binary classification of complex synthetic data	QSVM shows great results only if the kernel is hard to compute classically
Liu et al., 2020	Classification of the discrete logarithm problem	QSVM can classify the data unlike classical learners
Kariya et al, 2018	Identify if a bill is real or forged.	QSVM contains several technical difficulties for real-data applications
Schuld et al, 2019	Comprehensive study on kernel quantum methods	QSVM shows promising results in real world applications

Table 7: Comparison of QSVM and classical machine learning models. There is no study available that compares quantum vs classical machine learning for subsurface characterization.

Author	Goal	Results
Park et al, 2020	Classification of synthetic and simple low-dimensional data	QSVM shows similar performance to SVC, greater performance is achieved in complex datasets.
Heredge et al., 2021	Signal-background classification task in B meson decays.	QSVM outperforms classical methods when number of inputs is low
Havenstrein et al, 2018	Classification based on breast cancer dataset	QSVM shows good performance but high running time
Chelli et al, 2019	Predict a person falling based on acceleration and angular velocity	QSVM perfroms worse than ensemble bagged tree (EBT)

Table 8: Applications of QML for Petroleum Engineering tasks. The literature is still very limited and currently progressing.

Author	Goal	Results
Piantanida et al, 2021	Reduce feature dimensionality of sensors in separator units	Quantum autoencoder successfully captures non-linear relationships
Liu et al, 2021	Predict lithology	Quantum model achieves similar results with lower parameters compared to ANN

Application of quantum-enhanced machine learning are still in the early stages for the Oil and Gas Industry. The development of quantum models is ongoing with Saudi Aramco and ENI planning to implement QML applications (O'Shea, 2022).

Introduction

Background

The goal of this work is to compare performance of quantum-enhanced machine learning versus conventional data-driven methods for subsurface characterization. In particular, the proposed models classify the rock lithology based on petrophysical well logs. The dataset utilized for this project comes from the Norwegian oil company Equinor which provides open access to the data obtained during the exploration, development, and production of the Volve field. This offshore field, located 80 meters below sea level in the North Sea, was developed in 2008 with an estimated OOIP of 173 MMBBL of oil (Wang et al, 2021). The oil target formation is a sandstone from the Middle Jurassic age located at an average TVD between 2700 and 3100 meters below sea levels. Petrophysical well logs such as gamma ray, resistivity, neutron, and density are provided. In addition, for each 0.5 meter, the rock sample is labelled according to its lithology. Figure 12 shows the logs and lithological interpretation for one of the two wells considered in this work.

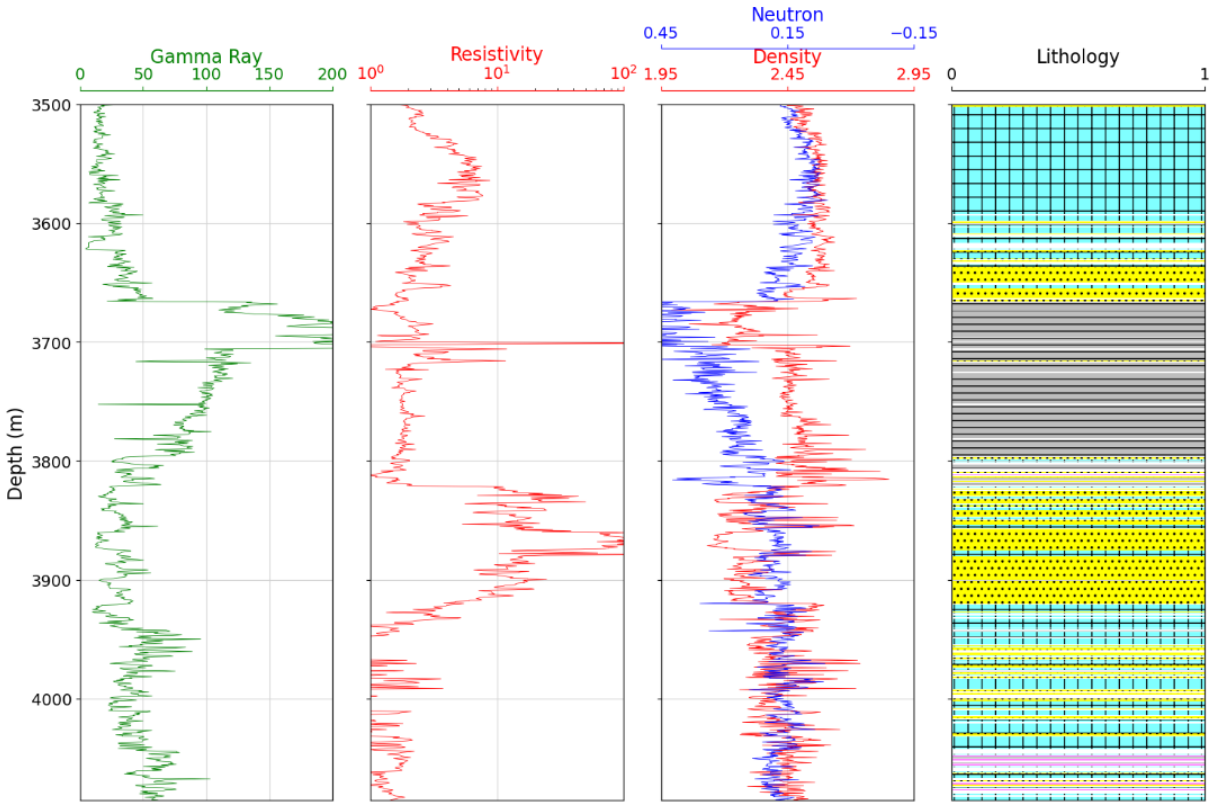

Figure 12: Well log data and lithology interpretation used in this work. The blue, yellow, and gray regions indicate carbonate, sandstone, and shale rocks respectively.

Well logs

Petrophysical well logs have been widely used in the industry for reservoir rock and fluid characterization. Well logs provide a measurement which can be used to estimate a particular reservoir property. One of the most popular combinations of well logs is the triple combo which provides a fundamental understanding of the reservoir lithology. This combination consists of 4 different petrophysical logs.

- Gamma Ray: This log measures the gamma ray frequency emitted from radioactive isotopes Potassium 40, Uranium 235/238, and Thorium 232. The presence of these radioactive isotopes is concentrated in clay-rich rocks which helps identify shale formations.

- Resistivity: This log measures the flow of current of the formation by sending an electric charge. Resistivity is the inverse of conductivity. This log can be used to predict the fluid type of a formation. Brine shows low resistivity whereas hydrocarbons show high resistivity.
- Density: This log delivers gamma rays signals into the formation. When the signal is sent at high energy levels, the tool measures the frequency of scattered gammy ray coming from the formation which is dependent on the average electron density. The resulting electron density can be correlated with the bulk density of the formation. Rock porosity can be derived when assuming on the rock matrix density.
- Neutron: This log estimates the porosity by measuring the hydrogen content of the formation. This process assumes that the hydrogen content comes entirely from the fluids occupying the pore space. The log is also susceptible to other elements such as chlorine which can cause inaccuracy when present in the formation.

Quantum-enhanced machine learning

The concept of quantum computing is still relatively new and complex. The fundamental difference between a quantum and a classical computer resides in the object of information. In quantum computer, the fundamental object of information is the qubit whereas in classical computing is the bit. Unlike the bit, which can only assume a value of 0 or 1, a qubit can have a value that is either 0, 1, or a quantum superposition of 0 and 1. The nature of the qubit allows the computer to perform calculations on many different numbers simultaneously and then interfering all the results to get a single answer (Swami, Amrata, 2011). As a result, in quantum computers are intrinsically probabilistic unlike classical computers which are deterministic due to the binary nature of the bit. Quantum computer have been successful in improving computational speed and

accuracy in several applications which includes simulation, cryptography, search algorithms, and machine learning (). Due to the expensive budget required for the creation and maintenance of a quantum hardware, quantum simulators have been utilized in research to predict the behavior of a real quantum computer. For this project, the quantum simulator was accessed from the open-source software development kit Qiskit from IBM Quantum. After subscription, the software allows to run code directly on an interactive jupyter notebook. Libraries and documentation are also provided to further understand different quantum applications. One of the most researched quantum-enhanced machine learning models is Quantum Support Machine Vector (QSVM). QSVM is a quantum version of the classical Support Vector Classifier (SVC). Researchers are still debating on the efficacy of kernel-based quantum models on real-data applications. Havlicek argues that QSVM only provides a quantum advantage over classical approached if the corresponding quantum kernel s hard to estimate classically (Havlicek et al, 2018). However, Liu shows it is possible to achieve quantum advantage over all classical machine learning models regardless of complexity (Liu et al, 2020). In this work, a practical application of quantum-enhanced machine learning is tested on real-data to compare its performance with classical models.

How to access Quantum-enhanced models

The proposed quantum-enhanced model is developed using the open source qiskit-machine-learning library which can be access both locally and on the browser after registration through the IBM Quantum website. Figure 13 shows a schematic of the process of running a QSVM model.

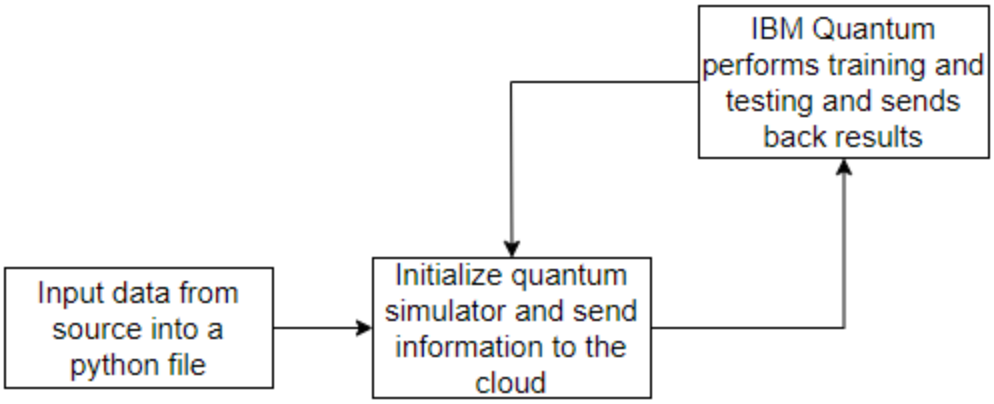

Figure 13: Process of running a QSVM model through IBM Quantum. The code is built locally and sent on the cloud to obtain results.

Alternatively, quantum-enhanced models can also be explored with the tensorflow-quantum library which uses Google's Cirq quantum simulator. Both tensorflow and qiskit allow the code to be built locally and then tun on the cloud to process and return the results. Since the two simulators are using different proprietary hardware, comparison of run time should be avoided.

Machine learning models

Support Vector Classifier

The classical kernel-based machine learning model is Support Vector Machine. This method can be used for both regression and classification tasks. For classification, the model is referred to as Support Vector Classifier (SVC). This model is a supervised machine learning algorithm in which both features and target from a training set are fed into the model to create a prediction on the testing set. This supervised learning model creates a decision boundary to separate the data into their appropriate classes. The decision boundary, referred to as the hyperplane, has the same dimensions as the number of features in the model. Margins represents

parallel hyperplanes that pass through the nearest points to the original decision boundary. These points are referred to as support vectors which can be seen as samples of one class that have the most similarities with points from the opposing class. SVC tries multiple orientations of the hyperplane that best separates each class in the dataset by maximizing the distance between the hyperplane and the margins. By doing so, the model can generalize and thus, obtain a better performance when applied on a testing set. Figure 14 gives representation of SVC applied on binary dataset containing shale and sandstone samples based on well log information. To illustrate the process on a 2D plot, the number of features has been reduced to 2 principal components.

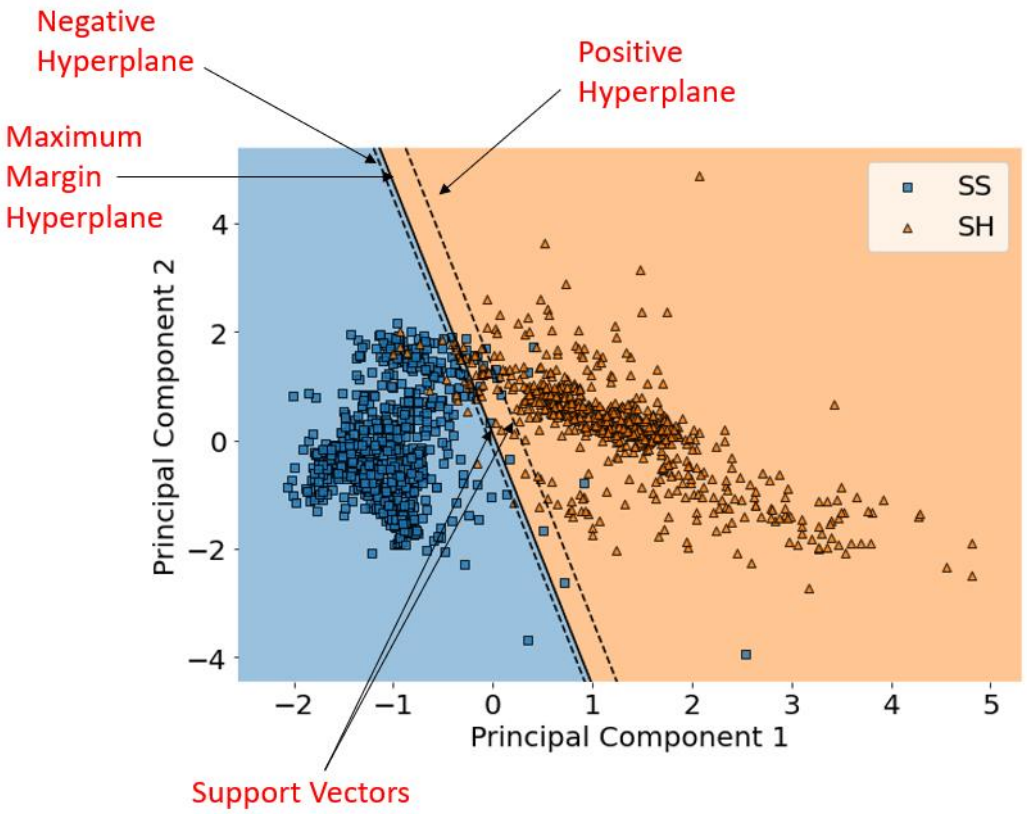

Figure 14: The mechanism of action of SVC. The hyperplane classifies the datapoints as either shale or sandstone. The support vectors influence the position and orientation of the decision boundary. The model underperforms when there is no clear separation between the classes.

The application of SVC in Figure 14 represents an example of linearly separable classes. The issue arises when the data is non-linearly separable. In this case, SVC creates a kernel function that transforms the original data into a higher dimension to create a non-linear hyperplane able to separate the classes in the data. Polynomial, RBF, and Sigmoid are the main kernel functions used in SVC. All three kernel functions aim to achieve the same objective but with different mathematical formulas. The choice of one over the other is strictly dependent on the data in question. SVC optimizes the hyperplane position based on a regularization parameter C. High value of C can lead to overfitting whereas low values can generate a poor training performance. Gamma controls the strength of non-linear hyperplanes. Higher gamma values result in poor generalization due to overfitting the training dataset. Overall, SVC represents an effective supervised learning model for both regression and classification tasks. However, the model tends to underperform when classes are overlapping.

QSVM

Quantum support vector machine represents the quantum solution of the classical SVC model. The features from the original dataset are encoded on a quantum feature map. The model estimates the quantum kernel matrix on both training and testing datasets. The quantum kernel is then fed as a callable function to a classical SVC model. The classical model performs the training and testing using the quantum kernel function to obtain an accuracy score. Figure 15 describes the steps in the application of a QSVM model.

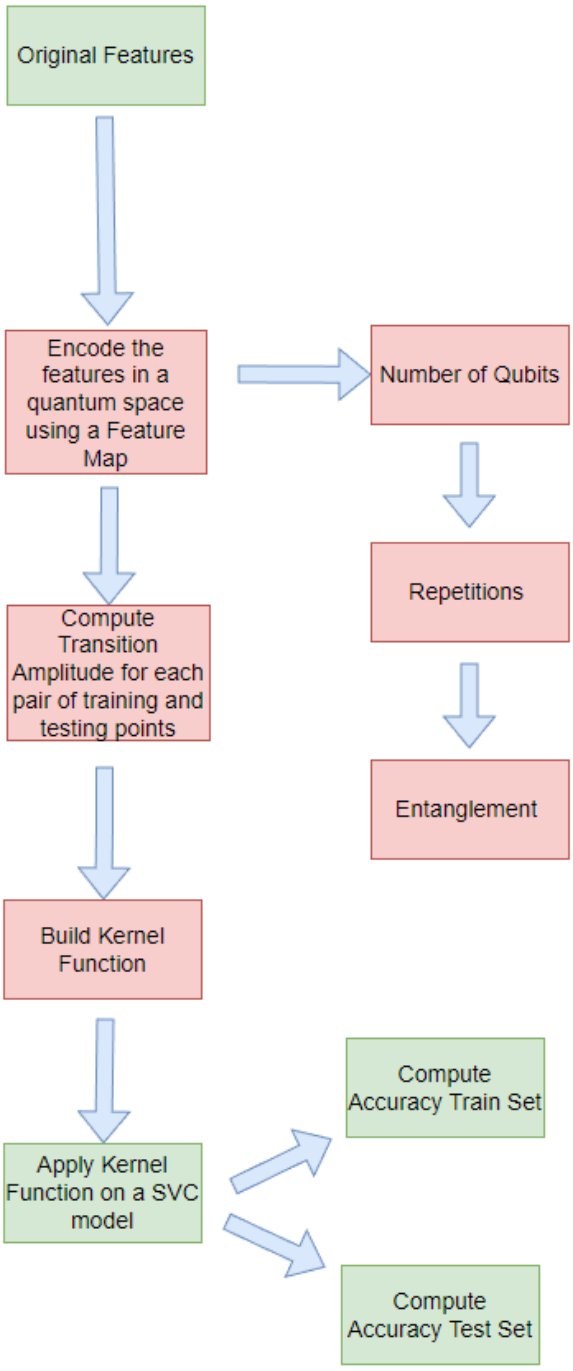

Figure 15: Flowchart of the structure of a QSVM algorithm. The green cells represent steps performed on a classical computer whereas the red cells show the steps performed by a quantum simulator.

The first for the QSVM method is encoding the original features into a quantum space. This process is performed using a feature map. For each datapoint, the feature map creates a link from the classical feature vector to a vector with 2^n dimensions where n is the number of qubits. Based on the literature presented, the Pauli first order (Z), and Pauli second order (ZZ) are the most studied feature maps. Unlike ZZ, Z map does not model entanglement between qubits. As a result, the ZZ Feature map is conjectured to be hard to simulate with a classical computer and can be implemented on quantum devices (Havlicek et al, 2019). The goal of a feature map is to represents the individual samples into a higher dimensional space to quantify the distance more accurately between each pair of samples. An accurate computation of the distance results in a greater classification performance. The choice of feature map is dependent on the data. When selecting a feature map, the number of qubits, repetitions and entanglement type must be defined. The number of qubits refers to the feature dimension of the original dataset. Repetitions represents the number of times the feature map is repeated. Lastly, entanglement represents the connectivity between qubits which can be expressed as linear or full. In linear, qubits are entangled in series whereas in full, each qubit is entangled to each other. Once the feature map is defined, the kernel function is computed by calculating the transition amplitude for each pair of samples in the training and test set. The transition amplitude is derived by the inner product of each pair of samples. Figure 16 shows the kernel matrix obtained using the well log dataset. The matrix has NxN dimensions where N is the number of training samples. The darker regions indicate the greatest similarity between points whereas the lighter regions represent the lowest similarity between datapoints. The matrix has the diagonal value of 1 in which the similarity is at its peak since it's comparing each point to itself. The matrix is symmetrical with respect to the diagonal since the distance from point a to b is the same from b to a. The kernel matrix is

computed with and without linear entanglement and compared using an RBF kernel with gamma value of 1. Figure 16 shows a comparison between an RBF kernel matrix calculated classically, a quantum derived kernel without qubit entanglement, and a quantum kernel with linear qubit entanglement. The darker regions represent the highest similarity between a pair of points in the dataset. Based on the corresponding heatmaps, the RBF and no entanglement matrices are fairly similar. Although, all three methods localize the same high-similarity regions, the quantum kernel with entanglement show a much greater granularity in the results.

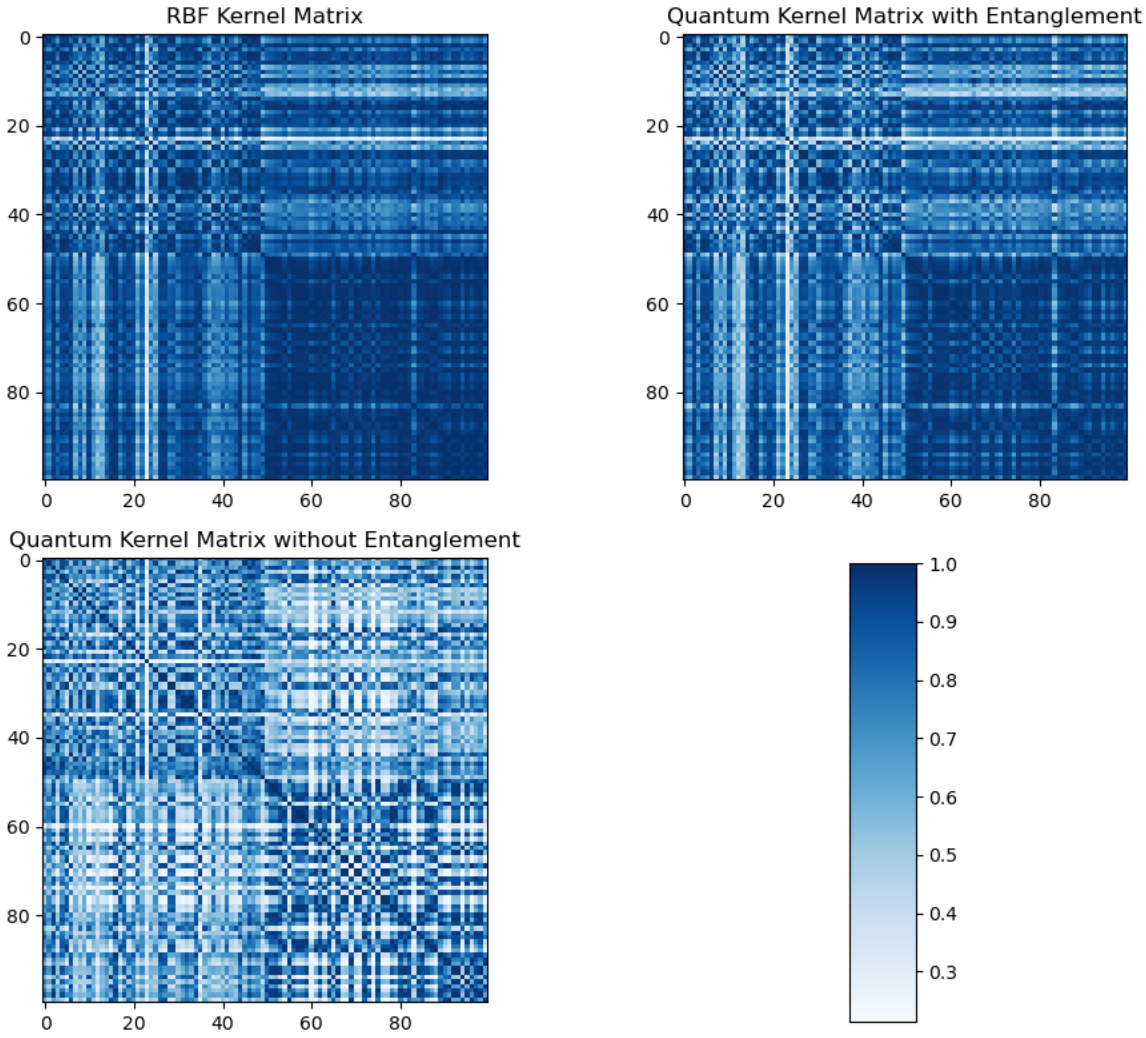

Figure 16: Comparison of kernel matrices between RBF with gamma 1, quantum kernel without entanglement, and quantum kernel with entanglement. The darker regions represent higher similarity whereas lighter regions represent lower similarity. When modelling entanglement, the results are more granular.

The kernel function is then passed onto a classical SVC model. Based on the provided function, SVC creates a hyperplane to differentiate samples depending on their class. Lastly, the resulting accuracy is computed for both training and testing to assess model performance.

Applications of QSVM

Applicability of QSVM is still debated in the literature. Havlicek argues that a quantum advantage can only be observed if the kernel is hard to estimate with a classical computer (Havlicek et al, 2020). Although Havlicek's results are based on a small synthetic dataset, several papers explore the application of QSVM on real-world dataset. Kariya demonstrate a practical application of QSVM for binary classification of forged banknotes which is made of 1372 grayscale images (Kariya et al, 2018). After further preprocessing, the 2-qubit model returns an average accuracy of 70% when tested on both a quantum simulator and actual quantum machine which significantly underperforms compared to classical machine learning methods which exhibit accuracies over 90% . Comparison of QSVM with traditional machine learning model is presented extensively by Park which compares extensively SVM and QSVM on simple and complex datasets from the python scikit-learn library (Park et al, 2020). The results show that a quantum advantage is achieved in complex datasets such as the scikit learn hand-written number dataset in which the underlying boundaries between classes is too complex for a traditional kernel to estimate. Based on the results of the proposed literature, real-world applicability of QSVM on current quantum simulators and machines is contingent upon the

complexity of the dataset in which traditional kernels are hard to classically compute. With simpler datasets such as the breast cancer dataset, classical machine learning models are perfectly capable to achieve a high classification score. The potential applicability of QSVM reside in more complex datasets such as the handwritten digits datasets shown in Figure 17.

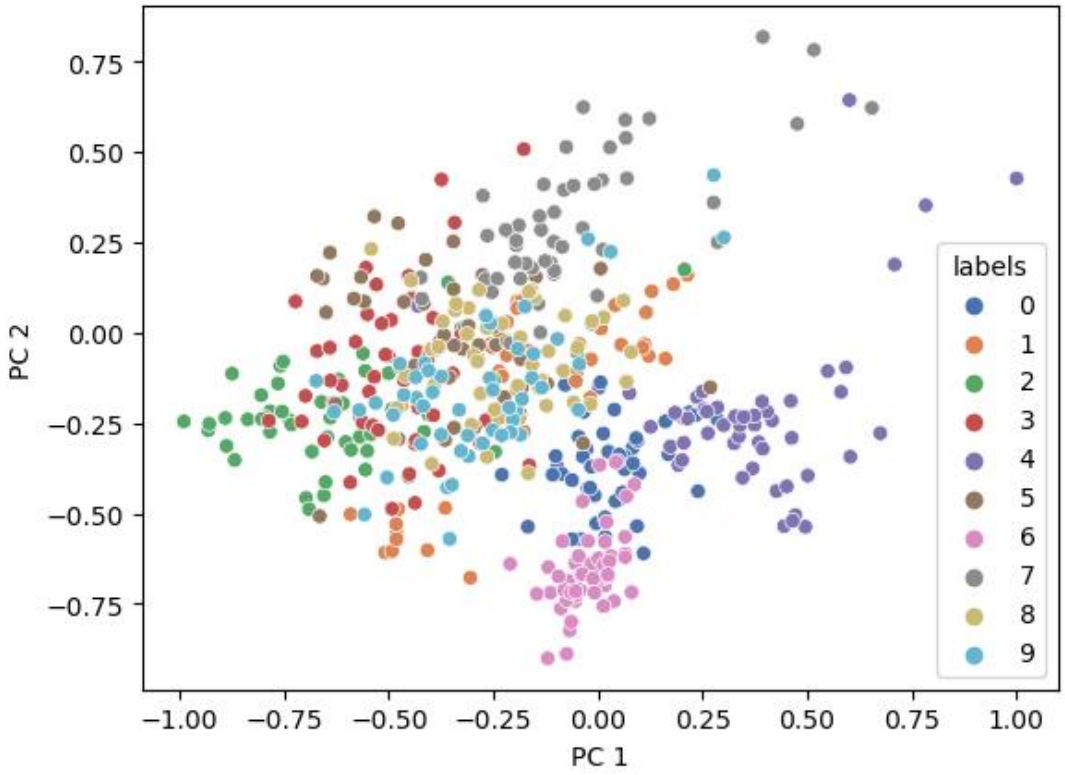

Figure 17: Complex dataset representing handwritten digits in which QSVM has shown greater performance compared to classical machine learning methods. The data is reduced to two principal components for visualization

Methodology

The workflow for the proposed method is shown in Figure 18. The method starts with raw data from the well logs and ends with a performance comparison between QSVM and SVC. For both models binary and multi-class classification are tested. For multi-class classification, all

facies in the dataset are included whereas in binary classification only sandstone and shale are considered. Samples at different sizes are selected from the original dataset. The goal is to compare the efficacy between quantum and classical machine learning model with varying sample sizes. For QSVM, feature map and quantum kernel are tuned to optimize accuracy. For SVC, conventional hyperparameter tuning is performed. The models are then compared for both binary and multiclass classification.

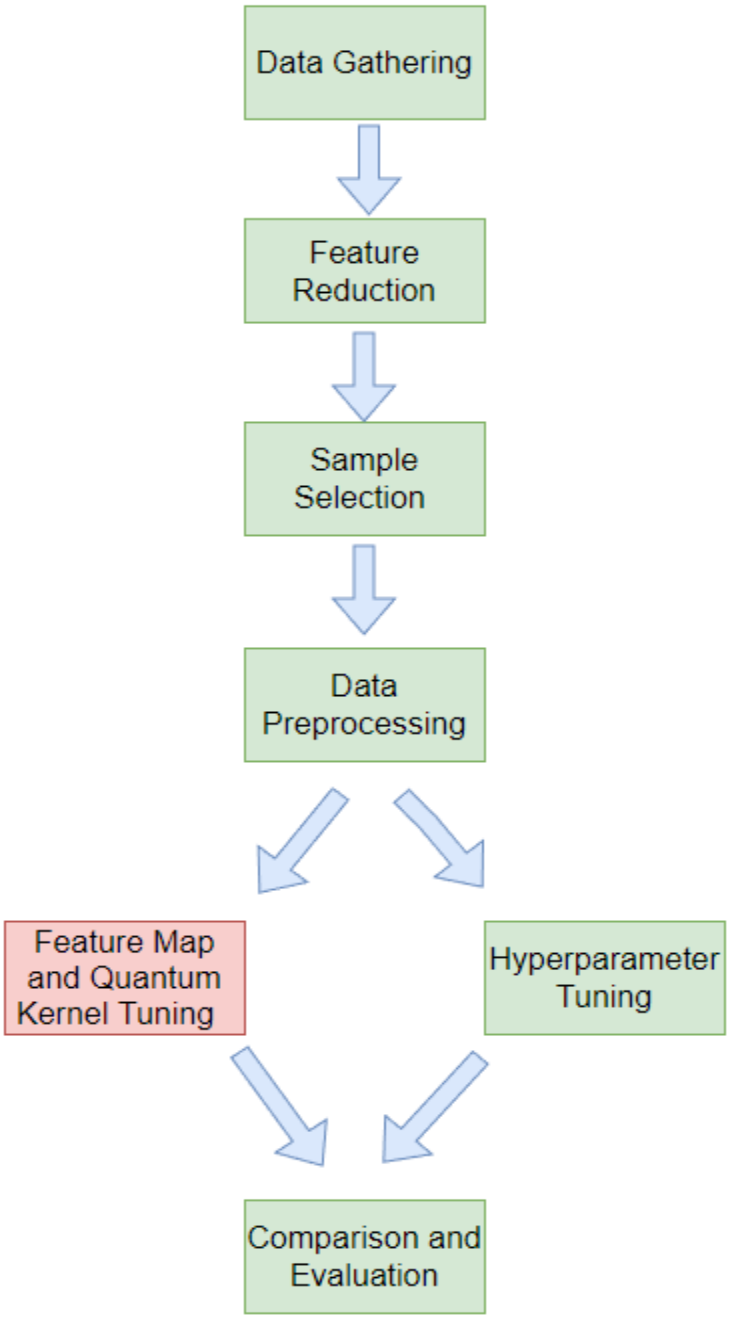

Figure 18: The flowchart represents the main steps taken in this work, from obtaining the raw features to comparing performance between QSVM and SVC. The green cells represent steps taken on a classical computer while red cells require a quantum simulator.

Data Gathering

The initial data comes from two separate wells in the Volve field named well 14 and well 15. For each well, gamma ray, resistivity, neutron, density logs are provided. In well 14, the depth interval of the logs ranges from 3178.5 m to 4220.5 m. The logs in well 15 investigate a similar depth range, starting from 3500 m to 4085.5 m. The logs perform a measurement every 0.5 ft. Each depth interval is labelled with a corresponding rock type. Rock typing is performed by extracting cores from the subsurface and performing analysis in the laboratory. This amount to a total number of recording of 2082 and 1159 respectively for well 14 and well 15. Both wells contain the same lithologies in similar proportions. Figure 19 shows a pie chart of lithologies for both wells respectively.

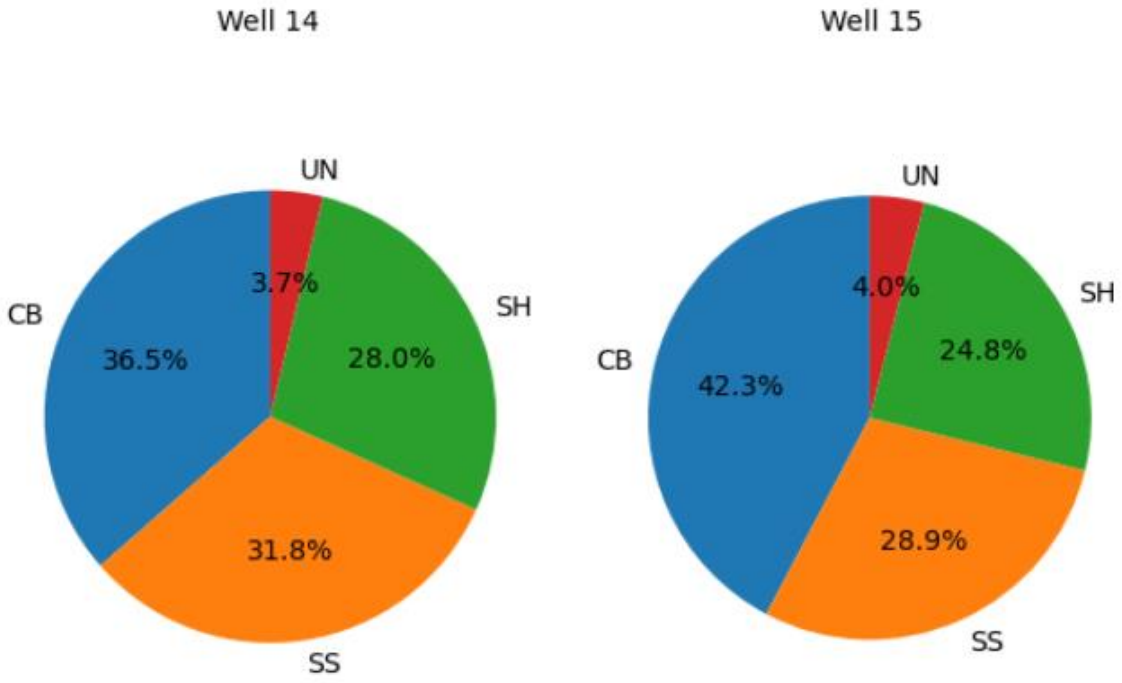

Figure 19: Proportions of different lithologies in the investigated depth for both well 14 and well 15. The lithology is consistent in both wells.

The rock types identified in this formation are sandstone (SS), carbonate (CB), and shale (SH). There is a fourth type (UN) that includes unnamed rocks. It is not known if this was done on purpose or because of potential data loss. This category does not represent a significant portion of the overall dataset in both wells. Therefore, for the purpose of this study, the UN class and associated log measurements are omitted from the dataset.

Outlier detection

Outlier detection is performed to improve both the quantum and classical models. Like in Chapter I, isolation forest is used to identify outliers. The contamination parameter is set to a small value of 0.05 which detects 13 outliers in the data. The outlier values are removed from the dataset.

Feature Selection

In feature selection, the original log measurements are compared to identify any potential features that do not contribute to the dataset. Each well log represents a feature. For this study, the original dataset already contains limited number of features. However, to optimize computational time, statistical-based methods are applied to evaluate any uninformative features. As referred to in Chapter I, Pearson's r correlation coefficient is useful to identify collinearity between features in the dataset. Figure 20 shows a correlation matrix created using the Pearson's r as a scoring metric. Each element of the matrix measures the collinearity between two features in the dataset.

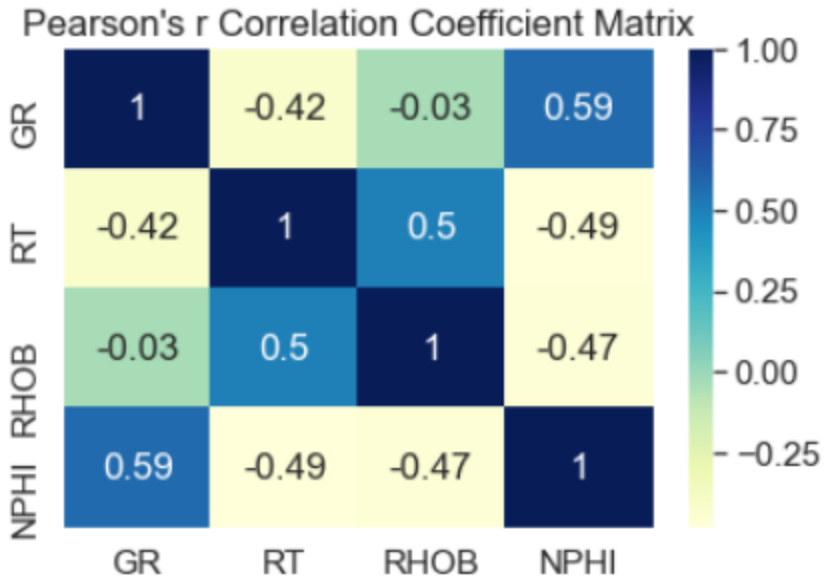

Figure 20: Heatmap of the correlation matrix using Pearson's r using data from both well 14 and well 15. Neutron and gamma ray logs display the highest collinearity.

The correlation matrix does not display any high r values. In Chapter I, the upper threshold for r value was set at 0.9. For this dataset, the collinearity between features is significantly below the threshold. Therefore, no features were excluded from the dataset.

Before moving to the next preprocessing technique, the data is split into a training and testing with an 80% ratio. Applying the split before further preprocessing ensures that no bias is transferred from the training to the test set. Stratification is applied to provide a proportional representation of rock types between training and testing datasets.

Standard Scaler

As discussed in Chapter I, features can affect the machine learning model depending on their magnitude. The features are scaled individually to a unit variance by setting the mean to 0 as shown in Equation 3. This scaling method is referred to as standard scaler.

$$\mathbf{X_{scaled}} = \frac{\mathbf{X - mean}}{\mathbf{standard\ deviation}} \qquad \textbf{Equation 3}$$

Principal Component Analysis

As described in Chapter I, PCA helps identifying linear combinations of features that retain the original information with a smaller dimensionality. For this work, the goal is to understand if the percentage of variance explained in the data decreases significantly when choosing a number of principal components lower than the 4 initial features. Figure 21 represents the percentage of variance explained by the number of principal components. Reducing the number of principal components from 4 to 3, does not cause a significant loss of variance explained. The reduction of the feature space in three principal components improves the computational performance of the machine learning model with a limited reduction in variance explained. When using a quantum simulator, feature dimensionality directly correlates to the model computational time.

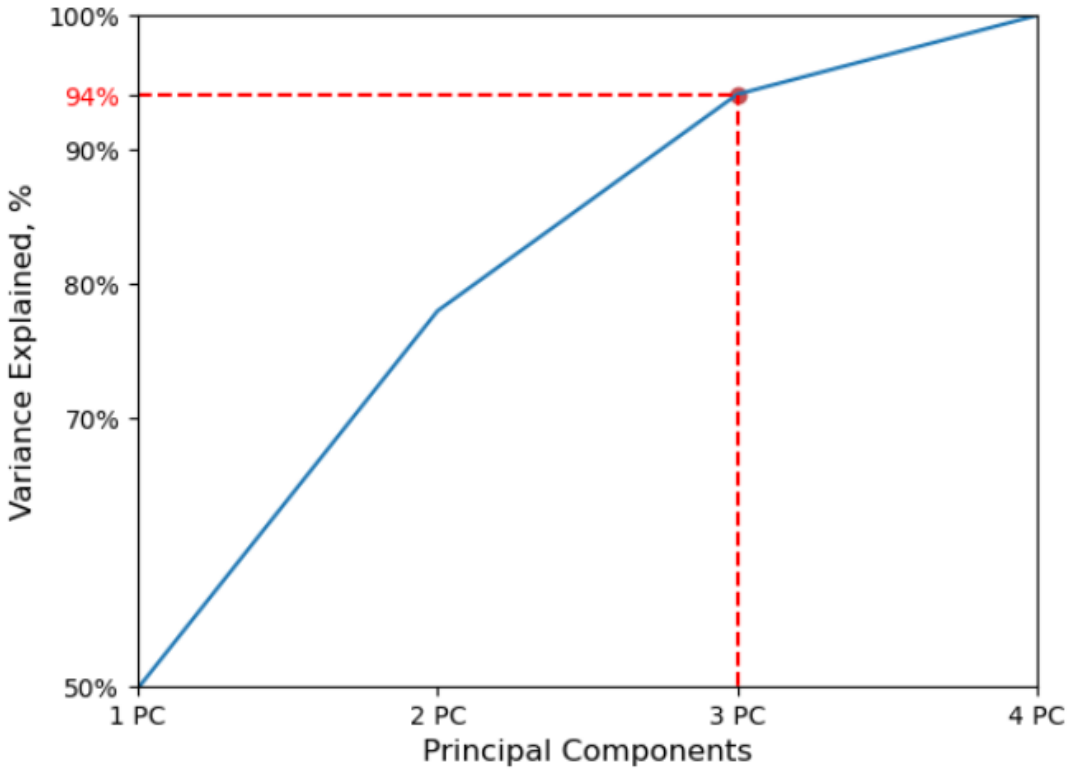

Figure 21: Percentage of variance explained as a function of principal components in PCA. Reducing the number of features from 4 to three principal components results in 94% of variance explained.

Min Max Scaler

Min max scaler is used to transform each feature to a standardized range between 0 and 1 with the following equation.

$$\mathbf{X_{scaled}} = \frac{\mathbf{X} - \mathbf{X_{min}}}{\mathbf{X_{max}} - \mathbf{X_{min}}} \qquad \textbf{Equation 4}$$

The application of a min max scaler is not necessary for this method since standard scaler was already applied. However, by scaling the feature range, it is possible to visualize the position of samples in the dataset with their respective label. Figure 22 shows a 2D plot generated by reducing the data to 2 principal components. There is not a clear separation between the three

features. Several elements belonging to the carbonate class are present in the sandstone region. The shale class shows the least contamination with the other two lithologies. These findings are consistent when considering the well logs included in the dataset. In particular, gamma ray better differentiates shale from sandstone and carbonate based on the radioactive isotopes. For this reason, shale and sandstone are chosen when designing the binary classification model.

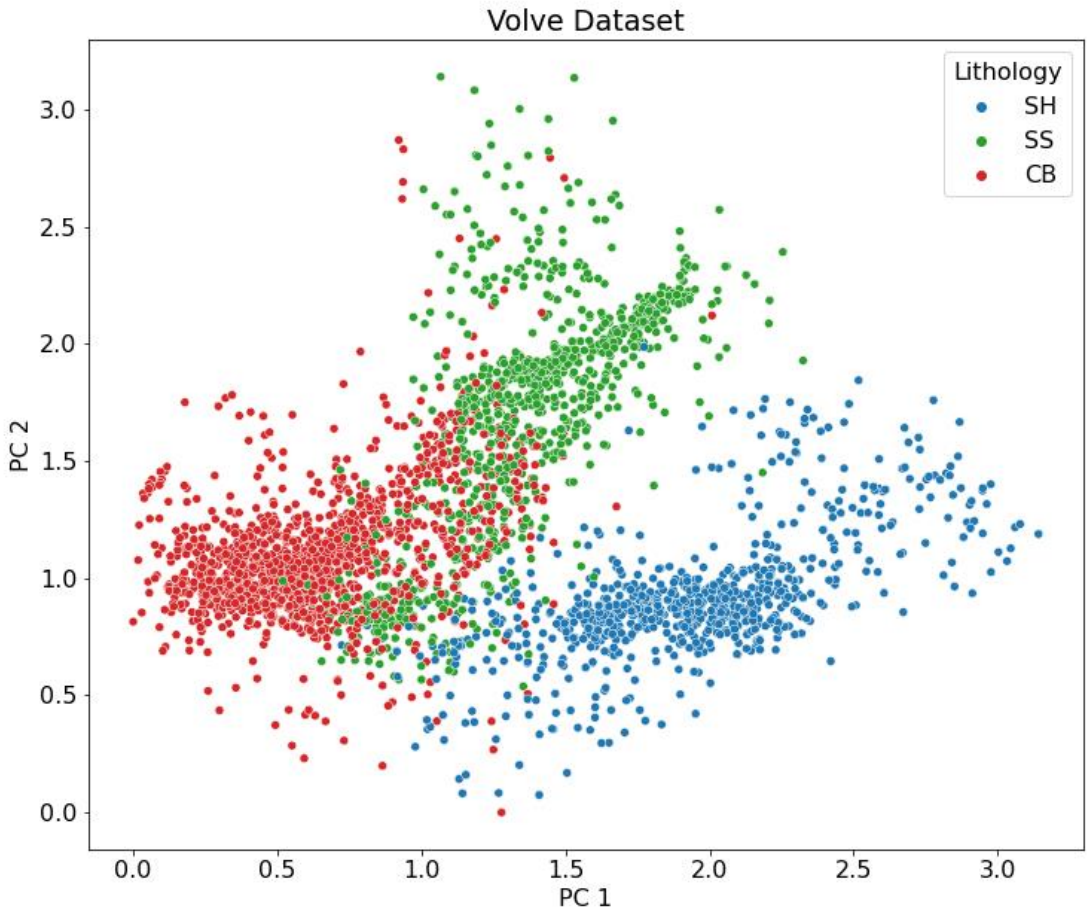

Figure 22: 2D plot of samples from well 14 and well 15 based on their lithology. There is a better separation of the shale class from the rest.

Data sampling

The study aims to assess the efficacy of quantum models at different sample sizes. For classical machine learning, the size of the dataset can play a significant role in the model's performance. Assuming no outliers, the more information available to the model translates to a greater ability to predict the target value. The original train and test datasets are randomly shuffled and divided into 4 sample sizes for both binary and multiclass classification. The binary classification includes shale and sandstone whereas the multiclass classification considers all three lithologies.

- Binary Classification
    - 40 samples
    - 80 samples
    - 160 samples
    - 320 samples
- Multiclass Classification
    - 42 samples
    - 81 samples
    - 162 samples
    - 321 samples

During this procedure, the class size proportions remained unaltered for both train and test samples. By selecting random samples, there is the possibility to obtain a dataset with classes of different size. This would negatively impact the performance of the machine learning model. Therefore, when selecting random samples, the class size proportions remained unaltered for all the cases.

Model Optimization

SVC Hyperparameter Tuning

As described in Chapter I, selecting the appropriate combination of hyperparameters improves the supervised model performance. For SVC a grid is created to train the model on each possible combination of hyperparameters. The tuning is performed on C, gamma, and kernel function. As previously discussed, gamma is only present with a non-linear kernel function. The model is trained with a 7-fold cross validation similarly to Chapter I. The best hyperparameter set is chosen based on the mean accuracy. The process is then repeated 10 times to record the mean and standard deviation of the train and test accuracies for both multiclass and binary classification. Table 9 represents a list of these parameters with their initial range and resulting optimal value based on a 7-fold cross validation. From the results obtained, in binary classification, the classes are linearly separable. For multiclass classification, a higher dimension non-linear kernel is necessary to separate the classes.

Table 9: Hyperparameter tuning of SVC model for both binary and multiclass classification

	Sample Size	**Range**			**Optimal Value**		
		C	**kernel**	**gamma**	**C**	**kernel**	**Gamma**
Binary Classification	40	0.1-100	Linear, poly, sigmoid, rbf	0.001-1	100	linear	/
	80				10	linear	/
	160				10	linear	/
	320				10	linear	/
Multiclass Classification	42				1	rbf	0.01
	81				1	sigmoid	1
	162				100	poly	1
	321				100	sigmoid	0.1

QSVM Optimization

In QSVM, a similar approach to hyperparameter tuning is taken to optimize the model performance. As previously described in the *Machine Learning Models* section, the choice of feature map can dictate the performance of the model. In this study, two feature map are considered: Z and ZZ. For each type, number of reps are varied to identify the combination with the greatest mean accuracy. For ZZ map, the entanglement type is varied between full and linear. This process is performed on both binary and multiclass classification. Table 9 shows initial range and optimal value for feature map selection.

Table 9: Feature map tuning for QSVM in both binary and multiclass classification

	Sample Size	**Range**			**Optimal Value**		
		Feature Map	**Entanglement**	**Reps**	**Feature Map**	**Entanglement**	**Reps**
Binary Classification	40	Z, ZZ	Linear, full	1-5	ZZ	linear	1
	80				ZZ	full	2
	160				ZZ	linear	1
	320				ZZ	linear	1
Multiclass Classification	42				Z	/	1
	81				Z	/	2
	162				ZZ	linear	1
	321				ZZ	linear	2

Discussion of Results

Test on Benchmark Dataset

The QSVM model is first applied and tested on a benchmark dataset. The breast cancer dataset is a public binary classification dataset widely used in ML applications. The initial data is preprocessed and reduced to 2 principal components using the same workflow as the well log dataset as shown in Figure 23.

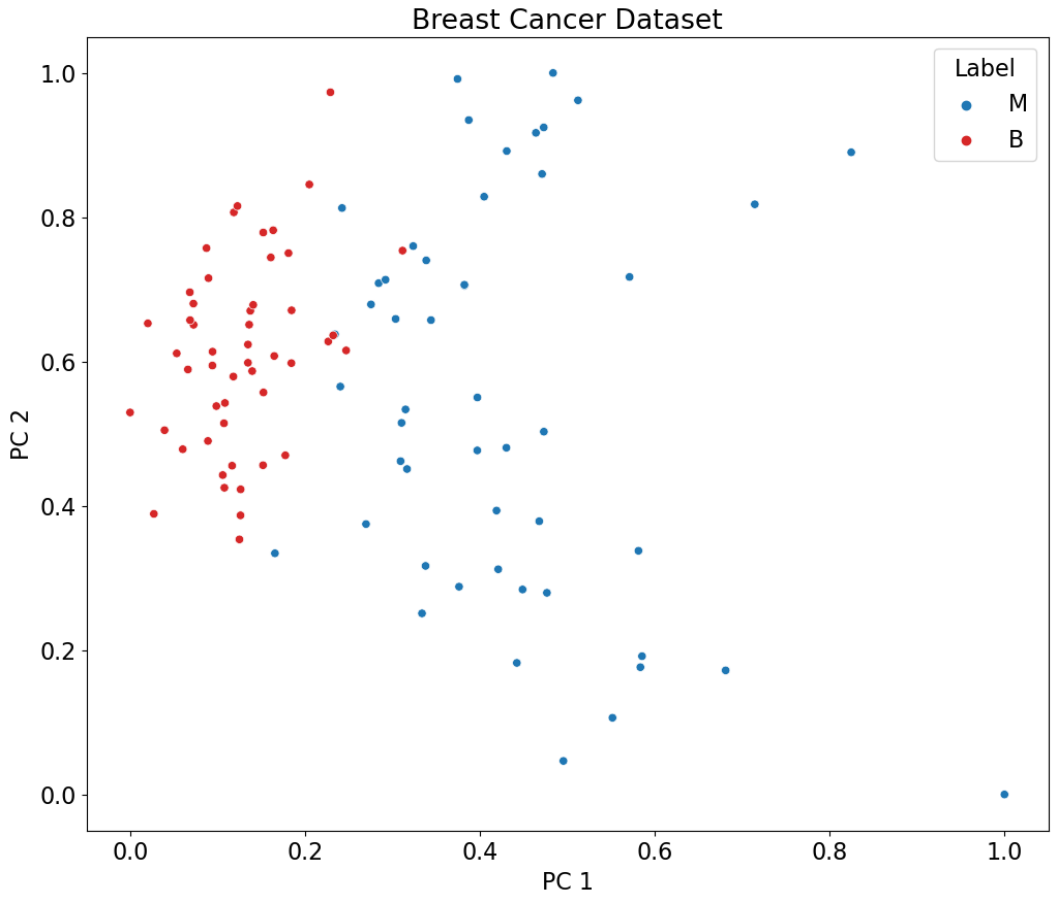

Figure 23: Preprocessed training set for breast cancer dataset. The initial data is scaled from 0 to 1 and reduced to 2 principal components. The goal is to classify the samples as either malignant or benign.

The QSVM model is designed and optimized using the same range of hyperparameters and feature map combinations. The 2D feature space is divided into 70 segments along each dimension forming a 70x70 matrix. The QSVM model is then tested on each element of the matrix and colored accordingly to the result to draw a decision boundary. Figure 24 shows the decision boundary of QSVM along with the test samples. The model separates the data non-linearly. For this specific train-test split, the model achieves 0.9 accuracy due to the 2 false positive values that belong to the malignant class.

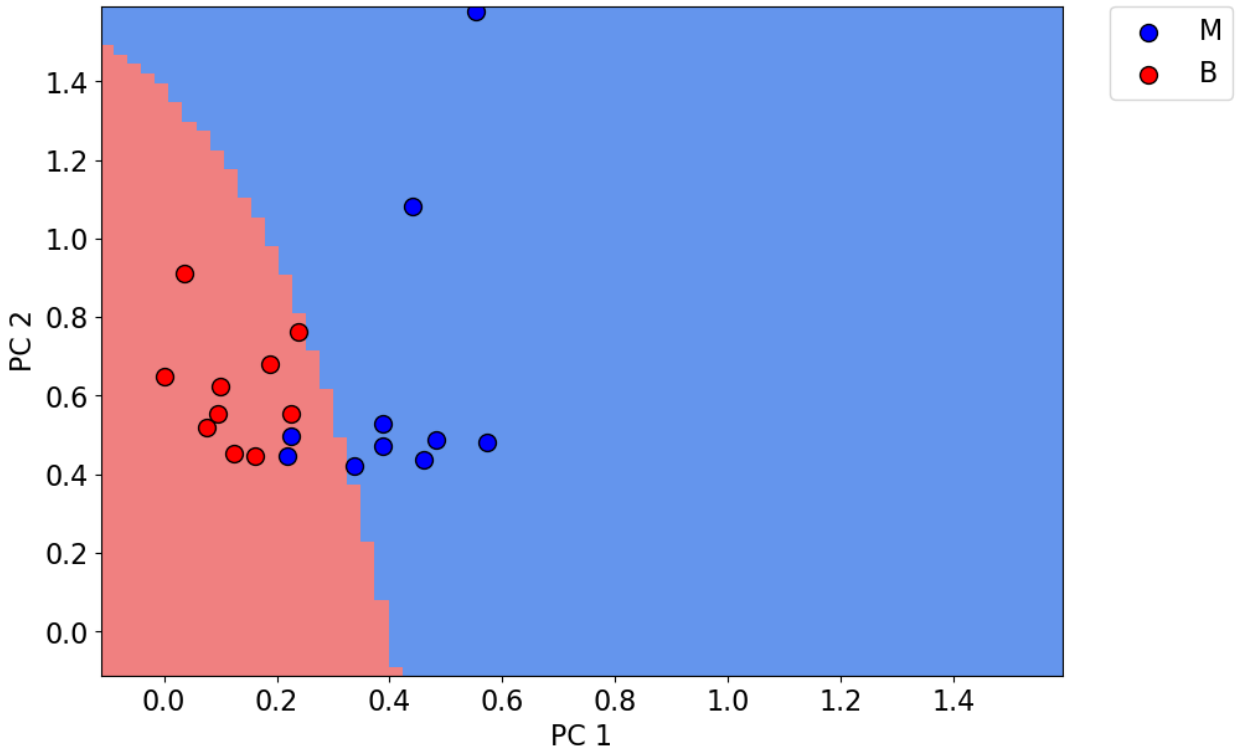

Figure 24: Decision boundary of QSVM on the breast cancer dataset. The model draws a non-linear decision boundary that achieves 0.9 accuracy.

Binary Classification

SVC and QSVM models are compared based on accuracy score on both train and test data at different sample sizes ranging from 40 to 320 for binary classification and 60 to 480 for

multiclass classification. For each sample, both models are run 10 times to record the mean and standard deviation of the accuracy score. Figure 25 and Figure 26 show the comparison of the two models for binary classification on training and testing data respectively.

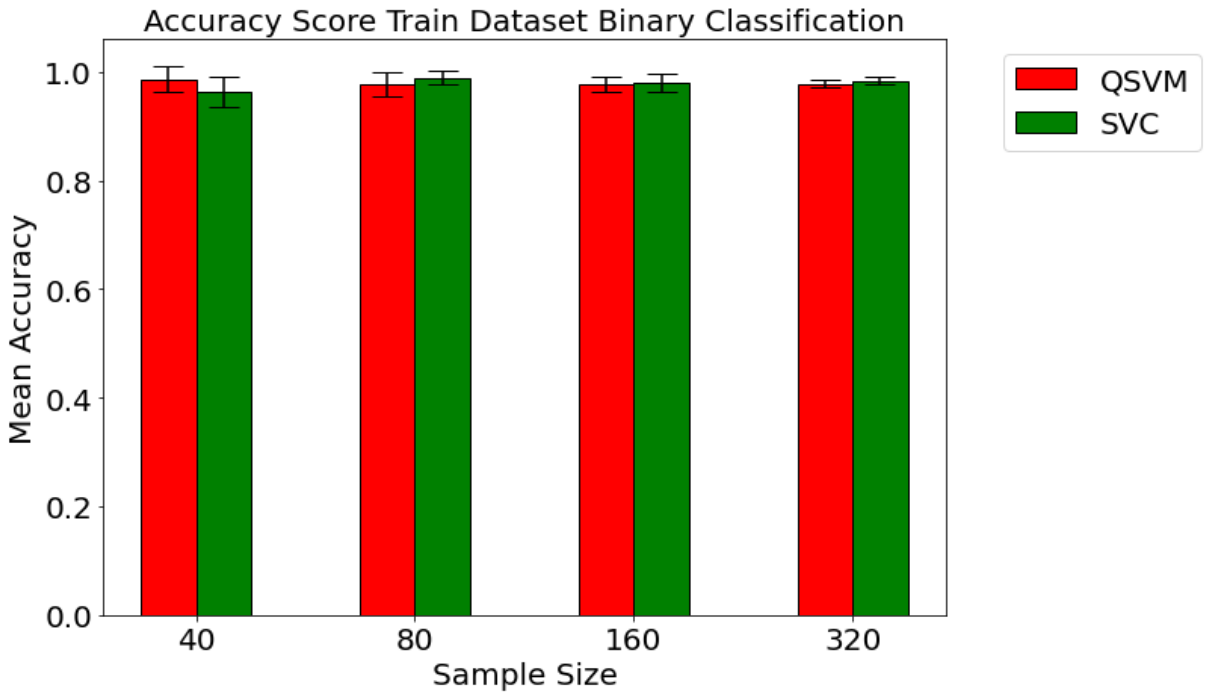

Figure 25: Recorded accuracy score on training dataset for QSVM and SVC at the respective sample sizes. The models perform a binary classification between sandstone and shale. For each sample size, the models were run 10 times to record the standard deviation of the accuracy score. The black error-bar represents the standard deviation in the accuracy score.

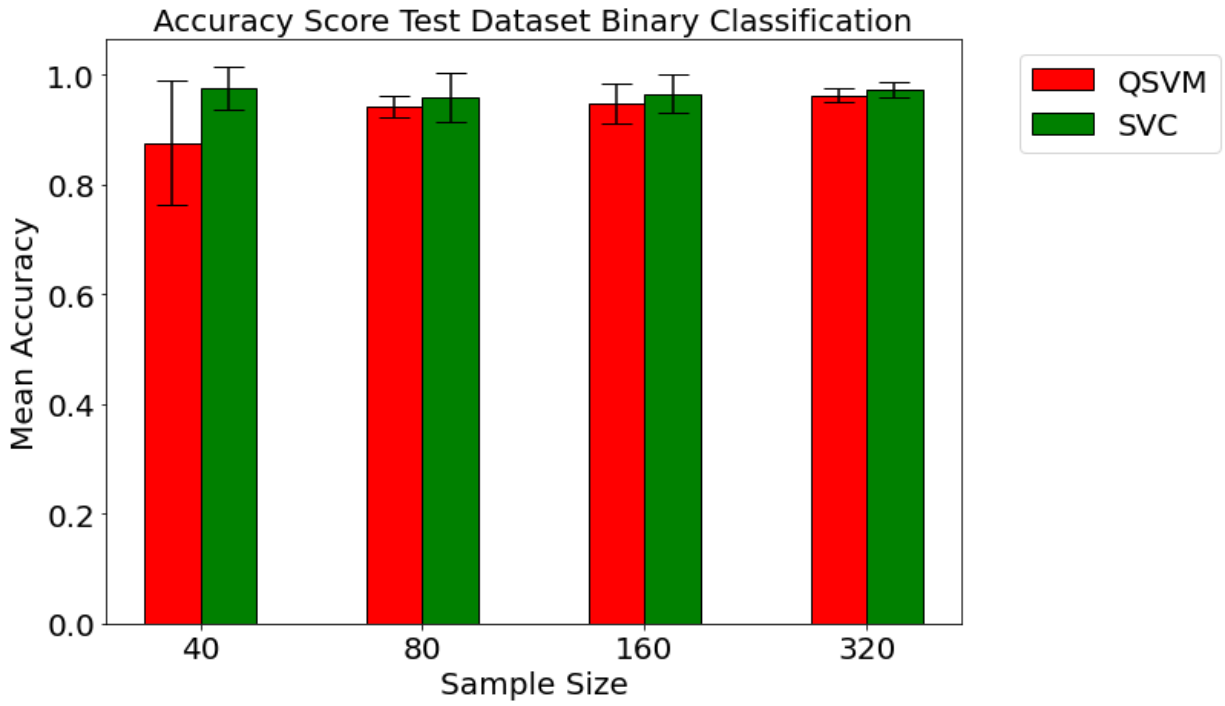

Figure 26: Recorded accuracy score on testing dataset for QSVM and SVC at the respective sample sizes. The models perform a binary classification between sandstone and shale. For each sample size, the models were run 10 times to record the standard deviation of the accuracy score. The black error-bar represents the standard deviation in the accuracy score.

For binary classification on train data, both models have good performance and low standard deviation in the accuracy score. On testing data, QSVM shows lower performance at lower sample size with a greater standard deviation when compared to SVC. Overall, for binary classification, the quantum-enhanced model tends to overfit the training data at lower sample size and underperform on the training set. At higher sample sizes, the difference in performance remains negligible between the two models. In addition, the decision boundaries of the 2 models are compared on the entire dataset by dividing the feature space into a 70x70 matrix. Figure 27 shows a comparison of the two decision boundaries. While SVC draws a linear decision boundary, QSVM decision boundary is non-linear. In the next chapter, a more in-depth analysis of the performance of QSVM on the entire well log dataset is provided.

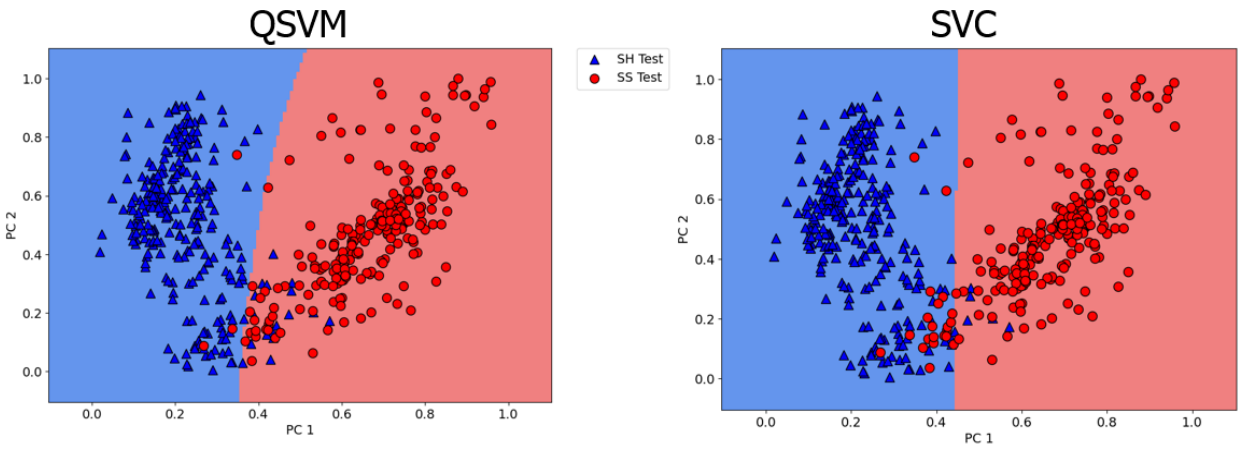

Figure 27: Comparison of decision boundaries for QSVM (left) and SVC (right) for binary classification between sandstone (red) and shale (blue). QSVM draws a linear decision boundary whereas SVC draws a linear decision boundary.

Multiclass Classification

For multiclass classification, the models classifies whether the sample is a sandstone, carbonate, or shale. Like with binary classification, both models are run 10 times to record mean and standard deviation of the accuracy score. Figure 28 and Figure 29 show the comparison of the two models for multiclass classification on training and testing data respectively.

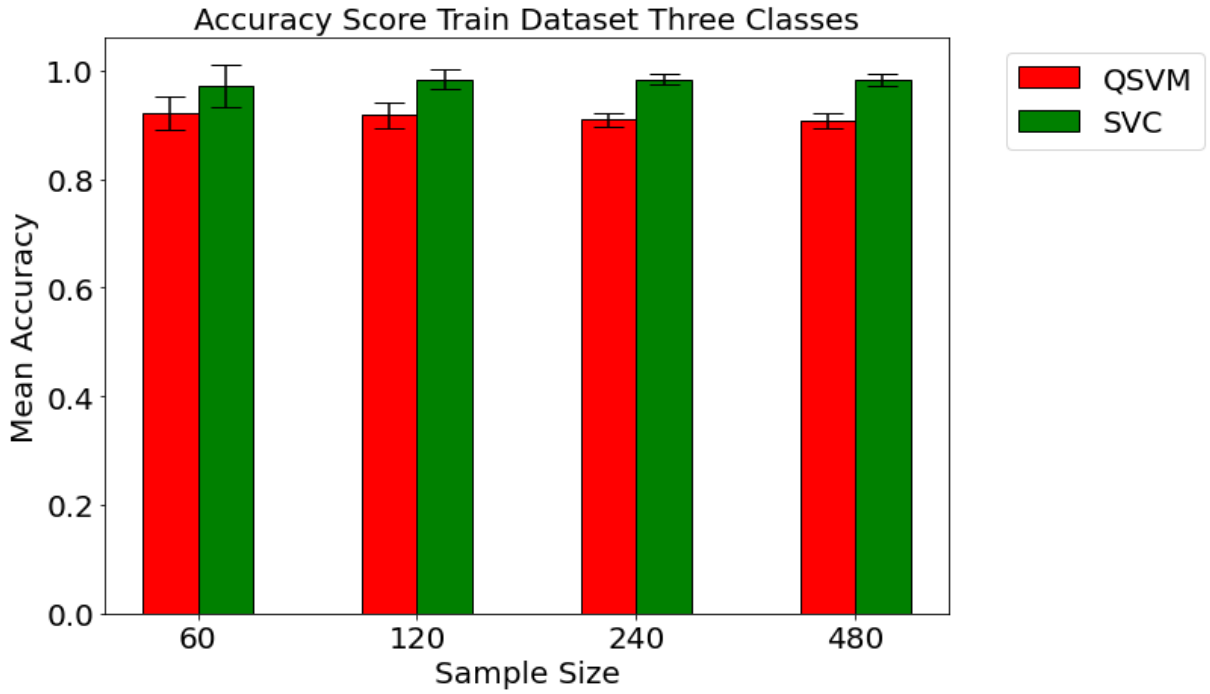

Figure 28: Recorded accuracy score on training dataset for QSVM and SVC at the respective sample sizes. The models perform a multiclass classification between sandstone, shale, and carbonate. For each sample size, the models were run 10 times to record the standard deviation of the accuracy score. The black error-bar represents the standard deviation in the accuracy score.

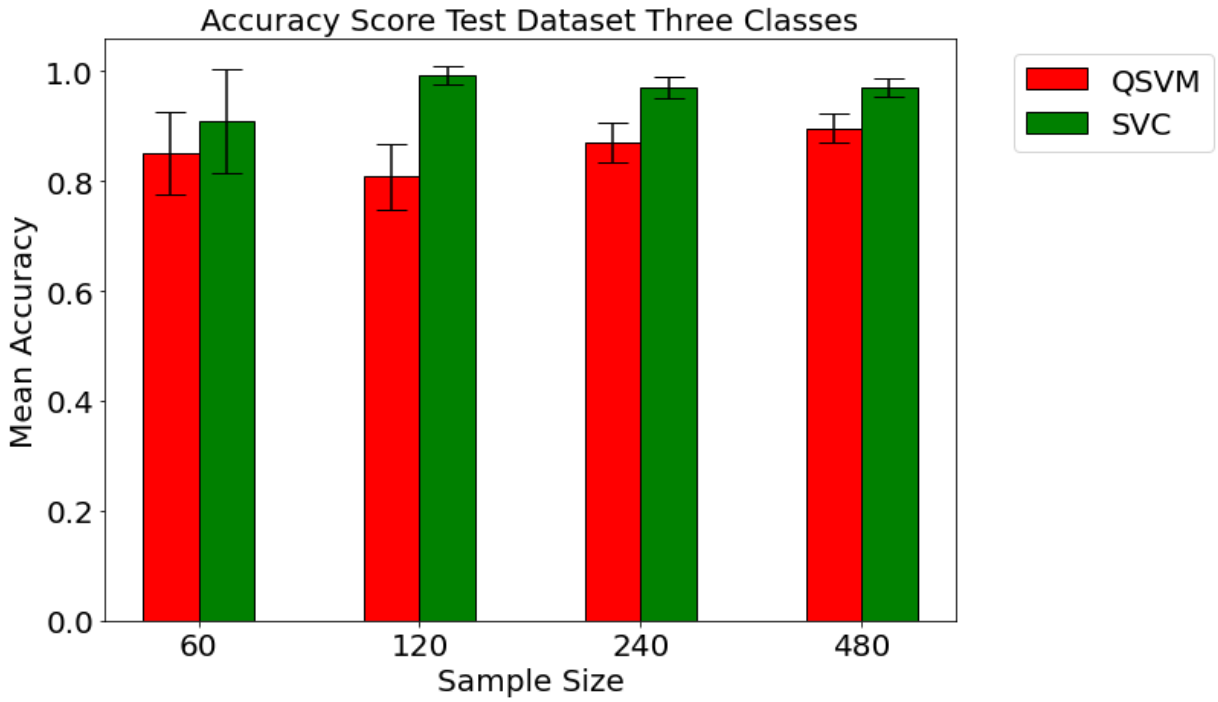

Figure 29: Recorded accuracy score on testing dataset for QSVM and SVC at the respective sample sizes. The models perform a multiclass classification between sandstone, shale, and carbonate. For each sample size, the models were run 10 times to record the standard deviation of the accuracy score. The black error-bar represents the standard deviation in the accuracy score**.**

In both train and test dataset, the QSVM model performs at a lower accuracy than SVC regardless of the sample size. Like in binary classification, QSVM does not improve the accuracy for multiclass classification. The quantum-enhanced machine learning model shows greater performance variability than the classical approach.

Lastly, the testing performance of QSVM was observed between binary and multiclass classification as shown in Figure 30. The quantum-enhanced model shows greater performance with binary classification as compared to multiclass classification regardless of the sample size.

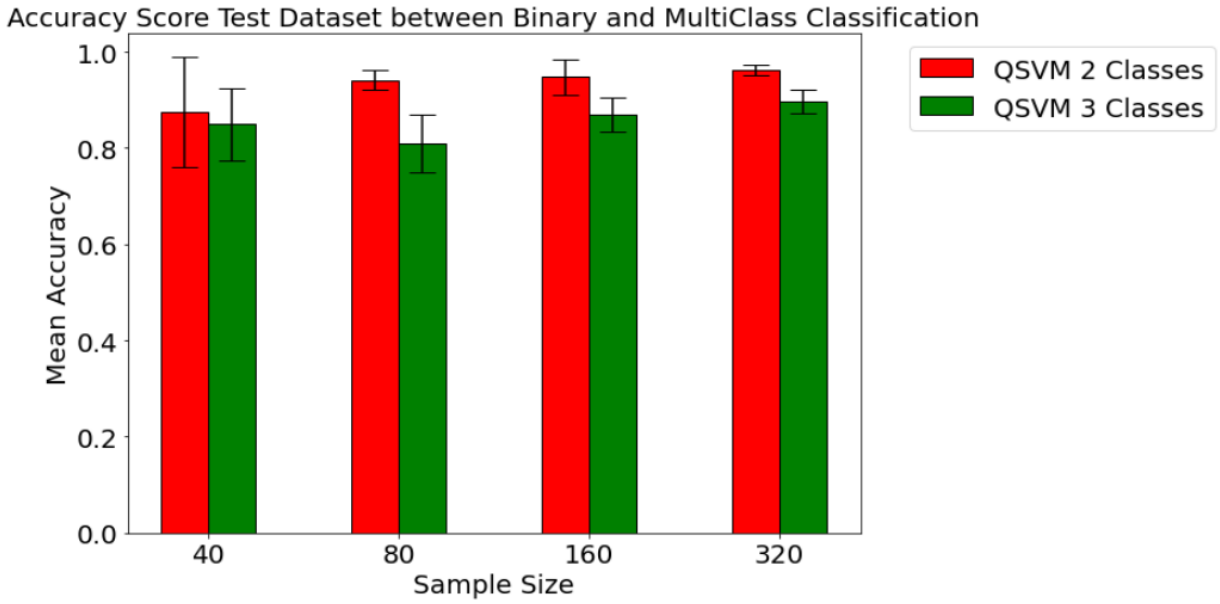

Figure 30: Recorded accuracy score on testing dataset for QSVM binary classifier and QSVM multi-class classifier at the respective sample sizes. For each sample size, the models were run 10 times to record the standard deviation of the accuracy score. The black error-bar represents the standard deviation in the accuracy score.

Run Time

An additional goal of this work is to compare the run time of initializing, fitting, and predicting the data between QSVM and SVC. As pointed by Havenstein, queuing times are very lengthy when accessing a quantum simulator (Havenstein et al, 2018). The same issue is experienced in this work when testing the QSVM model. Run times are exponentially longer when compared to a classical model. Figure 31 shows a comparison if running time between QSVM and SVC for binary classification. Run times increases exponentially for QSVM as sample size increases. On the opposite side, run time of SVC stays relatively constant. Future work can provide a better understanding of computational time if direct access to a quantum simulator is granted without any additional queuing times.

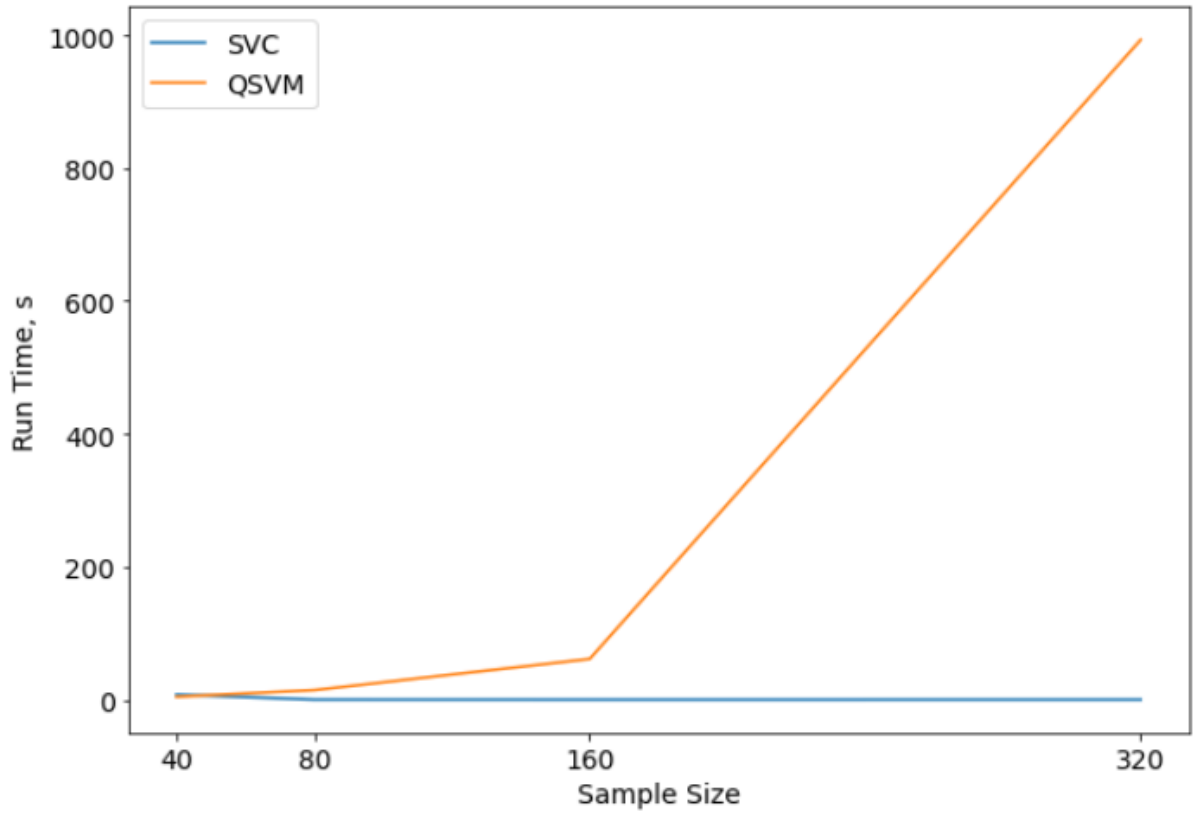

Figure 31: Comparison of run time between QSVM and SVC. The run time increases exponentially for the quantum model as sample size increases.

Answers to Fundamental Questions

- With the well log dataset analyzed in this chapter, an improvement in accuracy was not observed with kernel-based quantum-enhanced models. The well log dataset can be easily classified with traditional machine learning model and does not require a complex non-linear decision boundary.
- With smaller samples, QSVM underperforms when compared to SVC. As the sample size increases, the gap in performance between the two models becomes smaller. Sufficient sample size is necessary for QSVM to accurately classify lithology.
- A speed-up in computational time was not observed due to the queuing time necessary for accessing the quantum simulator

Conclusions

This work compares the performance between quantum and classical machine learning models in classifying lithology based on petrophysical well logs. Both methods are tested on binary and multiclass classification based on their mean accuracy. The results obtained demonstrate that there is no actual quantum advantage with the data considered. In both binary and multiclass classification, the classical model SVC outperforms the quantum model QSVM in both training and testing performance. The queuing time necessary to access a quantum simulator grows exponentially as the sample size of the data increases which makes this method impractical if no direct access is granted. Future work is necessary to assess the possibility of quantum advantage in other Petroleum Engineering problems.

CHAPTER III

COMPARISON OF CIRCUIT-BASED AND KERNEL-BASED QUANTUM-ENHANCED MACHINE LEARNING MODELS WITH ARTIFICIAL NEURAL NETWORKS FOR LITHOLOGY CLASSIFICATION FROM REAL-TIME DRILLING DATA AND PETROPHYSICAL LOGS

This chapter further explores the application of quantum-enhanced machine learning models for lithology classification. Variational quantum classifier and quantum support vector machine are trained and tested on binary lithology classification from petrophysical well logs and real-time drilling data. Both models are optimized based on their respective parameters. The performance and computational time are compared with a benchmark classical artificial neural network. Variation in decision boundary and performance metric distribution between the three models provide additional understanding on how quantum models differ from classical models.

Terminology

- Repetitions: the number of times the encoding feature map or variational circuit is repeated. The higher the repetitions, the more complex the model is. In variational circuit, increasing repetitions adds additional trainable weights to the model.
- Variational weights: like in neural networks nodes, weights in a variational quantum classifier iterate through multiple values based on the resulting cost function.
- Parity function: the process of converting quantum states back to classical information to derive an expectation value.

- Expectation value: probabilities associated with the occurrence of one sample belonging to a specific class.

Fundamental Questions

- Can quantum-enhanced machine learning models provide a greater performance in unbalanced datasets?
- Can a variational quantum classifier perform similar to a classical neural network using less parameters?
- Does increasing the number of qubits and repetitions improve the performance of quantum-enhanced machine learning models?

Novelty and Scientific Impact

- In a binary classification task, classical machine learning model draw a linear decision boundary whereas QML models draw a non-linear decision boundary
- Variational quantum classifier models have similar performance to artificial neural networks using a lower number of parameters on unbalanced datasets.
- Increasing the number of qubits results in greater performance of the variational quantum classifier whereas increasing the number of repetitions does not.
- QML models show a much greater variance in scoring parameters compared to classical models.

Literature Review

For implementation of variational quantum classifier (VQC) models, the present literature focuses mostly on synthetic data and particle physics applications as shown in Table 11. On average, the findings show similar performance between VQC and classical ML models.

Table 11: Literature review on applications of VQC models on synthetic and real-world datasets. No applications on subsurface characterization are found comparing VQC to classical methods.

Author	Goal	Results
Maheshwari et al, 2021	Binary classification on synthetic and diabetes datasets	VQC shows greater performance in synthetic datasets compared to real-data dataset. VQC performance is ls slightly lower compared to classical methods
Fuster et al, 2019	Classification of points in a circle	VQC shows same performance of ANN with lower parameters
Sierra-Sosa et al, 2020	Dementia prediction	VQC gives more consistent result compared to SVC with a linear kernel
Blance et al, 2021	Resonance search of di-top final state in particle physics	VQC outperforms ANN when trained on small amounts of data
Wu et al, 2021	Identification of rare signals in particle physics	VQC perform similarly to classical methods (SVC and BDT)

Introduction

The following chapter further explores the application of quantum-enhanced machine learning methods for binary lithology classification. Variational quantum classifier and quantum

support vector machine (QSVM) are tested and compared on two lithology classification datasets. A third model, a classical artificial neural network (ANN) is deployed as a benchmark reference for quantum-enhanced models.

Real-Time Drilling Dataset

The first dataset represents real-time drilling data from an offshore well in the North Sea which is part of the Volve open-source database. The information provided includes a total of 30 measurements at 270 specific depths obtained during the drilling process. Currently, lithology is predicted from a geological interpretation of drill cuttings. The use of data-driven machine learning methos for lithology prediction can aid to geological interpretation and ultimately benefit drilling engineers to identify the target formation. For each measurement taken every 5 meters, a lithology class is present from core analysis. Figure 32 shows the data distribution colored by lithology after reducing the dimensionality to 2 principal components. The dataset is severely underbalanced when comparing the different lithology groups. The sandstone class represents majority of the sample measurements.

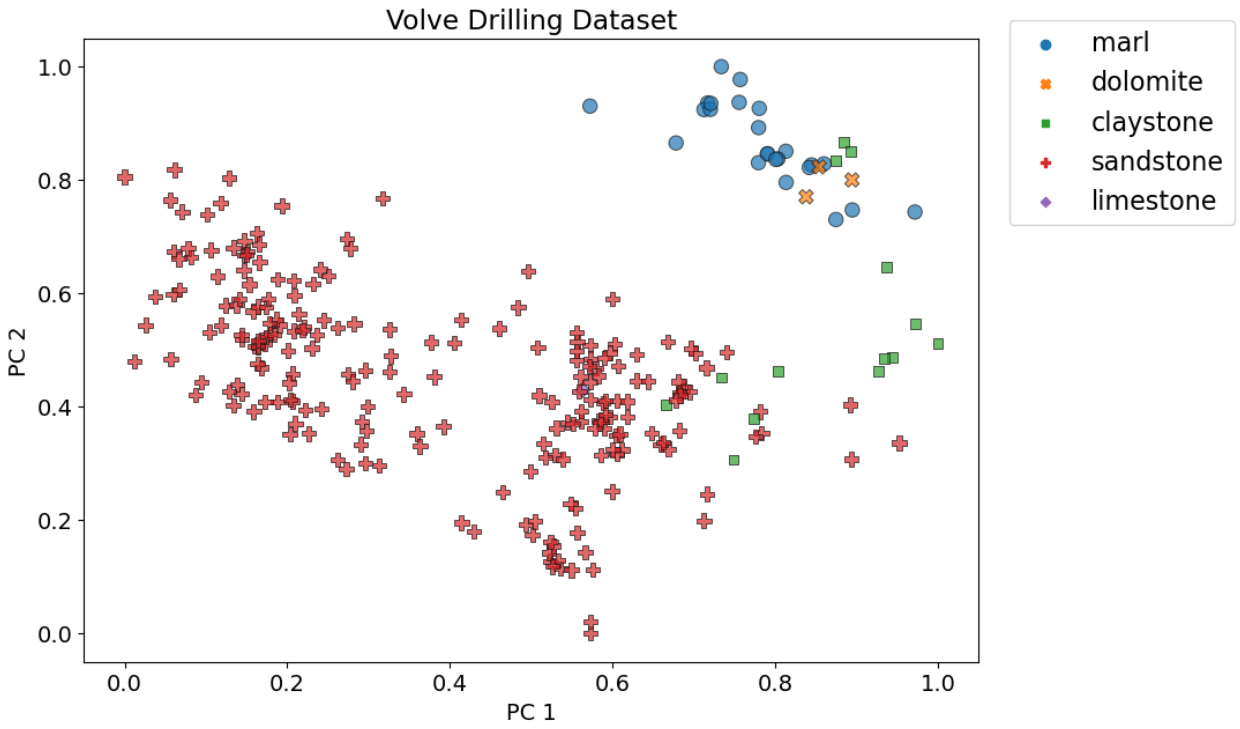

Figure 32: Sample distribution of the real-time drilling dataset of one well in the Volve field. The data is scaled and reduced to 2 principal components for visualization. The classes are severely unbalanced as sandstone samples represent majority of the dataset.

Well-Log Dataset

The second dataset implemented is the petrophysical well log data discussed in Chapter 2 . The same data preprocessing steps discussed in the previous chapter are applied to remove outliers and reduce the feature dimensionality. The objective for this dataset is to evaluate and compare quantum and classical models on the entire sample size. The data includes gamma ray, density, neutron, and resistivity logs for two wells in the Volve field for a total of 1600 samples.

Machine Learning Models

Artificial Neural Network

Artificial neural network (ANN) is a classical machine learning method used in both regression and classification tasks. The model is made of a collection of layers containing multiple nodes connected to each other. The input layer represents the initial data while the output layer represents the final result. For each node, the hidden layers located between the input and output layers, transform the input sample from one dimension to another using a non-linear activation function. Each node has a randomized initial weight associated with it. The activation function takes the weighted sum of all the inputs and calculates the next node for the subsequent layer. The final output node is then compared to the actual label using a cost function. Based on the resulting cost function, the weights of the ANN are updated using an optimizer model. The process of calculating the output from current weights is referred to as forward pass whereas the process of deriving each weight contribution to the error and optimizing the weights is referred to as backward pass. The epoch represents one forward pass followed by one backward pass. This model iterates through multiple epochs until it obtains the best combination of weights that provide the lowest cost function. In this work, a sequential ANN is built using the keras library. The model consists of initial input layer, three hidden layers, and one output layer. The activation function ReLU is used for the hidden layers whereas the Sigmoid function is used for the final output layer.

- ReLu: piecewise linear activation function which outputs the weighted sum of the previous layer's nodes if positive or 0 if negative.
- Sigmoid: function which outputs a value between 0 and 1 which represents the probability of the sample belonging to a specific class.

Overall, with the specified configuration, the ANN model is built on 381 trainable parameters.

Variational Quantum Classifier

The Variational quantum classifier (VQC) combines the same architecture of ANN with quantum computations. Much like the QSVM method discussed in Chapter II, VQC is a hybrid quantum-classical machine learning model used for classification tasks. Figure 33 shows the architecture of VQC from the initial input data to the final label prediction. As discussed in Chapter II, the feature map encodes the information of the initial data into a quantum state by modeling the qubit rotation. VQC takes this process a step further by implementing a variational circuit which consists of a quantum feature map with additional weights applied to the angle rotations and entanglement layers. The model is trained in an iterative process in which the weights are updated by an optimizer model based on the resulting cost function. The model completes the training procedure once the lowest cost is achieved.

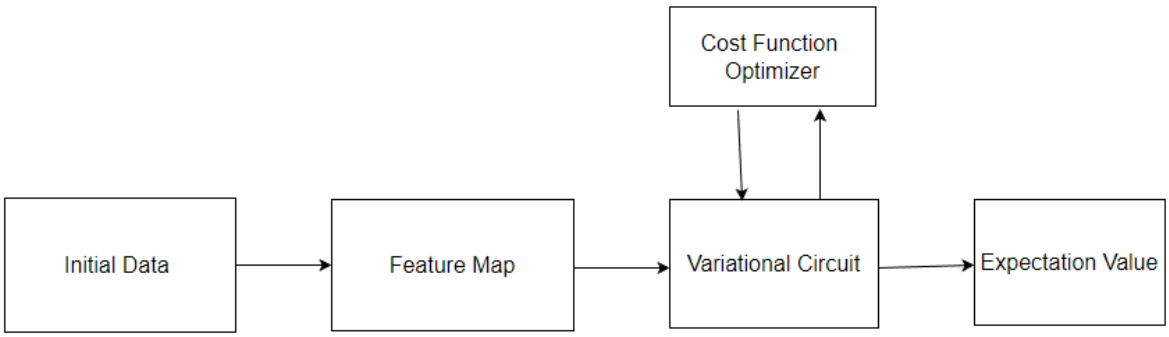

Figure 33: Structure of VQC. The initial input data is encoded into a quantum state with the use of a feature map. A variational circuit assigns different weights to the angle rotations of the qubits in the feature map. The weights are updated by an optimizer model based on a cost function.

The variational circuit represents the equivalent of a collection of layers in a neural network. The circuit expressed in Equation I applies weights to the qubit rotation and entanglement layers defined in the feature map .

$$\mathbf{U}(\mathbf{x}, \mathbf{w}) = \mathbf{S}(\mathbf{x})\mathbf{W}(\mathbf{w}) \qquad \textbf{Equation 5}$$

S represents the quantum feature map which encodes the classical data x into a quantum state. W represents the variational component which applies the weights w to the feature map. The variational circuit U can then be expressed in terms of classical input data x and weights value w. An additional redout qubit is added to apply a weight to the entanglement layer. The number of weights is dependent upon the structure of the feature map and the number of repetitions. The resulting variational circuit using a feature map with two qubits and linear entanglement employs 13 trainable weights as shown in Figure 34.

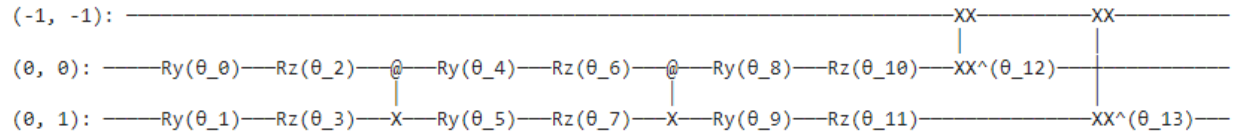

Figure 34: Structure of a variational circuit with two qubits and linear entanglement. @-X represents the entanglement layers. The redout qubit XX applies a weight on the qubit entanglement. Rotations respect to the y and z axis are performed depending on the weight applied.

The model translates the quantum output to a classical value by the parity function. The quantum output can be expressed as a probability distribution of each possible qubit state 2^n where n is the number of qubits. When using two qubits, the probability distribution of the qubit state can assume values of 00,01,10,11 with their respective associated probability. Even combinations such as 00 and 11 represent the 0 class and odd combinations 01,10 represents the 1 class. The associate probabilities from the probability distribution are summed to compute expectation

values depending on their even or odd state. Figure 35 represents an example of how expectation values are derived in a VQC for a binary classification task using 2 qubits.

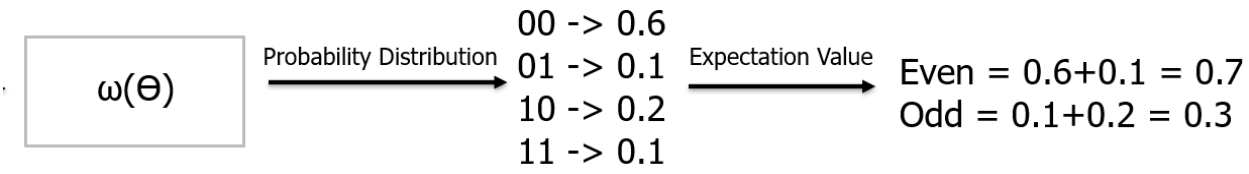

Figure 35: Example of expectation value derivation in a VQC. The output result from the variational circuit represents a probability distribution of the qubit states. Based on the even or odd class, the expectation value is derived by the sum of the probabilities associated with each class.

When comparing their structure, ANN and VQC are very much similar. Both models implement a "black box" format where multiple weights are trained and updated depending by on the specified cost function. In recent years, VQC has been researched for potential application in classification tasks and has often been referred to as quantum neural network. The premise behind the development of VQC is in its ability to produce accurate results with a lower number of weights which translates into lower computations.

Applicability of VQC

In the literature, VQC models have been tested on both synthetic and real-world dataset. Blance compares the efficiency of VQC and NN in classifying di-top final states in particle physics (Blance et al, 2021). The dataset is made of 2500 samples in a two-dimensional feature space. VQC shows a slightly greater boost in performance compared to ANN. Maheshwari conducts a similar comparison on the open-source diabetes and sonar datasets available in the python scikit-learn library (Maheshwari et al, 2021). The larger dataset (diabetes) shows greater performance. The addition of more qubits improves the average score of the quantum-enhanced

model. From the literature observed, VQC tends to perform better in smaller and higher dimensions datasets.

Cost Function

Binary Cross-Entropy

For each epoch during training, the ANN and VQC models predict the probability of a sample belonging to either one of the two classes. A cost function is required to define the accuracy of the prediction. The resulting cost determines the change in the models' weights for each epoch. The model training stops once the lowest cost value is achieved. Binary cross entropy represents the negative log average of a sample belonging to its original class as shown in Equation 6. The metric ranges from 0 to 1, where a value of 1 is only achieved if the predicted and true labels are identical.

$$\mathbf{B_{c\_e}} = -\frac{\mathbf{1}}{\mathbf{N}}\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{y_i} * \mathbf{log}\big(\mathbf{p}(\mathbf{y_i})\big) + (\mathbf{1} - \mathbf{y_i}) * \mathbf{log}\big(\mathbf{1} - \mathbf{p}(\mathbf{y_i})\big) \qquad \textbf{Equation 6}$$

- N = total number of samples
- y = class, either 0 or 1
- p(y) = probability of the sample belonging to the specific class

Optimizer

Adam

For both ANN and VQC, the training procedure requires constant update of the weights after each iteration to minimize the cost function. The optimizer represents the algorithm that modifies the weights of the machine learning model according the cost function value obtained

from the previous iteration. The proper selection of the optimizer method is necessary to reach the combination of weights that result in the lowest cost function at a low computational time. Figure 36 shows a simple representation of how an optimizer preforms. Gradient vectors are derived to determine the direction of the steepest descent. The algorithm modifies the input weights according to a specified learning rate until the optimal minimum is achieved.

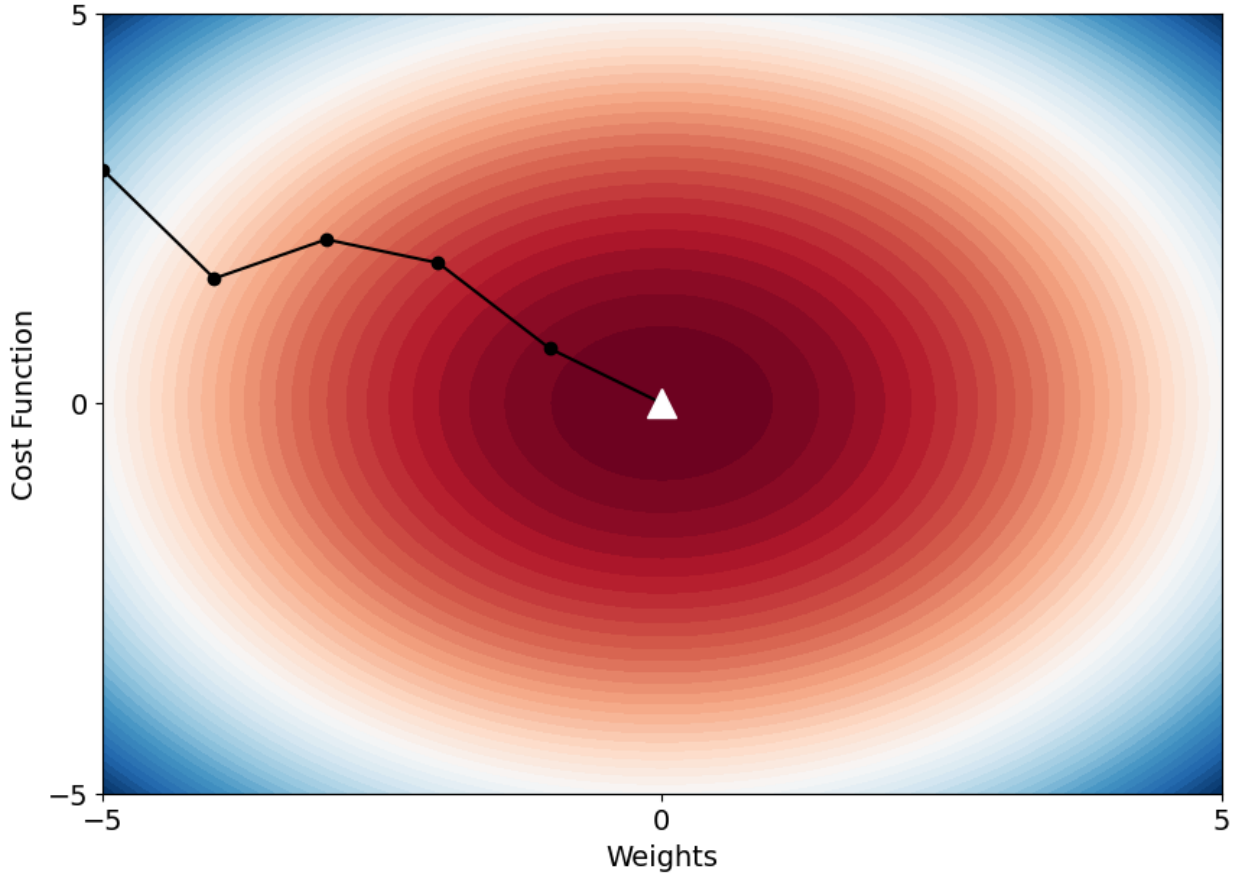

Figure 36: Example of an optimizer algorithm. The gradients are represented by the black segments connecting the points. The optimal minimum is represented by the lowest cost function as a white triangle. The algorithm modifies the weights and learning rates until the lost cost function is reached.

The adaptive moment estimator (ADAM) optimizer has been recognized to be extremely effective when implemented in neural networks models (Kingma et al, 2014). This method calculates the gradients based on batches from the initial sample pool. Unlike other optimizer,

this method updates the learning rate of each weight in the neural network individually. The weights are updated based on the weighted and moving average of the gradients. This functionality allows the ADAM optimizer to continue the learning process when passing through a local minimum until the absolute minimum is reached.

Methodology

The proposed workflow is presented in Figure 37. The goal is to compare the lithology classification performance between classical and quantum models in a real-time drilling and well log datasets. VQC and QSVM are compared to a classical ANN model. As previously mentioned, VQC represents a variational circuit-based QML model which implements an optimization method that variates the circuit weights to achieve the lowest binary cross-entropy. QSVM represents a kernel-based method in which the distances between each point are computed in a kernel matrix. The kernel matrix is then passed to a SVC model that computed the support vectors for classification. The raw data coming from real-time drilling logs is preprocessed to eliminate outlier values and scaled to the same magnitude. Feature selection is performed on the original 30 features to identify the ones that do not provide sufficient information. ANN, VQC, and QSVM models are initialized and optimized accordingly based on different hyperparameters. The models are then run on 100 different train- test split and the median and standard deviation are recorded. Similar procedure is performed using the preprocessed petrophysical well log data in Chapter II. Additionally, the performance results of quantum models are compared using 2 and 3 qubits to understand how increasing model complexity improves classification performance. Comparison of performance and training

computational time give insights of future implementation of QML models for lithology classification.

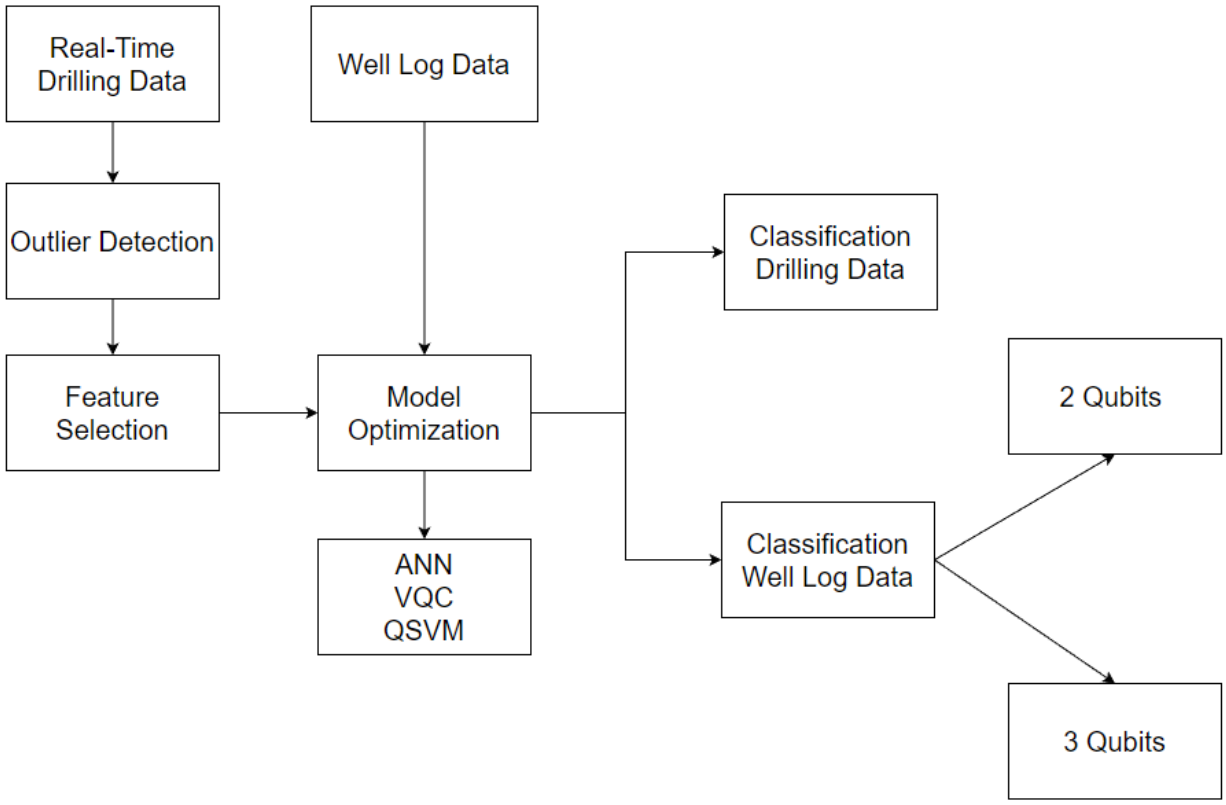

Figure 37: Workflow applied for lithology classification on both drilling and well log datasets. The real-time drilling dataset is preprocessed first whereas the well log dataset is already preprocessed form Chapter II. ANN, VQC, and QSVM are tuned and compared on performance and computational time.

Data Preprocessing

The original drilling log dataset contains 5 different lithologies for a total of 270 samples. The lithology classes are severely underbalanced with sandstone being the most prevalent as shown in Figure 38. For this work, binary classification is tested. The goal is to identify if the lithology presented is a sandstone or not. Therefore, the labels are redistributed such that samples of sandstone have a value of 0 and samples other than sandstone have a value of 1. Even with

this layout, there is still a great disparity in sample size between the two classes. To compensate for this, classification metrics such as F1 weighted score and Matthew's correlation score are used to better evaluate the performance of the classical and QML models.

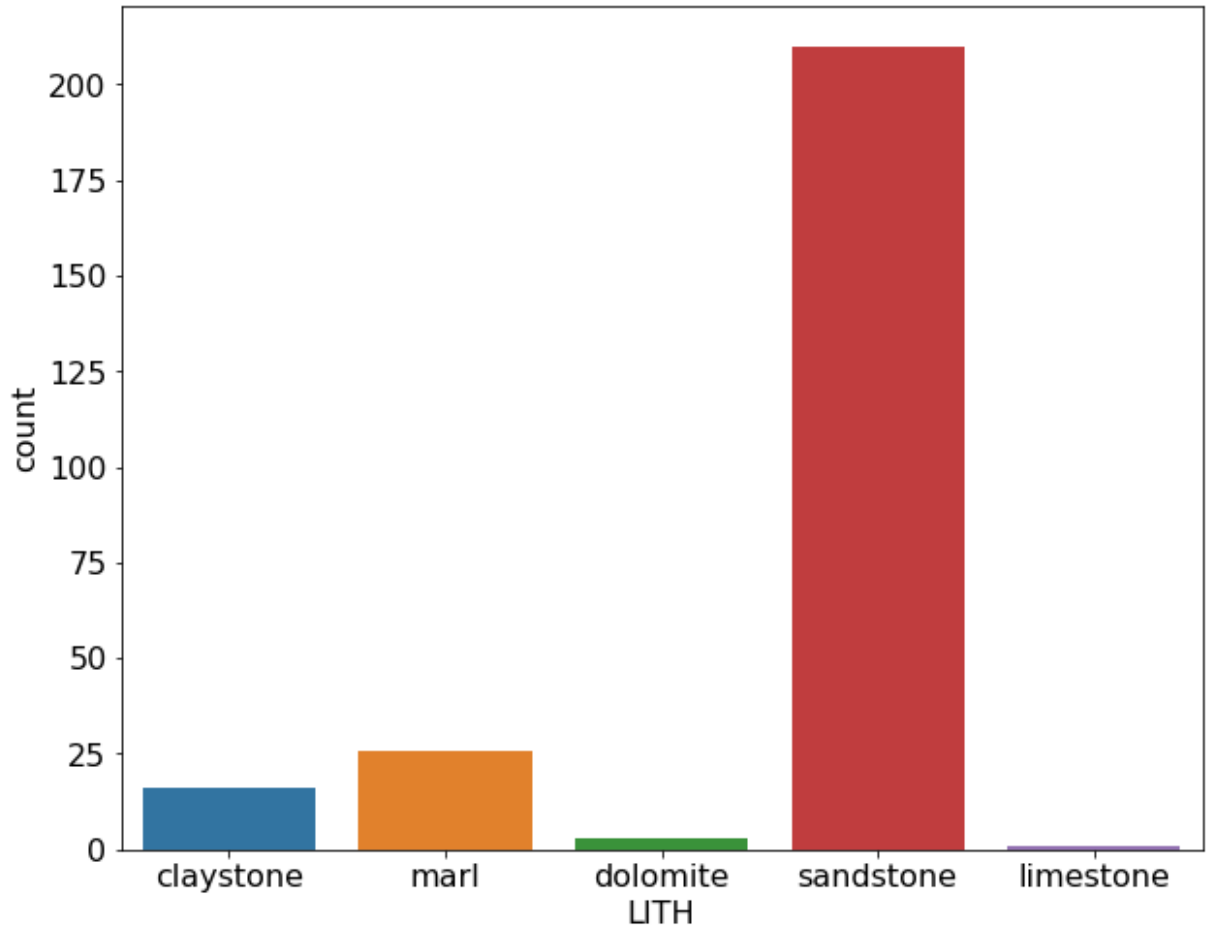

Figure 38: Lithology frequency of the drilling log dataset. The lithology classes are significantly unbalanced. For binary classification, the sandstone samples are marked as 0 and the non-sandstone samples are marked as 1.

Train Test Split

Before applying any preprocessing techniques, the data is divided into train and testing set with a 70/30 ratio. This procedure ensures no bias is transferred to the machine learning models from the test set. The quantum and classical machine learning models are built and

trained entirely on the train set. The models come in contact with the test set for the first time only when evaluating the performance. As described in Figure 35, the classed in the drilling logs dataset are severely unbalanced. The split between train and test is stratified to ensure that the same proportion of samples in each class is preserved.

Isolation Forest

After splitting the original feature set, the first step of the workflow is designed to remove outlier values from the original dataset. Like in previous chapters, isolation forest is implemented to detect and remove outliers in the data. Due to the small sample size of 270, a small contamination value of 0.05 is used to honor the original drilling log values and prevent the sample size from dropping significantly. With the implementation of isolation forest, 14 samples were detected as outliers and removed from the dataset.

Standard Scaler

Feature scaling is necessary to optimally apply feature selection techniques such as principal component analysis. Standard scaler is used to equalize the features to a zero mean and a standard deviation of 1.

Variance Threshold

Due to the high number of features, selection techniques are deployed to reduce the feature space dimensionality. Variance threshold helps identify features with zero variance that do not present any useful information for the machine learning model. Using a small threshold of 0.01, 9 out of the original 30 features are omitted from the dataset due to their low variance.

Pearson's r correlation

Features with high collinearity do not contribute at improving the model performance. Pearson's r correlation value is used as the score metric to evaluate the collinearity between each pair of features in the dataset. As shown in Figure 39, a matrix of correlation is generated to visually locate the features with the highest collinearity. The correlation matrix is symmetrical with the diagonal having values of 1. Therefore, for purpose of visualization, only the lower triangular portion is shown. A threshold of 0.9 is imposed on the Pearson's r value. For each couple of features, if the absolute r value exceeds 0.9, one of the two features is omitted from the dataset. Following this procedure, 8 features are removed from the dataset due to high absolute collinearity.

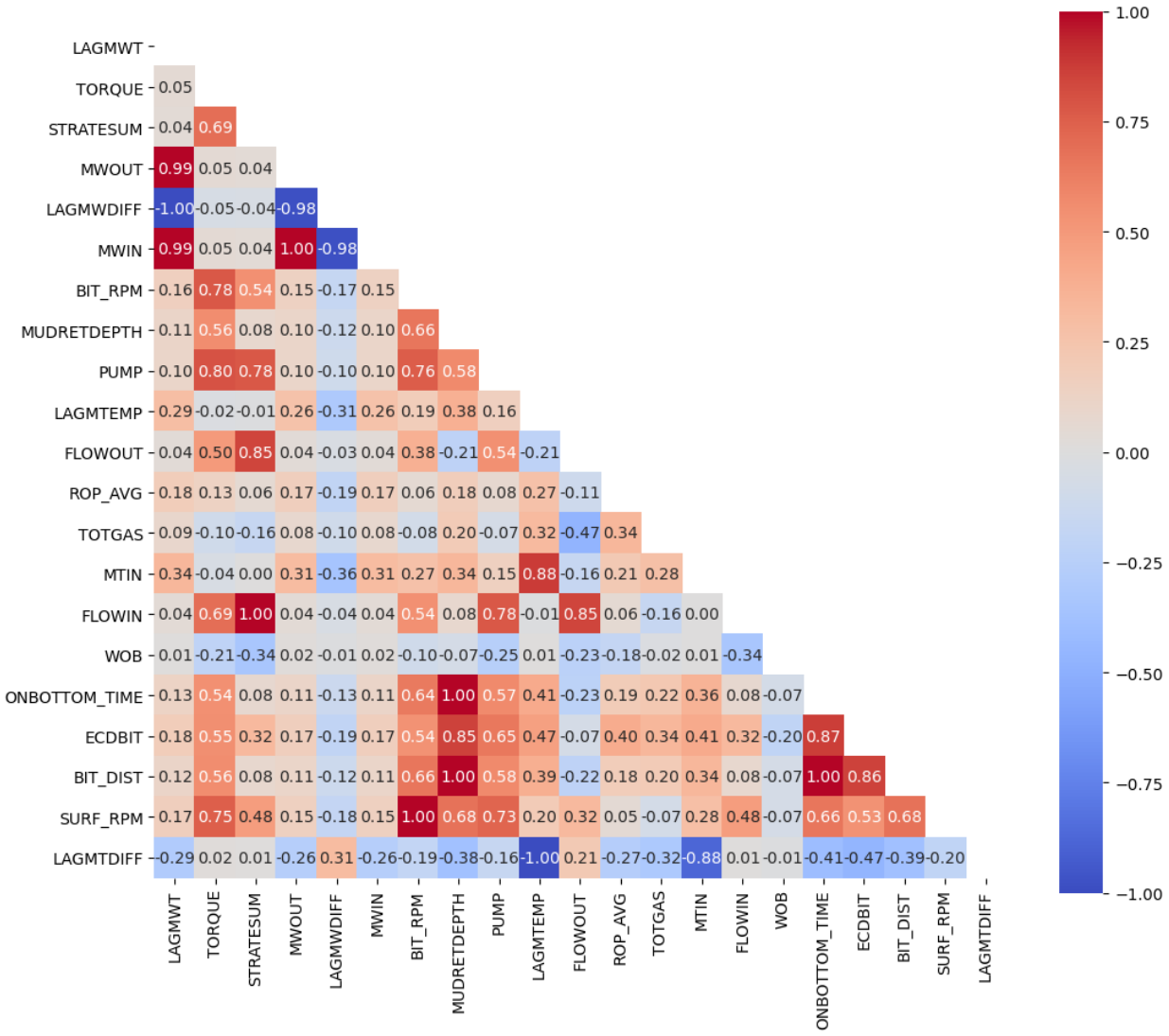

Figure 39: Person's r correlation matrix helps identify highly collinear features in the dataset. Red cells show high positive collinearity whereas blue cells show high negative collinearity. A total of 8 features are removed due to high absolute collinearity.

Visualizing the Data

Before proceeding to further preprocessing techniques, each feature is visually analyzed on a x-y plane. This helps identify any features that do not provide any information a machine learning model. From the subplots shown in Figure 40, there are 4 features that remain constant or linearly decrease throughout the depth interval. Mudretdepth measures the depth of the mud

which is equal to the depth recorded. Since the purpose of the machine model is to predict lithology regardless of the depth interval, this linear relationship is redundant. Lagmwdiff and lagmtemp measure the mud weight difference and mud temperature respectively. Both measurements reach a constant value of 0 only after 5 meters deep in the drilling process. The same behavior is observed in mtin which has a zero value for the majority of the dataset. The resulting number of selected features is 9.

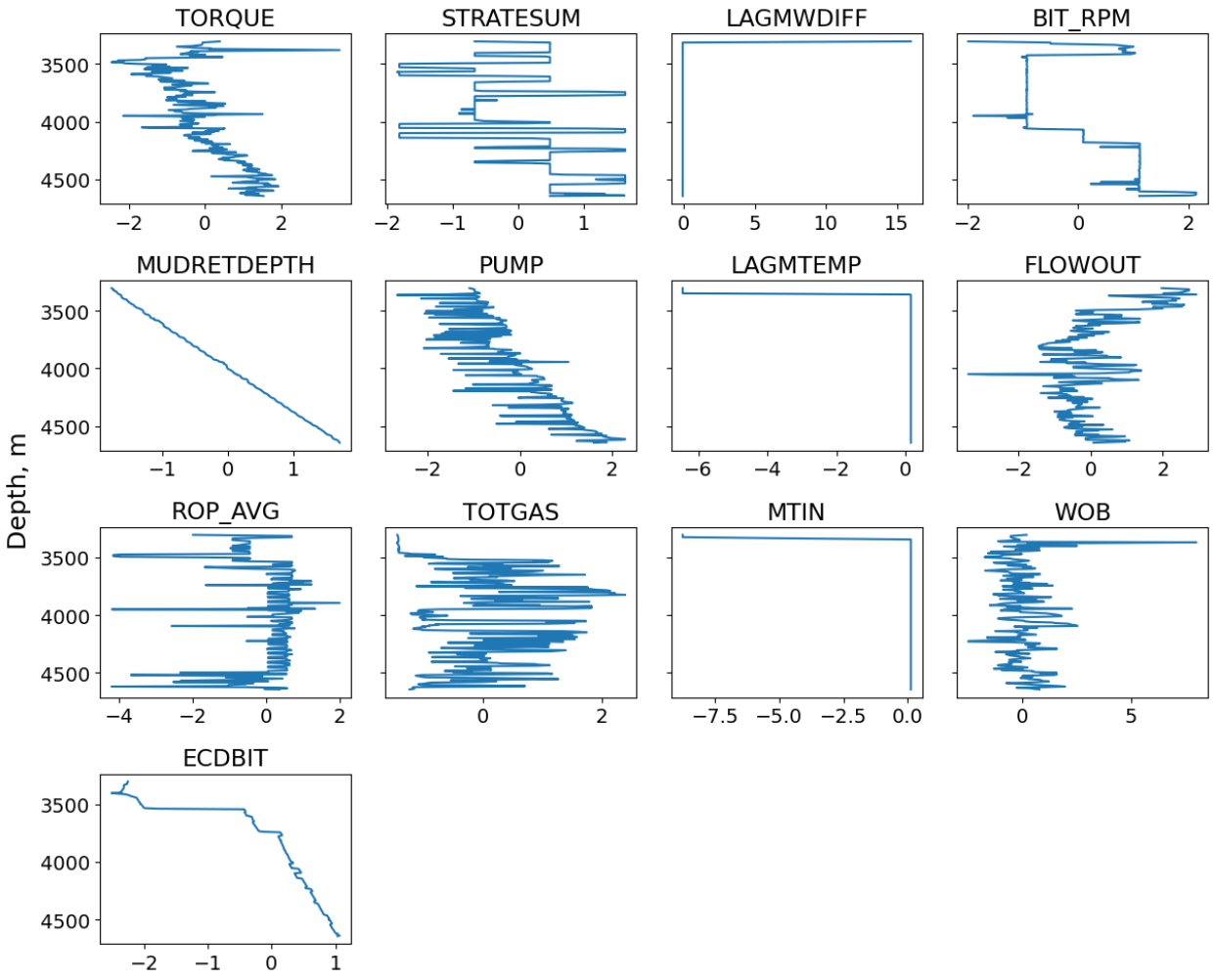

Figure 40: Visual representation of real-time drilling data as function of depth. Outlier detection and feature scaling is applied beforehand. Mudretdepth is linearly related to formation depth. Lagmwdiff, lagmtemp, and mtin have a constant value of 0 for the majority of the dataset.

Principal Component Analysis

The last step for data preprocessing is principal component analysis. As mentioned previously, the number of qubits is represented by the number of features in the dataset. For the real-time drilling dataset, the quantum models are tested with 2 qubits. As a result, the preprocessed drilling dataset is reduced to two principal components. The variance explained by each principal component is recorded to quantify the potential information loss due to the reduction of the feature space. Figure 41 shows the percentage of variance explained as function of the number of principal components. Unlike the petrophysical well log data observed in Chapter II, the variance explained with 2 principal components in the drilling dataset follows below 70%. Although a low percentage of variance explained decrease the machine learning model performance, the goal of this chapter is to create a benchmark between classical and quantum ML models. In both cases, the models are trained and tested on the same data with 2 principal components.

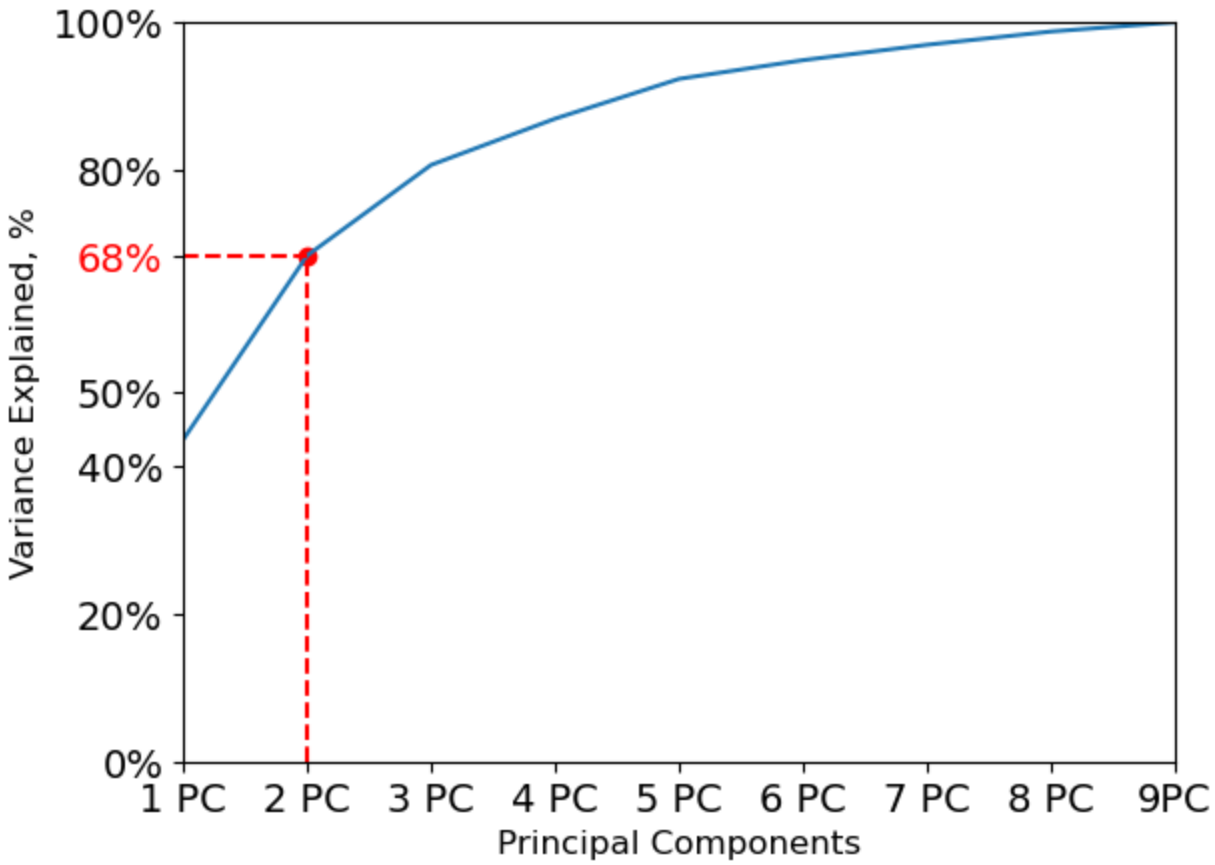

Figure 41: Percentage of variance explained by number of principle components during PCA. Using 2 principal components explains 68% of the variance in the data. Both classical and quantum models are trained and test on the same dataset with 2 principal components.

Model Optimization

Like traditional machine learning models, in QML models, parameter tuning is necessary to achieve optimal performance. As observed in Chapter II, quantum models achieve optimal performance when using feature maps that model linear entanglement between qubits such as the second order Pauli feature map (ZZ). With QSVM, the number of repetitions and regularization parameter C are varied. For VQC, the tuning is performed on the variational circuit and optimizer model. Much like a neural network, the variational circuit is trained on a different

number of weights depending on the number of repetitions and qubits. A linear relationship is observed in the number of trainable weights when varying the number of repetitions and qubits as show in Figure 42. When keeping the number of qubits constant, the total count of weights increases linearly as more repetitions of the circuit are present. Similarly, as the repetitions are kept constant, the number of weights increases linearly as more qubits are used. Overall, the number of qubits has a stronger impact on the total number of trainable weights.

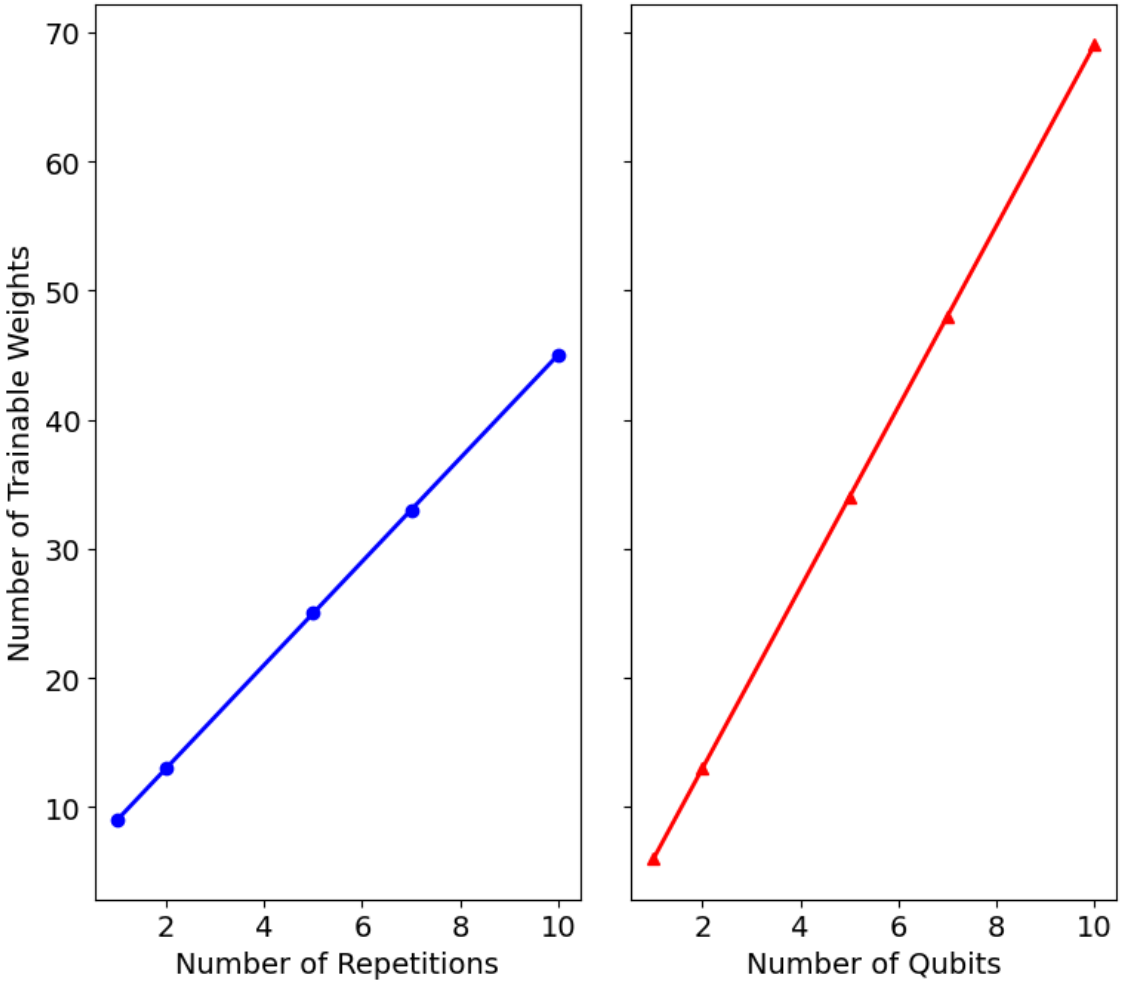

Figure 42: Number of trainable weights in a variational circuit as function of number of repetitions (left) and number of qubits (right). On the left plot, the number of qubits is kept constant at 2. On the right plot, the number of repetitions is kept constant at 2. In both cases, a linear relationship is observed.

When increasing the number of repetitions, the depth or complexity of the variational circuit increases linearly. The trainable weights associated with each qubit rotation are repeated in sequence. Model complexity can also be increased with a larger number of qubits. However, an additional qubit introduces weights that model the new qubit's rotation and entanglement with the rest. In summary, additional repetitions increase the number of weights by copying the existing circuit layout whereas additional qubits introduce new rotation and entanglement weights. Unlike adding more repetitions, increasing the number of qubits inherently adds more entanglement layers since more qubits are present. As a result, the rate of increase in complexity is significantly higher when varying the number of qubits as compared to the number of repetitions.

Tuning is performed on both the classical and quantum ML models for the two datasets. For the drilling dataset, the number of qubits is kept constant at 2 whereas for the well log dataset, the qubits count is varied between 2 and 3. The goal is to observe if a greater number of qubits improves model performance. In addition, VQC and ANN are tuned on the Adam learning rate and batch size. Lastly, for the classical neural network, different combinations hidden layer activation functions are tested. After iterating through multiple combinations, the optimal values are determined based on the resulting F1 score as described in Table 12.

Table 12: Model tuning for ANN, VQC, and QSVM. The models are tuned on both the drilling and well log datasets. The optimal values are determined based on the resulting F1 score.

Model	Parameters	Range	Optimal Value	
			Drilling Dataset	Well Log Dataset
ANN	Activation function	reLU, tanh	reLU	reLU
	Learning Rate	0.001-0.1	0.01	0.1
	Batch size	5-100	5	50
VQC	Repetitions	1-10	3	2
	Learning rate	0.001-0.1	0.01	0.01
	Batch size	5-100	5	10
QSVM	C	100-1000	1000	800
	Repetitions	1-10	2	2

From the results obtained on the VQC, increasing the number of repetitions does not translate in greater performance. Increasing the number of repetitions adds additional weights from copies of the existing variational circuit. This procedure increases the model complexity but does not lower the resulting cost function during training.

Discussion and Results

After the classical and quantum models optimization, the models are tested to evaluate and compare the performance. Lithology classification is performed on both real-time drilling data and well log data. In both datasets, the models are evaluated in their ability to accurately

classify the lithology from two classes for each depth interval. The decision boundary is visualized by testing each model on the feature space matrix. Since each sample contains two features ranging from 0 to 1, a 2D matrix can be generated with a predetermined dimension. For this work, a 150x150 matrix contains different combinations of the two features. By testing the models on each matrix element, it is possible to visualize a decision boundary. For real-time drilling data, due to the highly unbalanced classes, F1 weighted score and MCC are used as the performance metrics. For the petrophysical well log data, train/test accuracy and F1 score are recorded. The models are trained and tested on 100 different combinations of train and test data. For each iteration, the original dataset is divided into train and test sets and the previously described preprocessing methods are applied to that specific split. This procedure generates a distribution of score metrics. Due to the stochastic nature of machine learning models, probability distributions give a more comprehensive insight on their performance.

Lithology Classification from Real-Time Drilling Data

The dataset contains 270 measurements obtained while drilling for every 5 meters. For each depth interval, core evaluation is performed in the lab to label the sample's lithology. The data is divided into sandstone and non-sandstone samples with the sandstone class being the most prevalent. The data is divided into train and test sets and multiple preprocessing techniques are applied to eliminate outlier values and reduce the feature space to two dimensions. After optimizing the models parameters, ANN and VQC are compared on their training performance. Both models are run on 100 epochs with an EarlyStopping callback function. The function halts the training procedure if binary cross entropy does not decrease after 10 epochs. Figure 43 shows the changes in binary cross entropy as the training progresses after each epoch. ANN reaches the

lowest cost function at a faster rate than VQC. In both models, the callback function stops the training procedure before the 100 epochs are completed. The lowest binary cross entropy achieved during training is relatively similar in both model with ANN showing a slightly lower cost than VQC.

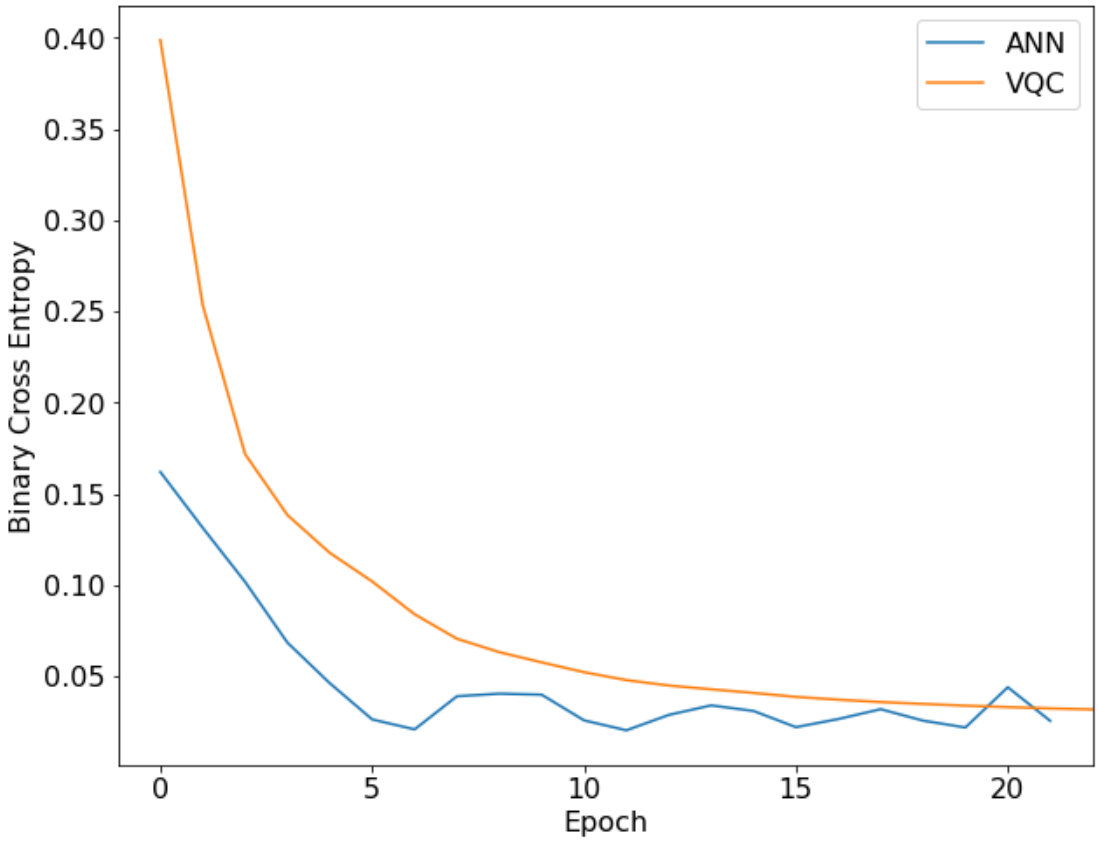

Figure 43: Comparison of binary cross entropy loss throughout each epoch between ANN and VQC in lithology classification of real-time drilling dataset. The callback function stops the training procedure after 21 epochs (ANN) and 22 epochs (VQC). ANN reaches a slightly lower cost function than VQC.

After training, ANN, QSVM, and VQC are evaluated on their classification performance on the testing set. Figure 44 shows the decision boundaries for each model. The test dataset labelled by class is placed on top of the feature space matrix ranging from 0 to 1. Both ANN and QSVM draw a linear decision boundary to classify the data. On the other hand, the shape of

VQC's decision boundary mimics a sinusoidal function. Due to the low-test size, the number of false positives and negatives can be quickly assessed. ANN shows the best performance with only 2 false negatives. VQC has a lower number of false negatives but higher false positives. Lastly, QSVM shows the poorest performance with 3 false negatives and 2 false positives.

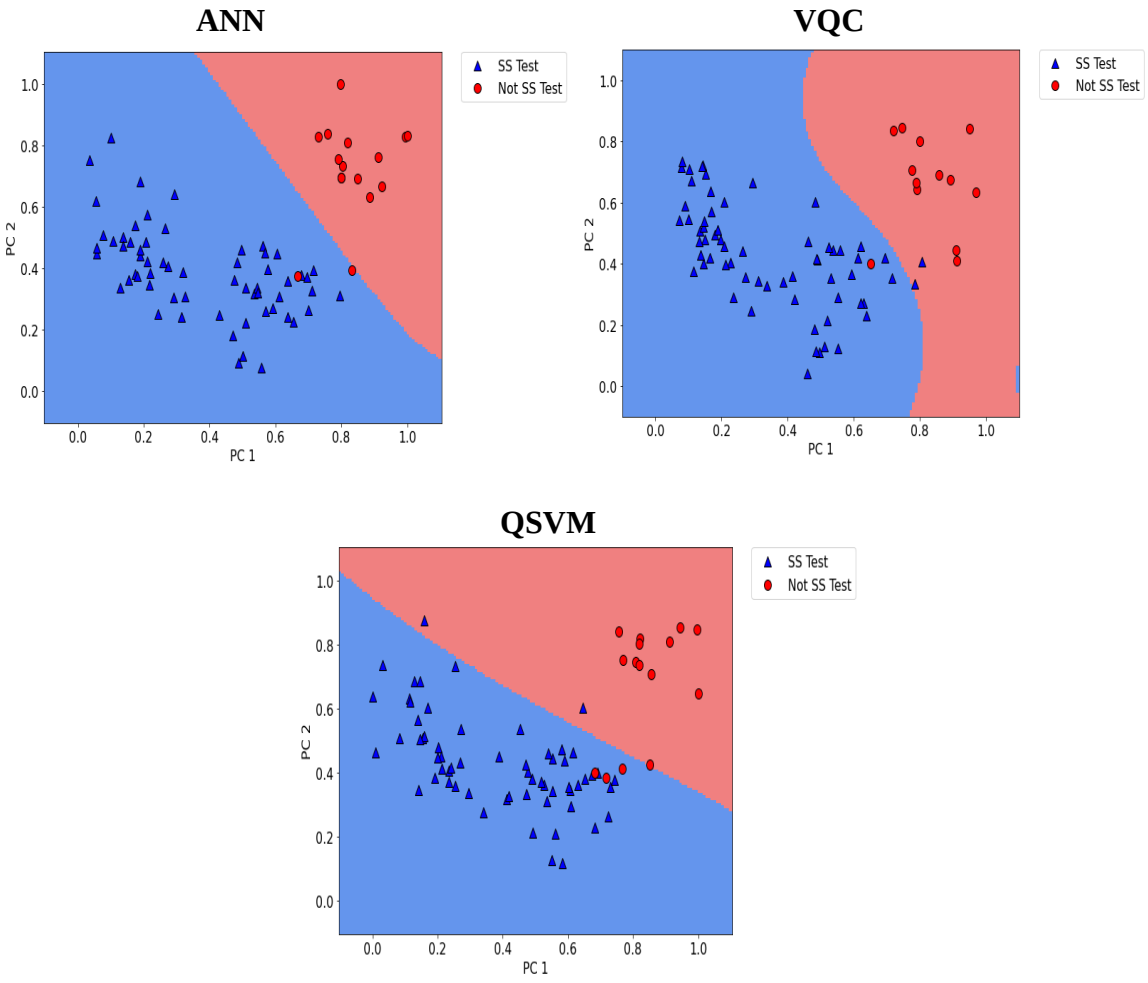

Figure 44: Decision boundaries of ANN (top left), VQC (top right), and QSVM (bottom center) on lithology classification of real-time drilling data. When analyzing the number of false negative and false positives, ANN shows the greatest performance followed by VQC and QSVM.

To concretely evaluate and compare the performance, F1 weighted score and MCC score are computed from the testing results. The performance metrics are recorded 100 times for each

train-test split. Figure 45 and Figure 46 show the distribution of F1 weighted and MCC scores respectively in a boxplot format. In the plot, the box represents the IQR defined by the difference between the 25^{th} and 75^{th} percentile. The error bars range 1.5 times the IQR where any outlier point outside the range is shown as a circle. For both metrics, ANN and VQC have nearly identical median value. As seen in the decision boundary analysis, QSVM shows the poorest performance with a large number of outliers outside the interquartile range. Compared to the classical ANN, quantum models exhibit higher variability in the results.

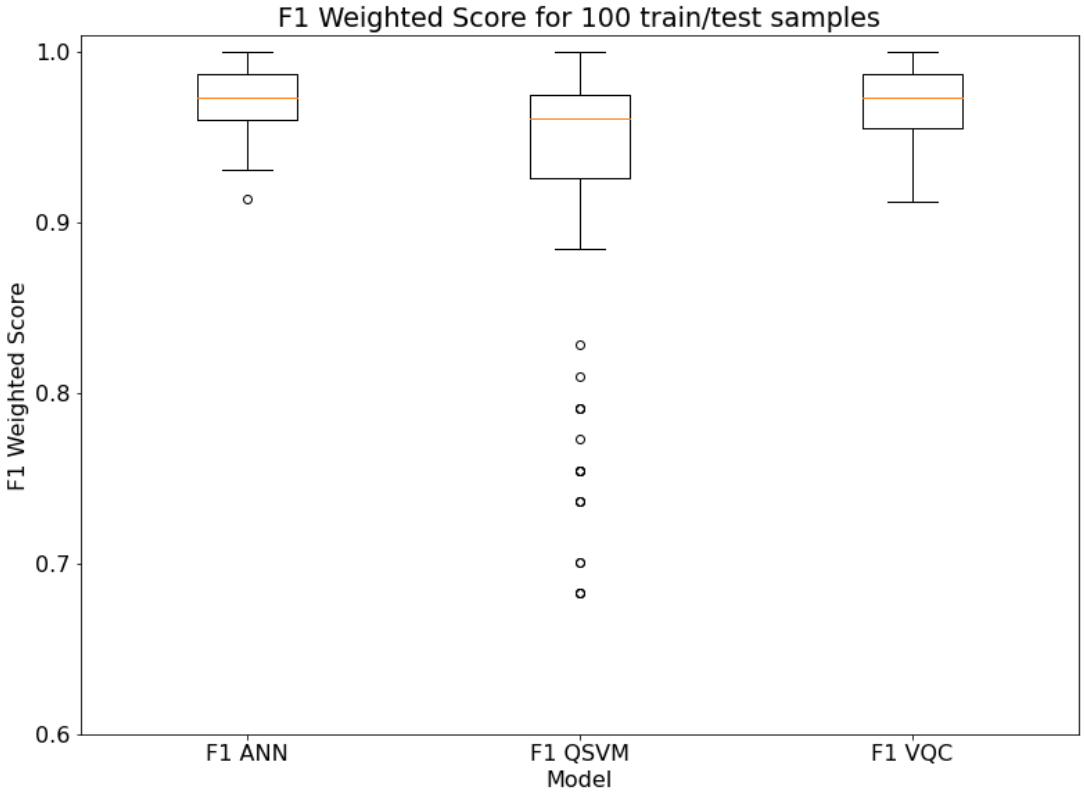

Figure 45: F1 weighted score distribution between ANN (left), QSVM (center), and VQC (right). The orange line represents the median of the distribution. ANN and VQC show very similar performance. QSVM displays a greater number of outliers.

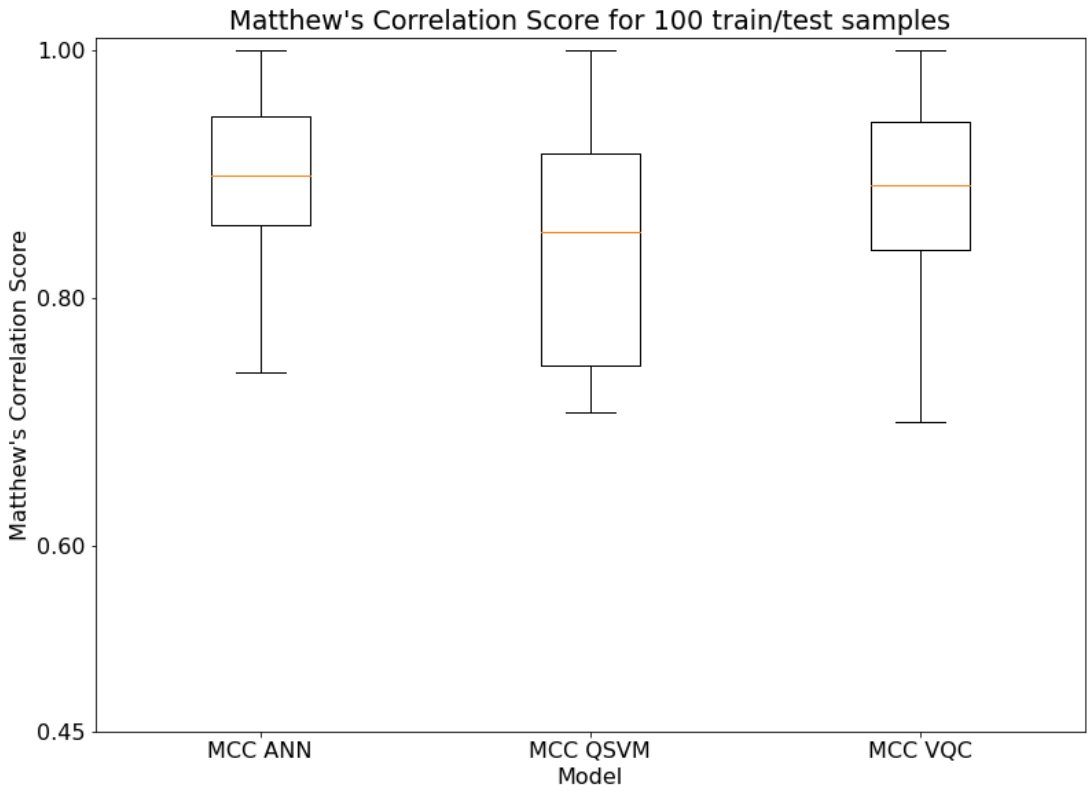

Figure 46: Matthew's correlation score distribution between ANN (left), QSVM (center), and VQC (right). The orange line represents the median of the distribution. ANN and VQC show very similar performance. QSVM shows much greater metric variation.

Lithology Classification from Petrophysical Well Log Data

An additional analysis of binary classification performance between classical and quantum-enhanced machine learning models is performed on the well log database discussed in Chapter II. The same preprocessing steps outlined in the previous chapter are applied to the data to eliminate outlier values and reduce feature dimensionality. Training/testing accuracy, and F1 score are selected as scoring metrics since the two classes shale and sandstone are fairly balanced unlike the real-time drilling dataset. The goal is to understand if VQC shows the same advantages in performance over QSVM in a larger dataset with more balanced classes.

Isolation forest is applied to detect and eliminate outliers. Standard scaling and principal component analysis are implemented to reduce the feature space to two dimensions. The decision boundary is shown for each model in Figure 47. QML models utilize a non-linear decision boundary to classify the data unlike the traditional neural network.

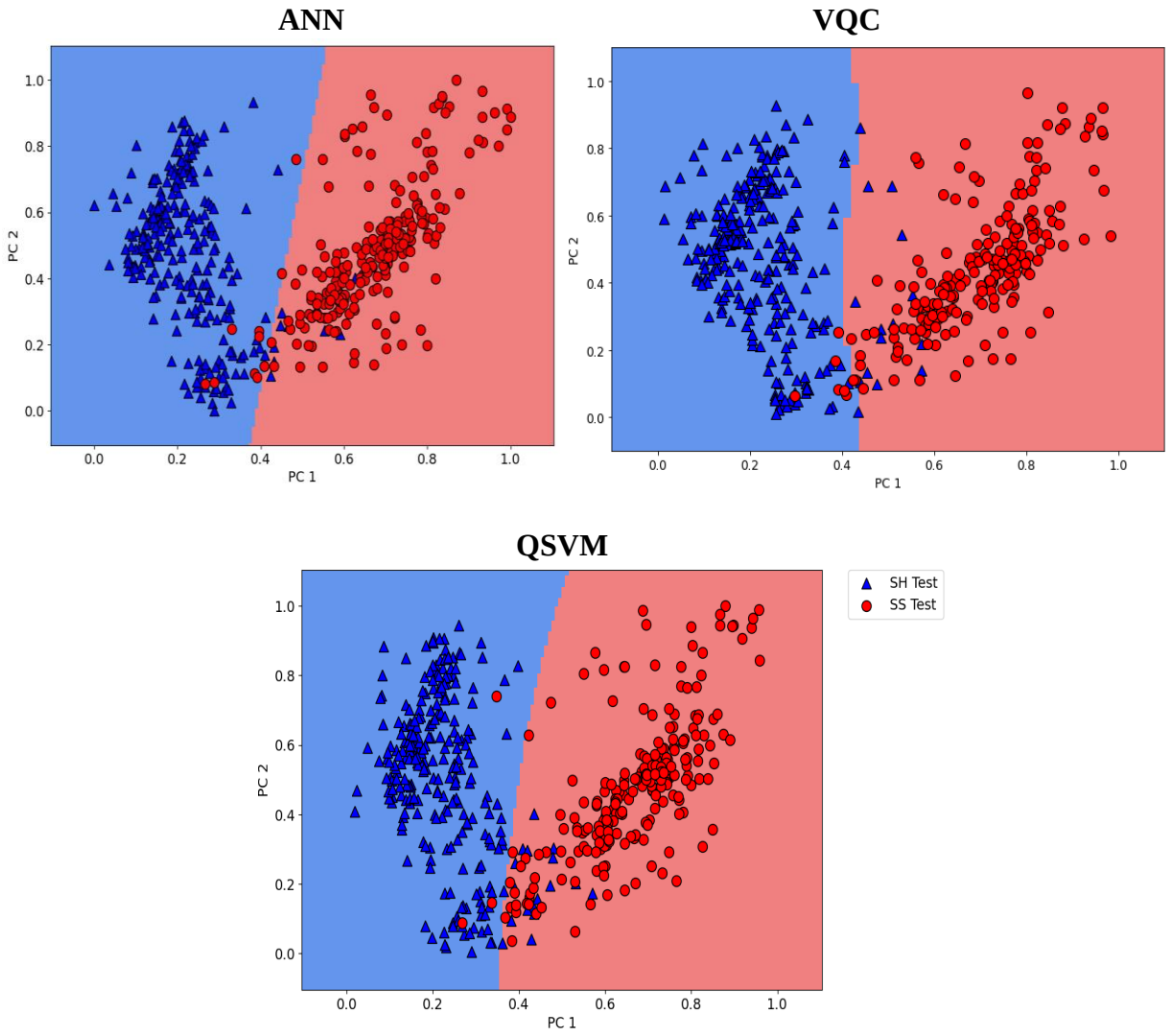

Figure 47: Decision boundaries of ANN (top left), VQC (top right), and QSVM (bottom center) on lithology classification of petrophysical well log data. The classical ANN model divides the data with a straight line whereas the QML models utilize a non-linear curve decision function.

The model training and testing is performed 100 times to record summary statistics of the resulting performance metrics. A boxplot of training/testing accuracy and F1 score is shown in Figure 48. In this dataset, QML models underperform significantly when compared to the classical neural network with VQC having the poorest performance. Moreover, quantum models display much greater variability in the results.

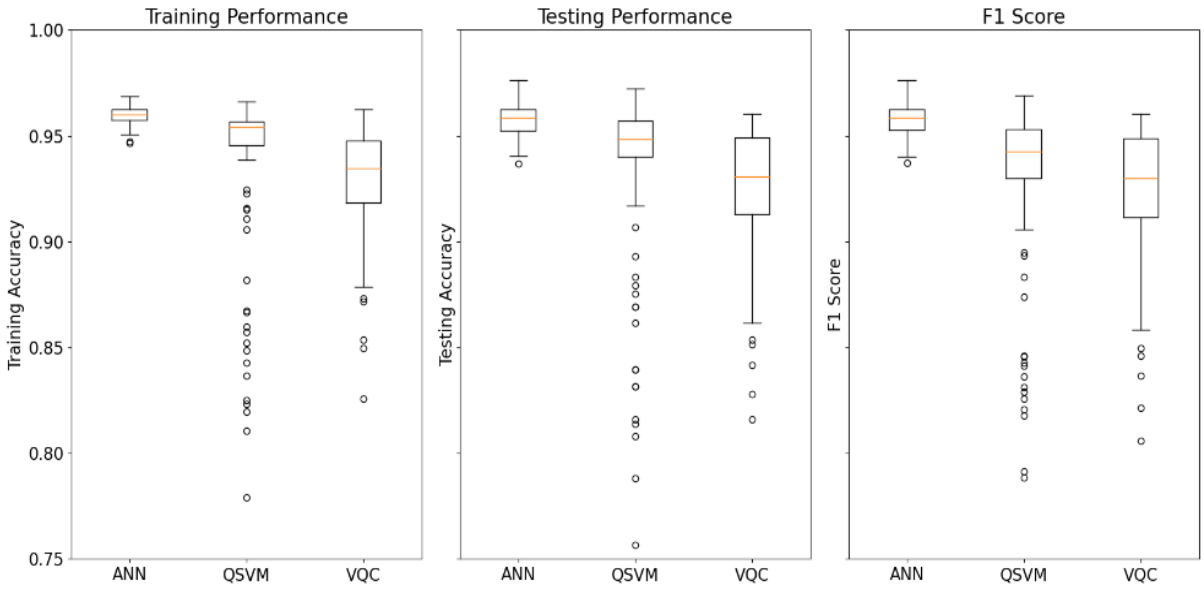

Figure 48: Performance metrics distribution in a boxplot format for training accuracy (left), test accuracy (center), F1 score (right) for lithology classification from well log data using two qubits. The quantum models both underperform compared to a classical neural network.

In addition, the secondary goal is to understand if the number of qubits in each QML model is directly related to its performance. As mentioned previously, the number of qubits in a quantum variational circuit is equal to the dimension of the feature space. In this last section, the models are applied using three qubits with the corresponding three principal components as features. Figure 49 represents the resulting distribution of score metrics for the three models.

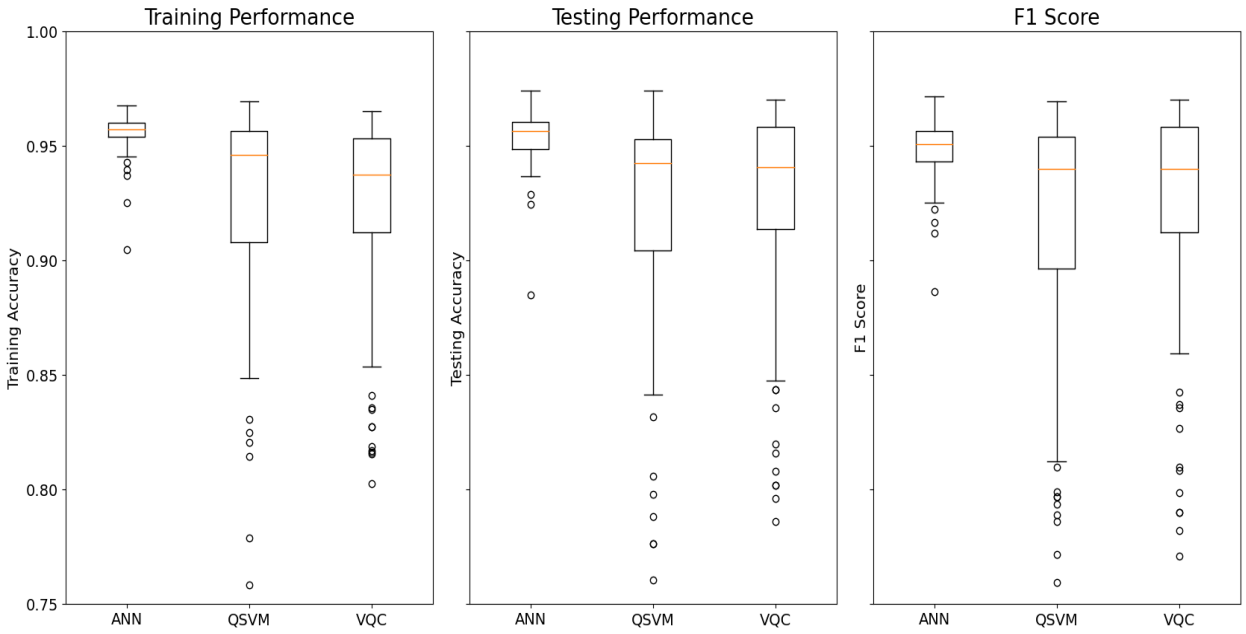

Figure 49: Performance metrics distribution in a boxplot format for training accuracy (left), test accuracy (center), F1 score (right) for lithology classification from well log data using three qubits. The quantum models both underperform compared to a classical neural network.

While the median of the three metrics remains relatively the same between the two and three qubits trials for ANN and QSVM, the performance of VQC increases. The improvement in VQC performance can be directly related to the increase in number of weights present in the variational circuit. Similar to the previous trial, QML models exhibit grater variability in the results. Overall, even with an increase in qubit count, quantum models slightly underperformed when compared with classical machine learning methods.

Computational Time

As discussed in Chapter II, computational time of quantum models is affected by the queuing time necessary for the quantum simulator to receive the data. As mentioned previously, QSVM is ran on the IBM qiskit library while VQC on the tensorflow-quantum library. The two

libraries utilize different quantum simulators with varying queuing times. However, as shown in Figure 50, computational time of classical and quantum models gives a benchmark comparison.

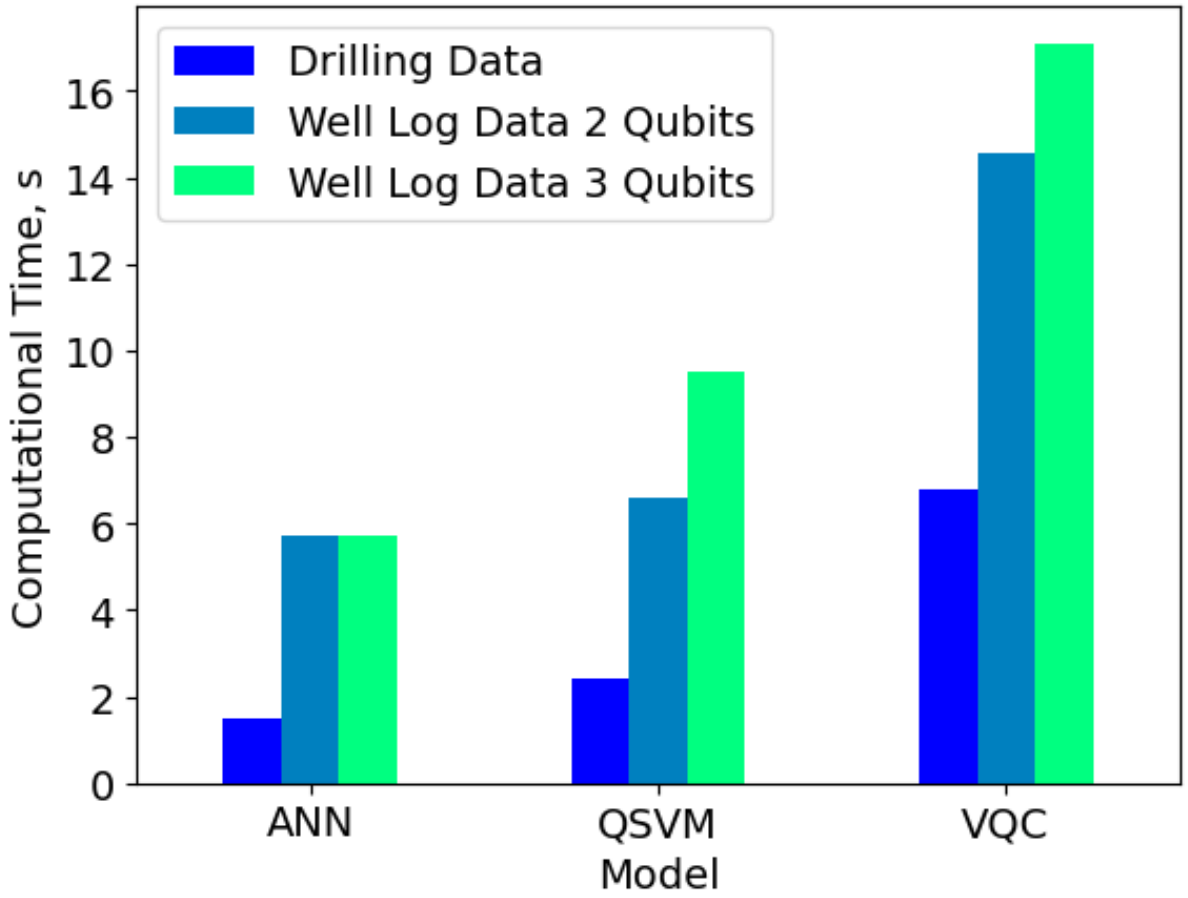

Figure 50: Comparison of computational time between ANN (left), QSVM (center), and VQC (right) for one training/testing iteration. Quantum models tend to be slower depending on the number of qubits used. Overall, classical methods perform much faster.

The time recorded includes the duration of training and testing the model once. In agreement with Chapter II analysis, the computational time increases with QML models due to the queuing time of quantum simulators. The increase in data size between the real-time drilling dataset (270 samples) versus the well log data (1600 samples) prolongs the time needed to train and test the models. Lastly, the increase in number of qubits results in a more complex feature

map and variational circuit which causes the computational time to increase. When comparing the two QML models, QSVM is much faster than VQC. This behavior can be attributed to QSVM performing the actual classification on a classical SVC algorithm as compared to VQC which uses a variational circuit in quantum space to classify the data.

Answer to Fundamental Questions

- The variational quantum classifier is tested on a highly unbalanced dataset for binary classification. From the results obtained, VQC shows almost identical median performance with ANN for both F1 weighted score and Matthew's correlation score. Although the performance did not improve, VQC is able to replicate the results of a 4-layer artificial neural network with 13 trainable weights as compared to 381 weights of the classical model.
- For the underbalance drilling dataset, VQC achieve similar results to ANN with less parameters. For the larger balanced well log dataset, VQC show significant lower performance.
- When increasing the feature space and number of qubits from 2 to 3, the median performance of VQC improved for both accuracy and F1 score on the test set. QSVM did not show any improvements when varying the number of qubits. For both quantum-enhanced models, the variability of the resulting metrics increases with higher qubits.

Conclusions

This chapter further explores the application of quantum-enhanced machine learning methods for real-word applications on petroleum engineering problems. The goal achieved is to

compare the performance and computational time of QSVM and VQC to a benchmark classical neural network. While QML methods achieved similar scoring metrics to ANN, a quantum advantage was not observed in binary lithology classification. The noisiness of current quantum devices results in a greater variability of expectation values as the input size increases. When analyzing the performance distribution on 100 different splits of train and test data, quantum models show a much greater variance in classification metrics. The use of a larger number of qubits to express the classical data into a quantum state shows a slight improvement in the parameterized model VQC which is, however, underperforming when compared to ANN. Despite the results obtained, the implementation of quantum theory on machine learning models can be proven advantageous on real-world applications with future development of quantum devices.

CHAPTER IV

RECOMMENDATIONS FOR FUTURE WORK

Image-based machine learning model for viscosity prediction

The image-based machine learning workflow lacks the ability to accurately perform regression on target viscosity. As a result, possible future work should include the use of additional image-based filters to better describe the texture of side-wall rock samples under white and UV light. Performing blob detection using the difference of Gaussian can enhance the characterization of rock sample images. In addition, the current work is limited to the available 660 image-viscosity pairs at constant temperature. If more samples become available, future work should be performed on a larger sample size.

Quantum-enhanced machine learning applications for lithology classification

QML models requires careful tuning performed on the feature map. The process of encoding a feature vector into a quantum state can be achieved using a variety of feature map and number of qubits which model entanglement differently. Future work should consist of optimizing the feature map structure based on the initial data by testing several combinations. Specific to VQC, a variational circuit should be designed with more than three qubits to achieve a number of trainable weights similar to complex artificial neural networks for fair comparison. If available, QML models should be tested on a much larger sample size to better understand their behavior. Advancement of QML models can be inherited from future development of quantum machines which are currently susceptible to high noise.

REFERENCES

Al-Amoudi, L. A., Patil, S., & Baarimah, S. O. (2019, March). Development of artificial intelligence models for prediction of crude oil viscosity. In *SPE Middle East Oil and Gas Show and Conference*. OnePetro.

Amir, Z., Jan, B. M., Wahab, A. K., Khalil, M., Ali, B. S., Tong, C. W., ... & Shafi'i, M. R. (2016, August). Density and viscosity prediction of super lightweight completion fluid SLWCF at reservoir conditions. In *IADC/SPE Asia Pacific Drilling Technology Conference*. OnePetro.

Alali, A., Marfurt, K., & Pranter, M. (2016, October). Calibrating seismic fracture prediction using borehole image logs, application to the Mississippian limestone. In *2016 SEG International Exposition and Annual Meeting*. OnePetro.

Brownlee, J. (2020). Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery.

Blance, A., & Spannowsky, M. (2021). Quantum-enhanced machine learning for particle physics using a variational quantum classifier. *Journal of High Energy Physics*, *2021*(2), 1-20.

Chelli, A., & Pätzold, M. (2019). A machine learning approach for fall detection and daily living activity recognition. *IEEE Access*, *7*, 38670-38687.

de Figueiredo, L. P., Bordignon, F., Exterkoetter, R., Rodrigues, B. B., & Duarte, M. (2019). Deep 3D convolutional neural network applied to CT segmented image for rock properties prediction. In *SEG Technical Program Expanded Abstracts 2019* (pp. 2614-2618). Society of Exploration Geophysicists.

de la Porte, J. J., Riesco, N., & Vesovic, V. (2009, October). Prediction of Heavy Oil Viscosity in Current Reservoir Simulators. In *SPE/EAGE Reservoir Characterization and Simulation Conference*. OnePetro.

Dembicki, H. (2016). Practical petroleum geochemistry for exploration and production. Elsevier.

Deumah, S. S., Yahya, W. A., Al-khudafi, A. M., Ba-Jaalah, K. S., & Al-Absi, W. T. (2021, October). Prediction of Gas Viscosity of Yemeni Gas Fields Using Machine Learning Techniques. In *SPE Symposium: Artificial Intelligence-Towards a Resilient and Efficient Energy Industry*. OnePetro.

Fernandez Diaz, P. R., Gonzalez Perez, H. J., Santos Molina, C. A., Cox, A., & Kellet, K. (2020, June). Novel and cost-time effective rock typing and mechanical properties prediction technique for unconventional samples based on 2D (radiographic) X-ray image analysis. In *54th US Rock Mechanics/Geomechanics Symposium*. OnePetro.

Foster, J., Misra, S., FALOLA, Y., & Bhatia, M. (2021). Preemptive Detection of High Water-Cut Wells in Delaware Basin using a Joint Unsupervised and Supervised Learning Approach.

Fuster, G., & Miquel, E. (2019). Variational Quantum Classifier. https://doi.org/http://hdl.handle.net/2445/140318

Han, L., Tian, Y., & Qi, Q. (2020). Research on edge detection algorithm based on improved sobel operator. In MATEC Web of Conferences (Vol. 309, p. 03031). EDP Sciences.

Havenstein, C., Thomas, D., & Chandrasekaran, S. (2018). Comparisons of performance between quantum and classical machine learning. *SMU Data Science Review*, *1*(4), 11.

Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, *567*(7747), 209-212.

He, J., Li, H., & Misra, S. (2019). Data-driven in-situ sonic-log synthesis in shale reservoirs for geomechanical characterization. *SPE Reservoir Evaluation & Engineering*, *22*(04), 1225-1239.

He, J., & Misra, S. (2019). Generation of synthetic dielectric dispersion logs in organic-rich shale formations using neural-network models. *Geophysics*, *84*(3), D117-D129.

Heredge, J., Hill, C., Hollenberg, L., & Sevior, M. (2021). Quantum support vector machines for continuum suppression in B meson decays. *Computing and Software for Big Science*, *5*(1), 1-9.

Hossain, Z. (2011, May). Relative permeability prediction from Image Analysis of thin sections. In *SPE EUROPEC/EAGE Annual Conference and Exhibition*. OnePetro.

Jin, Y., & Misra, S. (2022). Controlling mixed-mode fatigue crack growth using deep reinforcement learning. *Applied Soft Computing*, *127*, 109382.

Jobe, T. D., Vital-Brazil, E., & Khait, M. (2018). Geological feature prediction using image-based machine learning. *Petrophysics-The SPWLA Journal of Formation Evaluation and Reservoir Description*, *59*(06), 750-760.

Kariya, A., & Behera, B. K. (2021). Investigation of Quantum Support Vector Machine for Classification in NISQ era. *arXiv preprint arXiv:2112.06912*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, H., & Misra, S. (2018). Long short-term memory and variational autoencoder with convolutional neural networks for generating NMR T2 distributions. *IEEE Geoscience and Remote Sensing Letters*, *16*(2), 192-195.

Liu, Y., Arunachalam, S., & Temme, K. (2021). A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, *17*(9), 1013-1017.

Liu, N., Huang, T., Gao, J., Xu, Z., Wang, D., & Li, F. (2021). Quantum-enhanced deep learning-based lithology interpretation from well logs. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1-13.

Liu, R., & Misra, S. (2022). A generalized machine learning workflow to visualize mechanical discontinuity. *Journal of Petroleum Science and Engineering*, *210*, 109963.

Maheshwari, D., Sierra-Sosa, D., & Garcia-Zapirain, B. (2021). Variational Quantum Classifier for Binary Classification: Real vs Synthetic Dataset. *IEEE Access*, *10*, 3705-3715.

O'Shea, D. (2021, December 29). *Year-end commentary series: Quantinuum on the state of Quantum*. Inside Quantum Technology. Retrieved October 2, 2022, from https://www.insidequantumtechnology.com/news-archive/year-end-commentary-series-quantinuum-on-the-state-of-quantum-2/

Park, J. E., Quanz, B., Wood, S., Higgins, H., & Harishankar, R. (2020). Practical application improvement to Quantum SVM: theory to practice. *arXiv preprint arXiv:2012.07725*.

Rudzki, M. (2011). Automatic liver vasculature segmentation method using a modified multiscale vesselness filter. *In Proceedings of XIII International PhD Workshop*. Conference Archives PTETiS (Vol. 29, pp. 300-5).

Ryder, A. G., Iwanski, P., & Montanari, L. (2004). Fluorescence emission from petroleum; a valuable information source for petroleum analysis. *J. Petro. Chem*, *70*, 1-9.

Sairamya, N. J., Susmitha, L., George, S. T., & Subathra, M. S. P. (2019). Hybrid approach for classification of electroencephalographic signals using time–frequency images with wavelets and texture features. In *Intelligent Data Analysis for Biomedical Applications* (pp. 253-273). Academic Press.

Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., ... & Kikinis, R. (1998). Threedimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical image analysis*, 2(2), 143-168.

Schuld, M., & Petruccione, F. (2021). Quantum Models as Kernel Methods. In *Machine Learning with Quantum Computers* (pp. 217-245). Springer, Cham.

Sierra-Sosa, D., Arcila-Moreno, J., Garcia-Zapirain, B., Castillo-Olea, C., & Elmaghraby, A. (2020). Dementia prediction applying variational quantum classifier. *arXiv preprint arXiv:2007.08653*.

Vesovic, V. (2007, April). Flow Assurance: Reliable and Accurate Prediction of the Viscosity of Natural Gas. In *SPE Latin America and Caribbean Petroleum Engineering Conference* (pp. SPE-107154). SPE.

Wu, Y., Misra, S., Sondergeld, C., Curtis, M., & Jernigen, J. (2019). Machine learning for locating organic matter and pores in scanning electron microscopy images of organic-rich shales. *Fuel*, *253*, 662-676.

Wu, Y., Bao, W. S., Cao, S., Chen, F., Chen, M. C., Chen, X., ... & Pan, J. W. (2021). Strong quantum computational advantage using a superconducting quantum processor. *Physical review letters*, *127*(18), 180501.

Zhao, H., Memon, A., Gao, J., Taylor, S. D., Sieben, D., Ratulowski, J., ... & Creek, J. (2016). Heavy oil viscosity measurements: Best practices and guidelines. *Energy & Fuels*, *30*(7), 5277-5290.