A COMPARATIVE ANALYSIS OF EXTREME GRADIENT BOOSTING AND SUPPORT VECTOR REGRESSION FOR MODELING BENCHMARK CRUDE OIL PRICES

ALABI, Nurudeen O. & OJO, Gabriel O.
Department of Mathematics & Statistics, Federal Polytechnic, Ilaro, Ogun State.
Corresponding author email: nurudeen.alabi@federalpolyilaro.edu.ng

ABSTRACT
Organization of Petroleum Exporting Countries (OPEC) and non-OPEC supply, oil balance, oil demand by Organization for Economic Cooperation and Development (OECD) and non-OECD members, money market managers' long positions, US consumer price index and spot prices of crude oils like New York Mercantile Exchange West Texas Intermediate (NYMEX WTI), Intercontinental Exchange (ICE) Brent, OPEC Reference Basket (ORB), and other crude oils are basic elements driving the patterns seen in the market pricing of crude oils. Data between 2008 and 2022 were obtained for this study from OPEC Monthly Oil Market Reports. This research evaluates the performance of two machine learning models, Support Vector Regression (SVR) and Extreme Gradient Boosting (XGBoost), in predicting crude oil prices for three major benchmarks: OPEC Reference Basket (ORB), NYMEX WTI, and ICE Brent. Using key performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2), the study highlights the strengths and weaknesses of each model in both stable and volatile market conditions. SVR shows strong predictive accuracy, particularly for ICE Brent, but struggles with price volatility in the ORB and NYMEX WTI datasets. XGBoost is more robust in handling volatility and non-linear relationships. The findings have important economic implications for market participants, suggesting that while SVR is suited for stable pricing environments, XGBoost is better equipped to handle the unpredictability of more volatile markets.

Keywords: Ensemble Algorithms, XGBoosting, ICE Brent, NYMEX WTI, ORB, OPEC, OECD Support Vector Regression

1.0 INTRODUCTION

Ensemble algorithms are sophisticated techniques that combine the predictions of multiple base models to improve overall forecasting accuracy and robustness. They capture complex nonlinear relationships, handle high-dimensional data, and effectively deal with noisy and dynamic patterns inherent in data. These algorithms leverage the diversity of multiple models to mitigate overfitting and improve generalisation performance. By evaluating their accuracy, robustness, and computational efficiency, we seek to gain insights into the most effective algorithm for this specific application, which can inform decision-making processes, risk management strategies, and investment decisions in the global crude oil market.

The global crude oil market is highly robust, with relatively few producers. In 2021, the total world crude oil demand averaged 96.44 million barrels per day (mb/d), with OECD countries, non-OECD and OPEC-13 providing 29.56, 31.9, and 28.9 mb/d respectively. Crude oil prices are determined by the market forces of demand and supply, with higher demand in rapidly developing countries driving prices in the upward direction. Additionally, speculative activities of money managers through their total futures and net-long positions in crude oil futures contribute significantly to price volatility. The exchange value of the US dollar also plays an important role in the fluctuations of crude oil prices globally. Seven fundamental factors are responsible for crude oil price fluctuations. Financial markets, OPEC and non-OPEC crude oil supply, crude oil balance, oil demand by OECD and non-OECD members, and spot prices of crude oils such as NYMEX WTI, ICE Brent, EIA's Imported Refiner Acquisition Cost (IRAC), OPEC Reference Basket (ORB) and other crude oils determine the differences in other crude oils globally. ExxonMobil (2021) claims that worldwide crude oils can be divided into light, medium, and heavy categories based on density (API gravity). Sweet crude oils have a sulphur level between zero percent and 0.59 percent of the weight, while sour crude oils range from 0.62 percent to 3.85 percent by weight. The new OPEC Reference Basket contains thirteen crude oils, including Saharan Blend (Algeria), Iran Heavy (Iran), Zafiro, Basrah Medium (Iraq), Girrasol, Kuwait Export (Kuwait), Es Sider (Libya), Bonny Light (Nigeria), Djeno, Merey, Arab Light (Saudi Arabia), Murban (UAE), and Rabi Light. The American Petroleum Institute (API) gravity and Sulphur content of the WTI are respectively 39.6 degrees and 0.24 percent, while Brent has an API gravity of 38.3 degrees and contains 0.37 percent sulfur. The NYMEX futures price for crude oil is the value of a futures contract which depends purely on market forces on trading one thousand barrels of the light-sweet crude oil at a particular time. The IRAC represents the volume-weighted average cost of all heavy and light crude oils imported into the United States during a specific period. To ensure stable, open, and predictable oil markets, producers and consumers must act responsibly regarding supply and demand security. The behavior of crude oil. The behavior of crude oil market prices on a worldwide scale is influenced by several factors, as was previously mentioned. The directional relationships and interactions between these variables and spot prices are shown in Figure 1.

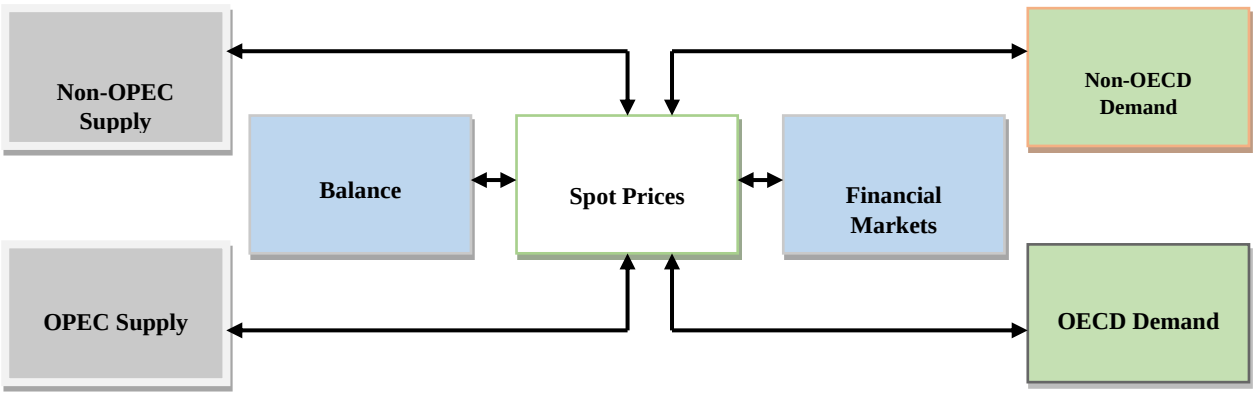


Figure 1: Determinants of Crude Oil Prices and their Interactions
Source: US Energy Information Administration (EIA, 2022)

These links serve as the foundation for this study, which aims to establish two objectives

- Compare the performances of three widely used ensemble machine learning algorithms such as extreme gradient boosting, and support vector regression on the crude oil datasets.
- To determine the most relevant determinants for modeling the benchmark crude oil prices in the global crude oil market.

Researchers from all around the world have conducted several studies on the modeling and forecasting of crude oil prices. For instance, Lu *et al*. (2021) created a framework for selecting and forecasting the key variables that affect the price of crude oil. The study used a Long-Short Term Memory Network, Spike-Slab LASSO, Bayesian model average (BMA), and an elastic net regularized generalized linear model (GLMNCI) (LSTM). Based on a random walk, six forecasting methods—Walvet neural networks (WNN), Elman neural networks (ENW), neutral ELM networks, autoregressive integrated moving average models (ARIMA), and generalized regression neural network model—were compared (GRNN). According to the analysis, the LSTM has the greatest precision. In a different work, Bai, Yuying, and Shonyang (2021) generalized integral-valued forecasts to include uncertainty and variability in the price of crude oil and presented a two-stage forecasting method based on interval-valued time series. This process outperformed some benchmark models when compared, it was discovered. Krzysztof and Liu conducted a study that is comparable in that it deals with the significant problem of uncertainty that is present in time series analysis in 2021. The study compared several time-varying VARs that included geopolitical risk as an endogenous component. They concluded that real prices for crude oil vary significantly over time.

Dondukova and Lin (2021) used the Euler-Mamyama scheme as an approximation of the Heston model to model the volatility of WTI and Brent. In particular, it was discovered that the stochastic volatility model outperformed all GARCH models when they were examined via the RMSE and MAE. Waqas et al. (2018) developed an ensemble empirical model decomposition (EEMD) To anticipate the price of crude oil. It has been demonstrated that this alternative method to traditional econometric methods aids in dealing with non-stationarity and non-linearity of time series, particularly crude oil prices. Wajdi and Dawud (2018) fitted a crude oil price regression on its key variables chosen using PCA. These are the geopolitical and fundamental factors. Analysis revealed that the most crucial factors affecting crude oil prices are fundamentals and the responsibilities of OPEC members. Wassin and Ibrahim (2018) analyse the linear and non-linear regression models to examine the relationships between crude oil prices and stocks. They used the informational value of oil demand and the link between crude oil and equities prices to fit the linear models. Similarly, a non-linear model was fitted using fuzzy neural networks and genetic algorithms. In terms of the accuracy of the statistical forecasts made outside of the sample, some of the fitted linear models performed the best.

By combining prior knowledge from the current and anticipated structure of the oil markets with the Bayesian technique, Chul-Yoon and Sung-Yoon (2016) predicted long-term crude oil prices. The model's stated factors for determining crude oil prices include factors including global oil demand and supply, economic conditions, upstream costs, and geographic occurrences. The OLS and neural network were contrasted using this model. The fitted model was found to perform better on the forecasting performance test alone than these two models. Merk (2016) used the daily return of crude oil prices to examine the relationship between global financial crises and volatility. They concluded that crude oil prices are extremely volatile and substantially respond to shocks from the global financial crisis after fitting volatility models such as the APGARCH and FIAPGARCH models. To estimate the price of crude oil, Ani and Rubaidah (2014) devised a hybrid model that combines wavelet and multiple linear regression. PCA and the Mallat wavelet transform are used in this work. To choose the best model for the multiple linear regression on the WTI, they used Particle Swarm Optimisation (PSO). The WMLR outperformed the ARIMA, MLR, and GARCH models when they were put side by side.

2.0 MATERIALS AND METHODS

2.1 Materials

The data used in this study includes 176 monthly observations between 2008 and 2022 of the prices of a barrel of ORB (orb_t), WTI (wti_t), and Brent ($brent_t$) in US dollars, as well as data on OPEC supply ($opec\_supply_t$), non-OPEC supply ($nonopec\_supply_t$), money market managers' net long positions ($mo\_net\_long_t$), OECD demand ($oecddemand_t$), non-OECD demand ($nonoecddemand_t$), the oil balance ($balance_t$), and US Consumer Price Index (us_cpi_t). The source of data used is the OPEC Monthly Oil Market Reports.

2.2 Methods

2.2.1 Extreme Boosting Machine Learning Algorithm

Extreme Boosting is a popular ensemble machine learning algorithm used in various domains, including time series forecasting. It encompasses implementations like XGBoost, LightGBM, and CatBoost, each with unique features and optimisations. The algorithm follows a boosting concept and ensemble methodology. These algorithms allow base learners, typically decision trees, to be sequentially trained to minimize the overall prediction error by emphasizing difficult instances. The algorithm iteratively adds decision trees to the ensemble, updating the predictions and minimising the loss function. Parameter tuning and model optimisation are crucial for effective application. Vital considerations include the learning rate, tree depth and complexity, and regularisation parameters. The formula for updating the ensemble model involves combining the previous predictions with the contribution of the newly added tree. Each iteration minimises a specific loss function concerning the ensemble's errors. The specific implementation and formulas may vary depending on the Extreme Boosting algorithm used. Additional optimisations and enhancements are introduced in algorithms like *XGBoost*, *LightGBM*, and *CatBoost* to improve training speed, accuracy, and handling of different data types. In Extreme Boosting, the algorithm aims to minimise a specific loss function during the training process. The choice of loss function depends on the task at hand, such as mean squared error (MSE) or mean absolute error (MAE) for regression problems. During each iteration of the boosting process, a new decision tree is added to the ensemble to reduce the overall loss. The tree is constructed to minimise the chosen loss function concerning the residuals or errors made by the ensemble model on

the previous iterations. By iteratively updating the ensemble and adding trees that target the remaining errors, the algorithm gradually improves the overall model performance and reduces the prediction errors. This process continues until a predefined stopping criterion is met, such as reaching a maximum number of iterations or achieving satisfactory performance. The iterative nature of Extreme Boosting, combined with the minimisation of the loss function, allows the algorithm to effectively learn complex patterns and make accurate predictions.

Mathematical Aspect of Extreme Gradient Boosting Ensemble

Given a dataset with *n* observations and *m* features

$$D = \left\{ \left(x_i, y_i \right) \right\} \left(\left| D \right| = n, x_i \in R^m, y_i \in R \right) \quad (1)$$

Where we predicted the value of the ensemble tree model using *K* additive functions expressed as

$$\hat{y}_i = \sum_{k=1}^{K} f_k \left(x_i \right), f_k \in F \quad (2)$$

Where *F* is the regression tree space computed as

$$F = \left\{ f\left(x \right) = \omega_x \left(x \right) \right\} \left(q : R^m \to T, \omega_q \in R^T \right) \quad (3)$$

q represents the structure of each tree, *T* is the number of nodes (leaves) in the tree and f_k is a function that corresponds to an independent tree structure *q* and leaf weights ω. Given a training dataset with input features *x* and target variable *y*, and an ensemble model represented by *f* (*x*), the goal is to minimise the loss function

$$L^{(t)} = \sum_{i=1}^{n} l\left(\left(y_i, \hat{y}_j^{(t-1)} \right) + f_t \left(x_i \right) \right) + \Omega\left(f_k \right) \quad (4)$$

where *l* is a differentiable convex objective function to calculate the error between predicted and measured values; y_i and y_i are regulated and predicted values, respectively; *t* shows the repetitions to minimise the errors; and **Ω** is the complexity penalised with the regression tree functions:

$$\Omega\left(f_k \right) = \gamma T + \frac{1}{2} \lambda \left\| \omega \right\|^2, \quad (5)$$

ω is the vector of the score for the blades, and *c* is the minimal loss required for the further isolation of a blade node. λ is the regularisation function. In addition, *c* and λ are parameters that can control the complexity of the tree, and the regularisation term helps to avoid overfitting by smoothening the final learned weights. Taylor expansion is applied to the objective function to simplify it further as

$$F = \sum_{i=1}^{m} \left[f_t \left(x_i \right) g_i + \frac{1}{2} \left(f_t \left(x_i \right) \right)^2 h_i \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2 \quad (6)$$

where g_i and h_i are the first and second derivatives obtained on the loss function, respectively. Specifically,

$$g_i = \partial_{y_i^{(t-1)}} l\left(y_i, y_i^{(t-1)} \right) \quad (7)$$

$$g_i = \sum_{i=1}^{n} l\left(y_i, y_i^{(t-1)} \right) + \sum_{k=1}^{t-1} \Omega\left(f_k \right) \quad (8)$$

$$h_i = \partial^2_{y_i^{(t-1)}} l\left(y_i, y_i^{(t-1)} \right) \quad (9)$$

The optimal objective function in equation (6) as a function of the 1st and 2nd derivatives is

$$L^{(t)} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T + C \quad (10)$$

$$G_j = \sum_{i \in I_j} g_i$$

and (11)

$$H_j = \sum_{i \in I_j} h_i$$

The *XGBoosting* process involves a series of steps where a new decision tree is added to the group to reduce errors. This new decision tree, denoted as *g*, is trained to make predictions that are closer to the actual target variable, *y*, compared to the current group of models. After training, this new tree becomes part of the ensemble. This cycle repeats, with each new tree trained to improve predictions based on the current ensemble. The ultimate goal of Extreme Gradient Boosting is to create an ensemble model, *f*(*x*), that minimizes the total errors across all these steps. By minimising errors at each step and updating the ensemble accordingly, *XGBoost* gradually enhances the model's predictive accuracy, leading to a more precise final model. In this study, the researchers compared *XGBoost* with other ensemble methods like bagging, random forest, and support vector regression, as proposed in the existing literature.

2.2.2 Support Vector Regression (SVR)

Support Vector Regression (SVR) is a non-linear regression method that can capture non-linear relationships between the input features and the target variable. It is based on the concept of support vector machines (SVMs), which were originally developed for classification tasks. SVR uses a kernel function to map the input features into a higher-dimensional space where the relationship between the features and the target variable may become linear. In this transformed space, SVR attempts to find a linear regression model that fits the data while maximising the margin (the distance between the regression line and the closest data points, which are the support vectors). The choice of kernel function, such as the radial basis function (RBF) kernel, allows SVR to model complex, non-linear relationships. It has the flexibility to capture both linear and non-linear relationships in the data, depending on the choice of kernel and model parameters. This makes SVR a powerful tool for regression tasks with complex data patterns. Support Vector Regression (SVR), support vectors, and weights play important roles in understanding the model. The support vectors are data points from the training dataset that have the most influence on the SVR model. These are the data points that are closest to the SVR's decision boundary (the hyperplane) and are used to define the margin. They are the points for which the model's prediction is either equal to the target value or falls within a certain distance (the margin) from the target value. They play a crucial role in determining the SVR model's performance and are the ones responsible for shaping the model's regression line. Hence, Support vectors are the critical data points that have the most influence on the model's prediction. In SVR, weights represent the importance or contribution of each support vector to the model's prediction. Each support vector is associated with a weight that signifies its significance in shaping the regression model. Weights are essentially the coefficients of the support vectors in the SVR equation. Support vectors with larger weights have a more substantial impact on the model's decision boundary and, consequently, on its predictions. Vector weights provide

information about the importance of different data points in the SVR model, helping to understand which data points are the most influential in making predictions. Support vectors are the data points closest to the decision boundary, and they have a direct impact on the SVR model's predictions. The weights associated with these support vectors indicate their relative importance in shaping the model. By examining support vectors and their weights, we gain insights into which data points are driving the SVR model's predictions and the significance of each of these points in the regression analysis. By analysing the weights, allows us to pinpoint the most influential data points and focus on their characteristics, which are valuable for understanding the key factors driving the SVR model's performance and enhancing its predictive capabilities. High-level SVR algorithm involves data pre-processing by standardizing or normalising the input features to ensure they are on a similar scale, choosing a kernel function (linear or radial basis function) to transform the input features and formulation of the SVR optimisation problem. The objective is to find a hyperplane that has the maximum margin while minimising the error (the difference between the predicted and actual values). The introduction of a soft margin allows some instances to be within the margin or even on the wrong side of the hyperplane. This is done to handle cases where a strict margin might not be achievable due to noise or outliers. We apply cross-validation to control the complexity of the model and prevent overfitting. We define a loss function that penalises errors in prediction. Common loss functions include epsilon-insensitive loss or mean squared error. Solve the dual problem using optimisation techniques. The solution provides the support vectors and their corresponding weights. Use the obtained support vectors and weights to make predictions on new, unseen data. The basic idea is to find a hyperplane that not only fits the data well but also has a maximum margin. SVR is especially useful when dealing with non-linear relationships between input features and the target variable, thanks to the kernel trick that allows for implicit mapping to higher-dimensional feature spaces.

To determine the support vectors and their associated weights, we formulated a dual SVR problem which was solved using the Lagrangian dual optimisation. Generally, the dual problem is a maximisation problem. The Lagrangian function was formulated by combining the objective function of the primal problem with the constraints each multiplied by the Lagrange multiplier α_i. The Lagrangian for linear and radial basis function kernels are given as follows.

Radial Basis Function Kernel (RBF)

The primal problem for SVR with an RBF kernel $K(X_i, X_j)$ is given as

$$Minimize\ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\ \sum_1^n(\xi_i + \xi_i^*) \qquad (12)$$

Subject to constraints:

$$\begin{aligned} y_i - f(x_i) &\leq \varepsilon + \xi_i \\ f(x_i) - y_i &\leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0,\ \forall\, i = 1, 2, 3, \ldots\ldots\ldots, n \end{aligned} \qquad (13)$$

ε is the acceptable error, ξ_i, ξ_i^* are the slacks and $f(x_i)$ are the predicted output using the RBF kernel. C in the minimisation problem is the regularisation parameter. The Lagrangian for the RBF kernel is

$$L(\mathbf{w}, b, \alpha, \alpha^*, \xi_i, \xi_i^*) = \frac{1}{2}\|w\|^2 + C\sum_1^n(\xi_i + \xi_i^*) + \sum_1^n \alpha_i^*\left(\xi_i^* - \varepsilon - y_i - f(x_i)\right) + \sum_1^n \alpha_i \left(\xi_i - \varepsilon + y_i - f(x_i)\right) \qquad (14)$$

$f(x_i)$ is the output of the RBF expressed as:

$$f(x_i) = \sum_{j=1}^n \alpha_j(\mathbf{x_j})K(\mathbf{x_j}, \mathbf{x_i}) + b \qquad (15)$$

where $K(\mathbf{x_j}, \mathbf{x_i})$ is the RBF kernel. The dual problem involves maximising the Lagrangian for the Lagrange multipliers α, α^* subject to the constraints $\alpha, \alpha^* \geq 0$ for all i. The RBF kernel function and its parameters play a crucial role in the dual optimisation problem.

Linear Kernel

Just like the RBF kernel, the primal for the dual problem in equation (1) for the linear kernel is subject to the following constraints

$$\begin{aligned} y_i - (\mathbf{w}\cdot\mathbf{x_i} + b) &\leq \varepsilon + \xi_i \\ (\mathbf{w}\cdot\mathbf{x_i} + b) - y_i &\leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0,\ \forall\, i = 1, 2, 3, \ldots\ldots\ldots, n \end{aligned} \qquad (16)$$

The Lagrangian for the linear kernel is

$$L(\mathbf{w}, b, \alpha, \alpha^*, \xi_i, \xi_i^*) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_1^n(\xi_i + \xi_i^*) + \sum_1^n \alpha_i^*(\xi_i^* - \varepsilon - y_i + \mathbf{w}\cdot\mathbf{x_i} + b) + \sum_1^n \alpha_i \left(\xi_i - \varepsilon + y_i - \mathbf{w}\cdot\mathbf{x_i} - b\right) \qquad (17)$$

The estimation process is one of finding a hyperplane with a maximum margin while minimising the error which involves solving the constrained optimisation primal problem. The slack variables allow for the introduction of soft margins which allows some instances to be within the margin or on the wrong side of the hyperplane. We employed the epsilon insensitive loss function and the mean square error to penalise errors in prediction. The solution of the Lagrangian dual optimisation provides the support vectors and their associated weights which were used to define the function $f(x)$ for predicting the target variable.

3.0 RESULTS AND DISCUSSION

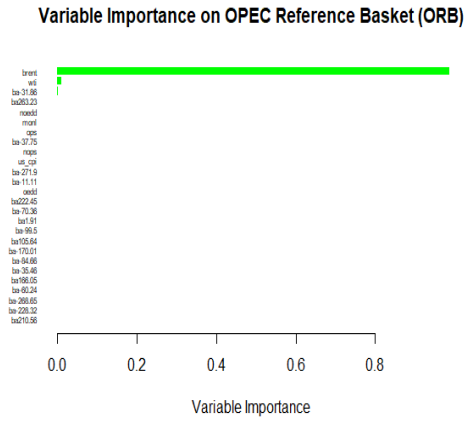
We present the analysis and results derived from the application of the two discussed machine learning models to forecast benchmark crude oil prices, including ORB, Brent, and WTI. The performance of the Support Vector Regression (SVR) and XGBoost was evaluated using key metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 for both training and testing datasets. By comparing these metrics, we aim to establish the models' effectiveness in predicting crude oil prices across different datasets, providing insights into their accuracy and generalisation capabilities. The following are the detailed findings and visual comparisons, leading to a thorough assessment of each model's predictive performance.

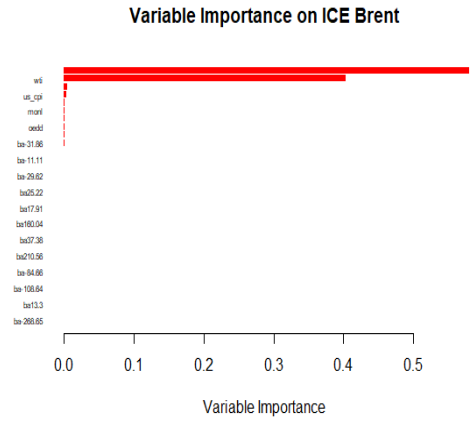
Table 1: Summary of Training on the XGBoost models with hyperparameters

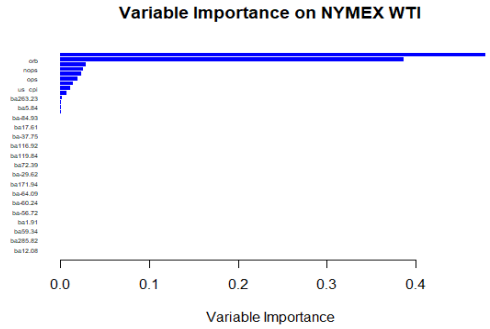
model	nround	max_depth	eta	gamma	colsample_bytree	min_child_weight	subsample
xgboost_wti	100	3	0.1	0.1	0.6	1	0.8
xgboost_brent	100	3	0.1	0.2	0.6	1	0.8
xgboost_orb	100	3	0.1	0.1	1	1	1

Three different models on ORB, Brent and WTI were built using stringent parameters. The training process for XGBoost models on the three oil benchmarks is outlined in Table 1. For each model, the tuning parameters *nrounds* and *max_depth* were held constant at 100 and 3, respectively. The models were optimised using Root Mean Square Error (RMSE) to identify the best-performing parameters. Common final values across the models include a learning rate (eta) of 0.1, a *min_child weight* of 1, and subsampling (subsample) values set at 0.8 for WTI and Brent and 1 for ORB. The specific differences in the models come from gamma, which regulates the minimum loss reduction for splitting nodes. It was set at 0.2 for WTI, 0.1 for Brent, and ORB. Additionally, *colsample_ bytree* (the proportion of features used for each tree) was 0.6 for both WTI and Brent but set to 1 (all features) for ORB. Each model was trained with 182 features, using 100 iterations (boosting rounds), and employed the reg: squared error objective to minimise squared error during regression. The table presents hyperparameters for three XGBoost models (*xgboost_vti*, *xgboost_brent*, and *xgboost_orb*). All models undergo 100 boosting rounds with a maximum tree depth of 3, ensuring the trees are shallow to control overfitting. The learning rate (eta) is set to 0.1 for all, balancing model performance and convergence speed. The minimum loss reduction to make a split (gamma) is set at 0.1 for *xgboost_vti* and *xgboost_orb*, allowing moderate splits, while *xgboost_brent* has a stricter requirement with a gamma of 0.2, limiting tree complexity unless significant gains are achieved. For feature sampling (*colsample_bytree*), *xgboost_vti* and xgboost_brent use 60% of the features for each tree, while xgboost_orb uses all features (100%). The minimum sum of instance weights needed to create a new leaf node (*min_child_weight*) is set to 1 across all models, ensuring trees do not split unless meaningful. Finally, *xgboost_vti* and *xgboost_brent* use 80% of the training data for each tree (subsample = 0.8), while *xgboost_orb* uses all the data (subsample = 1), potentially increasing the risk of overfitting for the latter.

Figure 2: Variable Importance on ORB, ICE Brent and NYMEX WTI







The three variable importance plots illustrate how different variables contribute to the prediction models for oil benchmarks: OPEC Reference Basket (ORB), ICE Brent, and NYMEX WTI. In the ORB model, Brent price overwhelmingly dominates the predictions with an importance close to 0.8, while the other variables contribute very little. This suggests that the model heavily relies on an ICE Brent to make accurate predictions for the ORB. For the ICE Brent model, a similar trend is observed, where one variable WTI shows the highest importance at around 0.5. However, the influence of other variables is minimal, indicating that the Brent model also focuses on NYMEX WTI as the key predictor for most of its forecasting accuracy. In contrast, the NYMEX WTI model exhibits a broader distribution of variable importance. While one variable (ORB) still stands out with an importance around 0.4, other variables play a more significant role compared to the other two models. This suggests that the WTI model captures a wider range of factors in making predictions, making it more dependent on multiple variables.

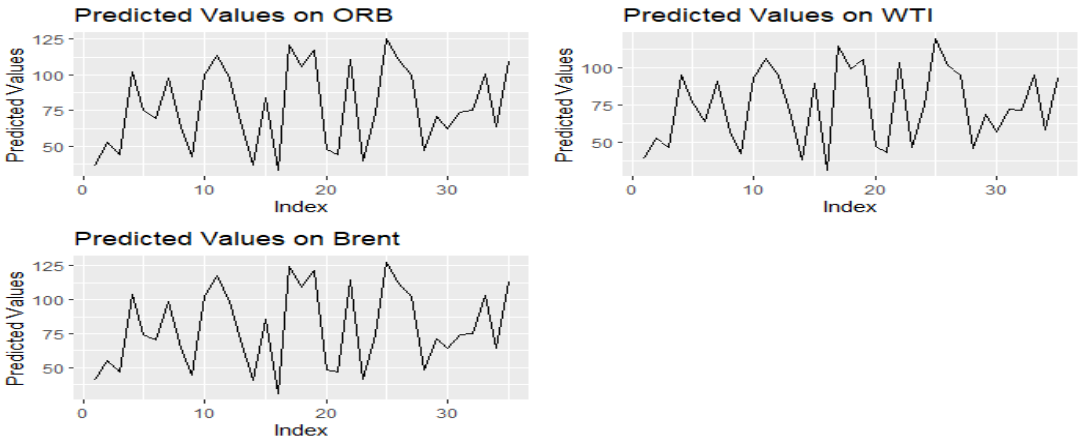


Figure 3: Plot on Predicted Values on ORB, WTI and Brent

The predicted values generated by the XGBoost models for the OPEC Reference Basket (ORB), NYMEX WTI, and ICE Brent benchmarks are presented in Figure 3. Across all three benchmarks, the predictions show a high degree of variability, with values fluctuating between 50 and 125. This suggests that the models capture the dynamic changes in the oil markets, reflecting how volatile and unpredictable the price movements can be. For the ORB and Brent benchmarks, the predictions display more extreme fluctuations, reaching up to 125, with noticeable peaks and troughs throughout the index range. This highlights that the models are sensitive to shifts in the market, perhaps responding to key influencing factors like supply and demand changes, geopolitical events, or production adjustments. The WTI model shows similar oscillations but within a slightly narrower range, peaking around 100. While the predicted values still show sharp changes, the range indicates that the WTI model may be capturing slightly less variability compared to the ORB and Brent models. Overall, all three models provide predictions that align with the complex and volatile nature of the oil price benchmarks they represent.

We carried out training on three SVR models based on a linear kernel and epsilon regression. Epsilon regression allows the model to ignore small errors within a specified margin (epsilon). The main parameters listed include cost (C), which controls the trade-off between maximising the margin and minimising error, and epsilon, which defines how much deviation from the actual values is tolerated. All models use a linear kernel, meaning the relationship between the input features and the target variable is assumed to be linear. The WTI model has a cost of 1, which strikes a balance between penalising errors and maximising the margin. The epsilon value is set at 0.1, allowing some tolerance for prediction errors. With 76 support vectors, this model has fewer data points influencing the decision boundary, suggesting a simpler model with a relatively straightforward decision margin. The higher cost in this model likely leads to better performance but with a risk of overfitting. The ORB and Brent models both have a lower cost (C = 0.1), meaning they tolerate more misclassifications in favour of a wider margin. They also have a smaller epsilon (0.01), making them more sensitive to small prediction errors. These models have more support vectors (118 for ORB and 116 for Brent), indicating a more complex decision boundary. The higher number of support vectors suggests that more data points are influencing the model, possibly due to the lower cost and the focus on minimising even small errors. We evaluated the performance of the Support Vector Regression (SVR) model in predicting crude oil prices for three major benchmarks: OPEC Reference Basket (ORB), NYMEX West Texas Intermediate (WTI), and ICE Brent. Each benchmark represents a critical component of the global oil market, and accurate price predictions are essential for risk management, market analysis, and strategic decision-making. By analysing major performance metrics such as MSE, RMSE, and R^2, we assess the predictive accuracy of the SVR model across both training and testing datasets. This analysis aims to highlight the model's strengths and potential areas for improvement in forecasting oil prices for these three critical benchmarks.

Table 2: Summary of performance metrics on SVR models using the training and test datasets

	Training			Testing		
Metric	ORB	Brent	WTI	ORB	Brent	WTI
MSE	2.71	1.126	5.593	1.502	1.501	42.72
RMSE	1.646	1.061	2.365	1.225	1.225	6.536
R^2	0.995	0.999	0.989	0.996	0.998	0.944

Table 2 presents performance metrics for the SVR models. The table is divided into training and testing sets for each dataset. While the

model performs well on the ORB and Brent datasets, it struggles to generalise to the WTI dataset. This implies that SVR is not well suited to the WTI dataset. This is also shown in Figure 4.

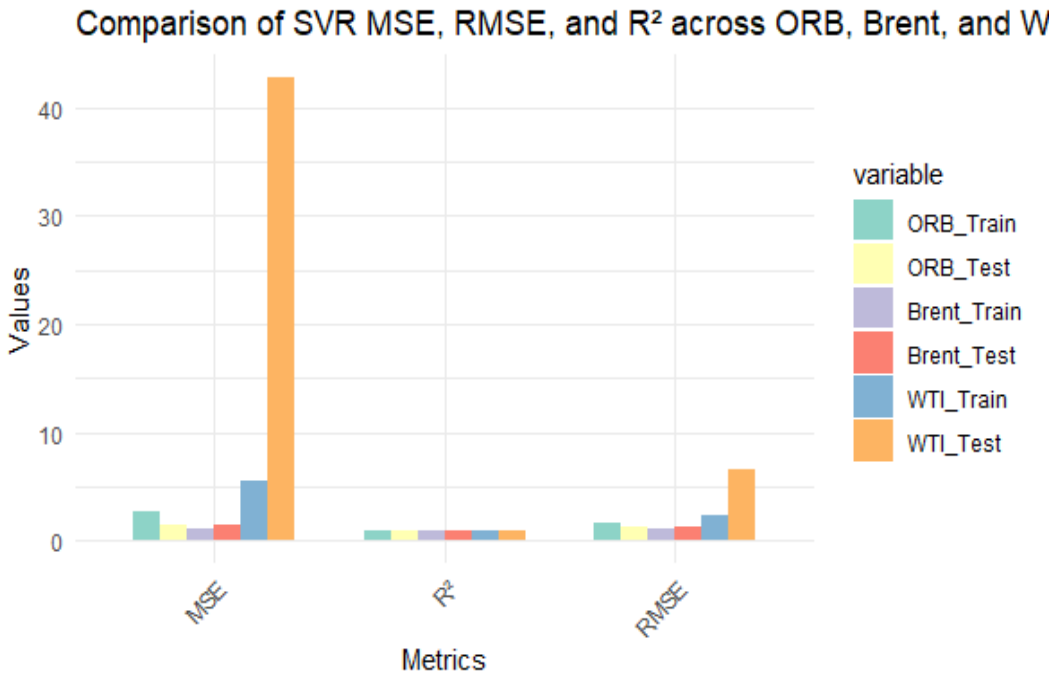


Figure 4: Bar Chart on Comparison of MSE, RMSE, and R^2 across ORB, Brent and WTI SVR models

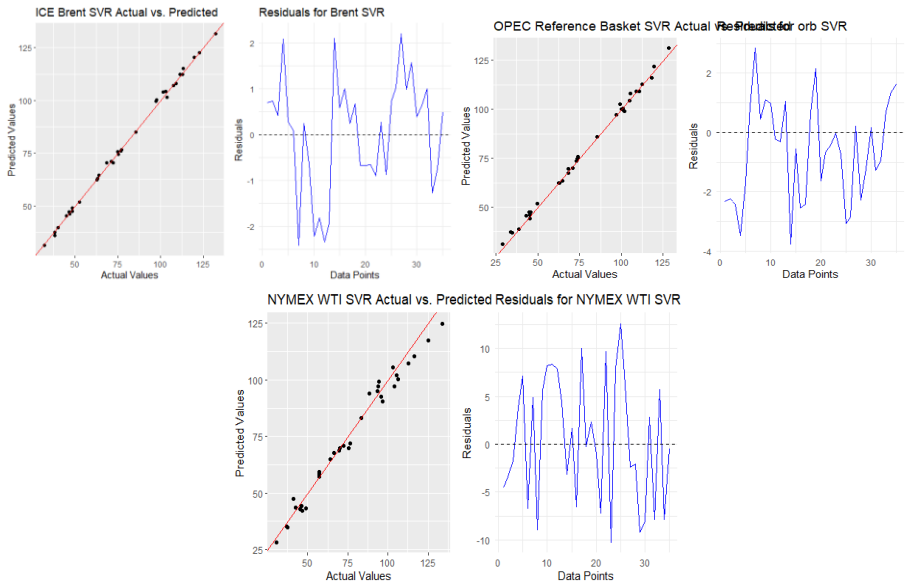


Figure 5: Plots on Predicted Values of SVR on ORB, NYMEX WTI and ICE Brent

For the OPEC Reference Basket (ORB), the actual versus predicted plot shows a good alignment between the SVR model's predictions and actual prices, with data points closely following the diagonal line (Figure 5). This suggests that the SVR model effectively captures the overall trends in ORB pricing. However, the residual plot reveals significant volatility, with wide fluctuations above and below the zero line. This indicates that while the model generally tracks price trends, it struggles to maintain consistent accuracy, especially during periods of price shifts, leading to larger prediction errors. For ICE Brent, the actual versus predicted plot similarly shows a strong alignment between predicted and actual values, indicating good model performance in forecasting Brent crude prices. The residual plot shows less volatility than the ORB, suggesting that the SVR model is more accurate for Brent pricing. The smaller fluctuations in residuals imply that the model is better at capturing Brent price movements, though some minor prediction errors remain.

In the WTI plots, the actual versus predicted values reveal that the SVR model performs quite well for NYMEX WTI prices. Most data points lie close to the diagonal line, indicating that the predicted values are fairly aligned with the actual values. This suggests that the SVR model is capable of capturing the general price trends in the WTI market. However, there are a few deviations, especially at the extreme values, implying that the model may have difficulty accurately predicting outliers or more volatile price changes. The residual plot shows the difference between the actual and predicted values. Residuals fluctuate significantly, indicating some instability in the model's predictions. The presence of positive and negative residuals means that the model occasionally underestimates or overestimates prices, with larger fluctuations seen in some data points. This suggests that while the model works reasonably well, it struggles to consistently predict prices with precision, particularly during periods of high volatility.

The SVR model performs well in predicting both ORB and Brent crude prices, with accurate alignment between actual and predicted values in both cases. However, the ORB residuals exhibit greater volatility, suggesting more significant prediction errors for this benchmark compared to Brent. For Brent, the smaller residual fluctuations imply more stable and reliable forecasts, which benefits market participants involved in Brent-linked contracts. Economically, better prediction accuracy for Brent means more precise hedging and risk management strategies, while the volatility in ORB predictions could increase uncertainty for stakeholders relying on ORB forecasts, leading to potential financial miscalculations.

The performance of XGBoost and SVR models using three metrics MSE, R^2, and RMSE are compared in Table 3 and Figure 4. Generally, the XGBoost consistently underperforms SVR across all datasets and metrics except the NYMEX WTI dataset, demonstrating lower prediction accuracy. Specifically, SVR achieves lower MSE and RMSE values, indicating better prediction accuracy. SVR's dominance is less pronounced in the WTI dataset. The superior performance of SVR over XGBoost in predicting crude oil prices has significant economic implications.

Table 3: Summary of Performance Metrics on XGBoost models using the training and test datasets

Metrics	XGBoost_ORB	XGBoost_Brent	XGBoost_WTI	SVR_ORB	SVR_Brent	SVR_WTI
MSE	3.066	2.901	2.896	1.502	1.501	42.71
RMSE	1.751	1.703	1.701	1.225	1.225	6.536
R^2	0.994	0.995	0.996	0.996	0.998	0.944

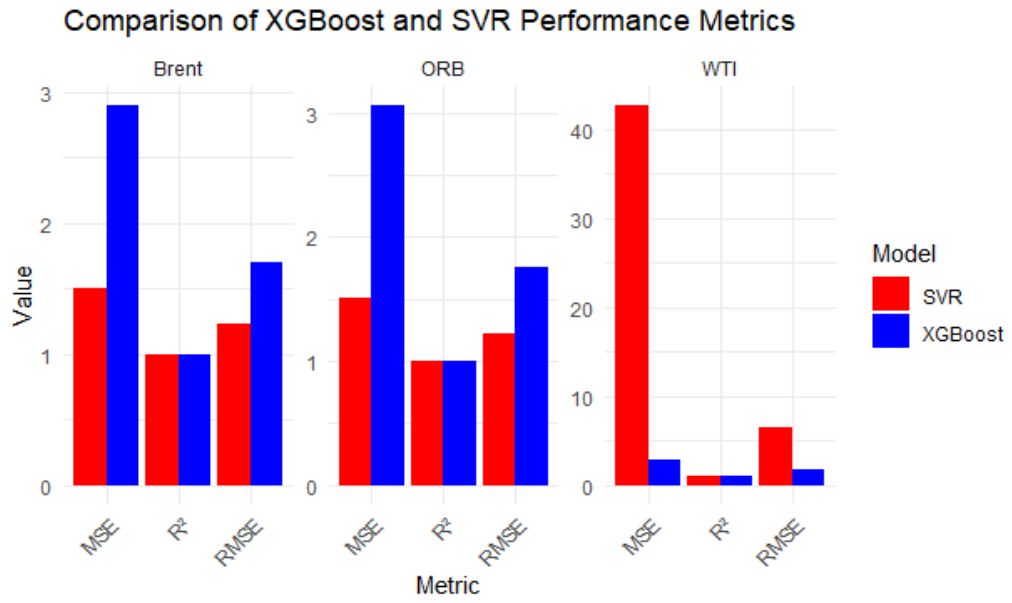


Figure 6: Comparison of XGBoost and SVR Performance Metrics using Multiple Bar Chart

Accurate forecasting of crude oil prices is crucial for energy markets, as it enables stakeholders to make informed decisions about production, consumption, investment, and hedging. XGBoost can help reduce financial risks associated with price volatility by providing more reliable predictions. Governments and policymakers can use accurate forecasts to develop effective energy policies and ensure energy security. Improved forecasting can contribute to a more efficient and stable crude oil market.

4.0 CONCLUSION

The comparison between XGBoost and SVR models reveals several strengths and weaknesses when applied to the ORB, NYMEX WTI, and ICE Brent crude oil datasets. SVR demonstrates strong performance in terms of alignment between actual and predicted values, especially for ICE Brent, where the residuals are relatively small, indicating higher predictive accuracy. However, SVR struggles with volatility, particularly in the OPEC Reference Basket (ORB) and NYMEX WTI datasets, as shown by higher fluctuations in the residuals, which leads to less stable predictions during periods of significant price swings. On the other hand, XGBoost exhibits better resilience to these fluctuations, performing well in volatile conditions due to its ability to capture non-linear patterns and complex relationships within the data. Nevertheless, XGBoost can occasionally overfit the training data, leading to slightly worse performance on the testing sets compared to SVR in some cases. Economically, these results suggest that SVR may be more suitable for stable pricing environments, while XGBoost's flexibility makes it ideal for more volatile markets. Accurately predicting crude oil prices is crucial for stakeholders involved in futures trading, hedging strategies, and policymaking, as it directly impacts investment decisions, pricing strategies, and financial planning across the global energy sector.

REFERENCES

Ani, S., & Rubaidah, D.S. (2014). Crude oil price forecasting based on hybridizing wavelet multiple linear regression model, particle swarm optimisation techniques, and principal component analysis. *The Scientific World Journal*, *3*, 854520.

Bai, H., Yuying, S., & Shonyang, W. (2021). A new two-stage approach with boosting and averaging for interval-valued crude oil prices forecasting in uncertainty environments. *Frontiers in Energy Res*earch, *19*, 1-11.

Dondukova, O., & Lin, Y. (2021). Forecasting the Crude Oil Prices Volatility with Stochastic Volatility Models, Sage Open, 1-8, DOI: 10.1177/21582440211026269.

ExxonMobil library (2021). Crude oil blends by API gravity and by sulphur content, https://corporate.exxonmobil.com/Crude-oils/Crude-trading/Crude-oil-blends-by-API-gravity-and-by-sulfur-content#APIgravity. Retrieved on 02/12/2021.

Krzysztof, D. (2021). Forecasting crude oil real prices with avenging time-varying VAR models. *Reforms Policy*, 74(C). https://doi.org/10.1016/j.resourpol.2021.102244.

Lu, Q., Sun, S., Duan, H., & Wang, S. (2021). Analysis and forecasting of crude oil price based on the variable selection –LSTM integrated model. *Energy Informatics*, *4(*41), 1-20

Merk U. (2016). Modelling Crude oil price volatility and effects of global financial crisis. *Sosyoekonomi*, *24*(29), 167-181.

Waqas, A., Muhammed, A., Umair, K., Muhammed, I., Nadeem, I., & Mukhtaj, K. (2021). A new approach for forecasting crude oil prices using median ensemble empirical model decomposition and group method of data handling. *Mathematical Problems in Engineering*, 2021:1-12. DOI: 10.1155/2021/5589717

Wajdi, H & Dawud, A. (2018). A regression analysis of determinants affecting crude oil price. *International Journal of Energy Economics and Policy*, *8*(4), 110-119.

Wassin, D., & Ibrahim, J. (2018). Predicting daily oil prices: Linear and non-linear models. *Research in International Business & Finance*, 146(C):149-165. https://doi.org/10.1016/j.ribaf.2018.01.003