A Comparative Analysis of Oil and Natural Gas Price Forecasting Using Deep Learning, Ensemble Methods, and Bayesian Optimization

Yiwen Zhang

A Thesis

in

The Department

of

Supply Chain and Business Technology Management

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Business Analytics and Technology Management (MSc) at

Concordia University

Montréal, Québec, Canada

November 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Yiwen Zhang**

Entitled: **A Comparative Analysis of Oil and Natural Gas Price Forecasting Using Deep Learning, Ensemble Methods, and Bayesian Optimization**

and submitted in partial fulfillment of the requirements for the degree of

Master of Business Analytics and Technology Management (MSc)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Danielle Morin

_____ Examiner
Dr. Rustam Vahidov

_____ Examiner
Dr. Dongliang Sheng

_____ Supervisor
Dr. Salim Lahmiri

Approved by _____
Rustam Vahidov, Chair
Department of Supply Chain and Business Technology Management

_____ 2024

_____
Anne-Marie Croteau, Dean
John Molson School of Business

Abstract

A Comparative Analysis of Oil and Natural Gas Price Forecasting Using Deep Learning, Ensemble Methods, and Bayesian Optimization

Yiwen Zhang

This study introduces a comprehensive framework for enhancing price forecasting in the oil (Brent and WTI) and natural gas (Henry Hub) markets, which play a critical role in the global economy. By integrating advanced deep learning models and ensemble methods, optimized through Bayesian Hyperparameter Optimization (BO), the research improves predictive accuracy. Utilizing an extensive dataset from January 2010 to February 2024, the models were trained and validated. Results indicate that, in the oil market, the weighted ensemble model combining LSTM and GRU performs best, leveraging the strengths of both models. In the natural gas market, post-optimization, CNN proves most effective in capturing the market's volatility and trends. XGBoost also demonstrates strong performance in both markets, balancing predictive accuracy with training efficiency. These findings offer valuable insights for risk management and decision-making in the energy sector.

Keywords: Oil price forecasting, natural gas markets, deep learning, ensemble methods, Bayesian Optimization, energy markets, predictive accuracy.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Salim Lahmiri, for his invaluable guidance and support throughout this thesis. His insights were crucial in shaping the research, and his supervision helped me develop the models and finalize the manuscript. I am also thankful to Dr. Danielle Morin, Dr. Rustam Vahidov, and Dr. Dongliang Sheng for their valuable time and feedback as part of my thesis committee. Their input was essential in refining my work.

I am also grateful to all professors and colleagues at the John Molson School of Business for their academic support and contributions to my learning experience. Their guidance has been instrumental in my academic journey and personal growth over the past two years.

Lastly, I would like to extend my heartfelt appreciation to my family. To my husband, who has been a constant source of support as we pursued our studies together at Concordia, and to my parents in China, for their continuous love and encouragement from afar.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research background

With the continuous growth of the global economy and the acceleration of industrialization, the international energy market, particularly fossil fuels, experiences significant volatility in demand and prices (S. Gupta & Pahwa, 2011). Key components of global energy consumption, such as crude oil and natural gas, have price fluctuations that directly impact the global economic structure and influence national policy-making. Increases in energy prices directly impact economic productivity, unemployment, inflation, and balance of payments equilibrium (Al-Roubaie, 2010). Moreover, geopolitical tensions, international trade policies, and currency exchange rate fluctuations further complicate these commodities' pricing. Technological advancements and market speculation also play crucial roles, as do environmental regulations and the costs associated with carbon emissions.

The complexity of forecasting in this volatile market has been underscored by recent unprecedented events, such as the COVID-19 pandemic, which have compounded these challenges by causing unprecedented shocks to the energy markets, as evidenced by the historic drop in 20 April 2020, when WTI crude oil futures dropped to a negative value at around negative 37 dollars per barrel (Lee, 2020). In addition, there has been a substantial rise in global natural gas prices since 2021 due to a combination of factors, including increased energy demand during the post-COVID economic recovery and supply constraints (Kotek, Selei, Tóth, & Felsmann, 2023). These dynamics highlight the critical need for more sophisticated forecasting tools to navigate and predict such complex

market behaviors effectively.

1.2 Motivation and research objectives

1.2.1 Motivation

Energy markets, known for their volatility and complexity, have traditionally relied on statistical methods and, more recently, machine learning models for forecasting. However, these approaches often struggle to effectively handle the multifaceted behaviors and unpredictable dynamics of such markets (Viviani, Di Persio, & Ehrhardt, 2021). Single-model techniques, in particular, frequently result in unreliable forecasts due to their inability to capture the full range of market patterns.

Recent advancements in deep learning offer sophisticated algorithms capable of uncovering intricate patterns within large datasets, yet their full potential remains untapped when used in isolation. The integration of multiple model architectures, particularly through ensemble methods, presents a promising solution. Ensemble techniques, by combining the strengths of diverse models, have been shown to reduce biases and variance, thereby improving forecasting accuracy (H. Wang, Lei, Zhang, Zhou, & Peng, 2019). This highlights a key gap in current research, where leveraging ensemble methods could offer significantly enhanced forecasting capabilities.

1.2.2 Research objectives

This study focuses specifically on deep learning and ensemble methods rather than traditional models like ARIMA and linear regression. This choice stems from the unique characteristics of energy price data. Oil and gas prices show complex, non-linear patterns with sudden changes and sharp fluctuations that simple linear models can't capture effectively. Moreover, energy prices are non-stationary data, meaning their statistical properties like mean and variance change over time. Traditional time series models like ARIMA are designed for stationary data where these properties remain constant. Although ARIMA can deal with some level of non-stationarity, it often falls short of capturing the complex and volatile patterns of energy prices (Dritsaki, 2018). In contrast, deep learning models can automatically learn and adapt to non-linear patterns. Models like LSTM and GRU are especially good at handling time-based data with changing patterns. Similarly, ensemble

methods like XGBoost and Random Forest can pick up complex relationships through their tree structures and boosting techniques.

Building upon these advantages of advanced models, the primary goal of this research is to develop a robust, weighted ensemble model that integrates predictions from various deep-learning models. This ensemble approach aims to address the shortcomings of single-model techniques by combining the strengths of multiple architectures, including One-Dimensional Convolutional Neural Networks (1D CNNs), Long Short-Term Memory Networks (LSTMs), Bidirectional LSTMs (BiLSTMs), Gated Recurrent Units (GRUs), Bidirectional GRUs (BiGRUs), and Deep Feedforward Neural Networks (DFFNNs). Additionally, tree-based ensemble methods—Extreme Gradient Boosting (XGBoost) and Random Forest—are incorporated to compare their performance and potential advantages within the forecasting framework.

Focusing on key international energy markets—Brent crude oil, West Texas Intermediate (WTI) crude oil, and Henry Hub natural gas—the research aims to optimize predictive performance using Bayesian Hyperparameter Optimization (BO). This technique will fine-tune the ensemble model by improving the efficiency of hyperparameter selection, ultimately boosting forecast accuracy.

The specific research objectives are as follows:

(1) Apply a range of deep learning and ensemble models (1D-CNNs, LSTMs, BiLSTMs, GRUs, BiGRUs, DFFNNs, XGBoost, and Random Forest) to multiple energy markets (Oil and Gas), evaluating their comparative performance in forecasting.

(2) Utilize Bayesian Hyperparameter Optimization (BO) to enhance model efficiency and accuracy through improved hyperparameter selection.

(3) Evaluate the predictive accuracy of various models to identify the most effective one for each market, offering actionable insights for improved forecasting and decision-making.

(4) Compare the optimized ensemble model against conventional forecasting methods and standalone models using historical energy market data, providing empirical evidence of the ensemble approach's effectiveness.

1.3 Main work and contributions

This research introduces a comprehensive framework specifically designed to address the complexity and volatility of energy markets, with a focus on improving forecasting accuracy in the oil and natural gas sectors. By applying a combination of advanced deep learning architectures and ensemble methods, this study aims to provide more reliable predictive tools for decision-makers in these markets. The research also leverages Bayesian Hyperparameter Optimization (BO) to systematically fine-tune model parameters, significantly enhancing performance.

The study utilized an extensive dataset spanning from January 4, 2010, to February 12, 2024, covering key oil and gas markets. The comprehensive nature of this dataset ensures the model is well-trained and validated across diverse market conditions, making the results applicable in real-world scenarios.

The main contributions of this research are as follows:

(1) Innovative Model Integration: This study applied a diverse set of deep learning models (1D CNNs, LSTMs, BiLSTMs, GRUs, BiGRUs, DNNs) and ensemble methods (XGBoost, Random Forest) to both oil and natural gas price predictions. This approach enabled the identification of market-specific model performance patterns.

(2) Bayesian Hyperparameter Optimization: The research employed BO to fine-tune model parameters, which resulted in reduced prediction errors and enhanced model performance across individual and ensemble models.

(3) Model Performance Comparison: A detailed comparison was conducted between ensemble models and standalone deep learning models, emphasizing the trade-offs in forecasting accuracy and computational efficiency across different markets.

(4) Extensive Dataset Utilization: The study utilized a robust dataset spanning from 2010 to 2024, covering oil and natural gas markets. This ensured that the models were trained and validated on real-world data, thereby enhancing their practical applicability.

(5) Practical Impact: The research led to the development of forecasting tools that provide actionable insights for policymakers, investors, and energy companies, aiding in better risk

management and decision-making processes.

Chapter 2

Literature review

Many methodologies have been explored in forecasting oil and natural gas prices, reflecting the complex nature of energy markets. These methods range from traditional statistical approaches, machine learning, deep learning, and hybrid models, illustrating the evolution of forecasting methods in response to the complex dynamics of energy markets. These methods demonstrate varying degrees of success in handling complex, non-linear, and non-stationary time series data characteristic of energy prices. Each methodology brings unique advantages and limitations, with recent trends showing a move toward combining diverse techniques to tackle the complexities of energy price forecasting effectively. Table 2.1 provides a comprehensive summary of the literature reviewed in this study, along with their respective research focus and key findings.

Previous research on energy market forecasting has focused on traditional statistical methods. The Autoregressive Integrated Moving Average (ARIMA) model is a prominent conventional statistical method applied to energy price forecasting. Alam, Murshed, Manigandan, Pachiyappan, and Abduvaxitovna (2023) utilized traditional statistical techniques such as ARIMA, Single Exponential Smoothing (SES), and K- Nearest Neighbor (K-NN) to forecast oil, coal, and natural gas prices in India before and after the COVID-19 scenario, highlighting the utility of conventional models in capturing price dynamics. The autoregressive conditional heteroskedasticity (GARCH) model also plays a crucial role in forecasting the volatility of oil prices. Haque and Shaik (2021) compared ARIMA and the Generalized GARCH model during the COVID-19 pandemic highlights ARIMA's

superior performance in forecasting under extreme market volatility, reinforcing the relevance of traditional methods in new, unpredictable scenarios. Furthermore, Kristjanpoller and Minutolo (2016) introduced a hybrid model that combines ANN with GARCH to enhance oil price volatility predictions. Their findings, which indicate a 30% improvement in accuracy over conventional models, demonstrate the potential of merging GARCH's volatility analysis with ANN's pattern recognition to tackle market unpredictability.

However, as the dynamic nature of crude oil markets became more apparent, the focus shifted towards integrating machine learning and artificial intelligence techniques to improve forecasting accuracy. With the advent of computational power, machine learning methods, including Support Vector Machine (SVM) and Random Forests (RF), have been extensively applied to price forecasting due to their efficacy in handling non-linear data. Abdollahi (2020) introduced an improved model combining Particle Swarm Optimization (PSO) with Support Vector Machine (SVM) for forecasting daily crude oil spot prices, showcasing the effectiveness of machine learning techniques. R. Gupta, Pierdzioch, and Salisu (2022) explored the predictive role of oil-price uncertainty on the UK's unemployment rate using Random Forests (RF), highlighting the application of machine learning in macroeconomic forecasts.

Deep learning allows the processing of large datasets and the learning of complex non-linear patterns. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are highlighted for their effectiveness in time series forecasting. Exploring deep learning techniques, particularly LSTM networks, marks a significant advancement in forecasting crude oil prices. Unlike traditional statistical models such as ARIMA and GARCH, which have been staples in analysing time series data for decades, deep learning approaches offer a nuanced understanding of nonlinear relationships and temporal dependencies within the data. Recent studies have demonstrated the application and efficacy of LSTM and GRU models in crude oil price forecasting. For instance, the work by Busari and Lim (2021), which integrated AdaBoost with LSTM and GRU models, showcases the potential of deep learning techniques to enhance predictive accuracy in the volatile domain of crude oil prices. Similarly, the innovative study by Huang and Deng (2021) employed Variational Mode Decomposition (VMD) in conjunction with LSTM networks, illustrating how combining advanced signal processing with deep learning can yield superior forecasting performance. The study

by Lahmiri (2024) explores various machine learning systems for forecasting fossil energy market prices. By optimizing these systems with Bayesian methods, the research evaluates the Gaussian regression process, support vector regression, regression trees, k-nearest neighbor algorithm, and deep feedforward neural networks. The findings highlight Gaussian regression's accuracy and deep learning's prediction stability.

Hybrid and ensemble methods combine multiple forecasting models and techniques to overcome the limitations of single models, aiming to improve prediction accuracy and robustness. Li, Zhu, and Wu (2019) developed hybrid forecasting models that combine Variational Mode Decomposition (VMD) with artificial intelligence techniques for monthly crude oil spot price forecasting. Incorporating multiple deep learning models in an ensemble framework has been shown to reduce overfitting risks and yield more reliable forecasts, as demonstrated by the works of Deng et al. (2023). The study by B. Wu, Wang, Lv, and Zeng (2021) that incorporated a novel approach incorporating sentiment analysis from financial news into crude oil price forecasting models, blending qualitative and quantitative data to enhance prediction accuracy, illustrates the innovative directions in hybrid modeling techniques. J. Wang, Lei, and Guo (2020) introduced a weighted hybrid model incorporating IPSS, SVR, and LSTM, showcasing the versatility of SVM in capturing non-linear patterns in energy prices. Ding, Fu, Ding, and Wang (2022) employed Random Forest (RF), XGBoost, and LightGBM in a hybrid forecasting model, demonstrating the effectiveness of ensemble methods in improving prediction accuracy.

The existing literature sets a foundation highlighting the utility of machine learning and deep learning in energy price forecasting but also reveals gaps in applying advanced optimization techniques, like BO, to fine-tune these models. Such optimization not only promises to refine model accuracy by identifying optimal configurations but also addresses critical challenges like model overfitting and computational efficiency. Besides, this study focuses on developing a comprehensive ensemble approach that leverages the unique capabilities of different deep-learning architectures to enhance forecasting accuracy. While LSTM, GRU, and CNN models are well mentioned in the literature, there is a potential research gap in systematically comparing and combining these and other deep learning architectures within an ensemble model. This research could fill this gap by identifying which architectures are most effective for different aspects of energy price forecasting

and how they can be best integrated.

Several studies have underscored the complexity and the risk of overfitting associated with advanced modeling techniques. A research opportunity exists to explore how ensemble methods, combined with BO, might alleviate these concerns. These methods can reduce overfitting and simplify model management by optimizing model parameters and selectively incorporating the most effective models into the ensemble. Lévesque, Gagné, and Sabourin (2016) notes that when enhanced by BO, ensemble methods adeptly tackle issues like overfitting and model complexity. This is achieved by refining model parameters and strategically selecting optimal models for the ensemble.

Table 2.1: Summary of literature review

Study	**Research Market**	**Data Period**	**Models**	**Findings**
Alam et al. (2023)	Oil, coal, and natural gas prices	2020-01 to 2022-05	ARIMA, SES, K-NN	Traditional statistical techniques capture price dynamics effectively.
Haque and Shaik (2021)	Oil prices	2020-02 to 2020-04	ARIMA, GARCH	ARIMA's superior performance under extreme conditions.
Kristjanpoller and Minutolo (2016)	Oil prices	2002-07 to 2014-05	Hybrid (ANN + GARCH)	The ANN-GARCH hybrid model improves forecasting accuracy by 30%.
Abdollahi (2020)	Daily crude oil spot prices	2015-06 to 2016-04	CEEMD, PSO, SVM, MS-GARCH	The proposed hybrid model improves crude oil price forecasting by better capturing nonlinearity and volatility.

Table 2.1: (continued)

Study	**Research Market**	**Data Period**	**Models**	**Findings**
R. Gupta et al. (2022)	Oil-price uncertainty and the U.K. unemployment rate	1859-09 to 2020-05	Random Forest, Lasso estimator	Random Forests outperform Lasso in forecasting unemployment rate changes.
Busari and Lim (2021)	Crude oil prices	2009-10 to 2021-06	AdaBoost, LSTM, GRU	Deep learning boosts accuracy in volatile markets.
Huang and Deng (2021)	Crude oil prices	1994-01 to 2018-07	VMD, LSTM	Advanced signal processing with deep learning improves forecasting.
Lahmiri (2024)	12 different fossil energy markets	1993-08 to 2023-06	Gaussian regression, SVR, regression trees, k-NN, DFFNN, BO	Gaussian regression delivers the best forecasts for fossil energy markets, while DFFNN provide stable predictions. Price prediction is more challenging for natural gas, coal, and propane than for crude oil.
Li et al. (2019)	Monthly crude oil spot prices	1994-01 to 2018-07	Hybrid (VMD + AI techniques)	Variational Mode Decomposition is effective in complex time series.

Table 2.1: (continued)

Study	**Research Market**	**Data Period**	**Models**	**Findings**
Deng et al. (2023)	Chinese crude oil futures	2020-01 to 2021-12	Multiple Time-frame eXtreme Gradient Boosting, Non-dominated Sorting Genetic Algorithm-II, etc.	Ensemble deep learning models reduce overfitting and offer reliable forecasts.
B. Wu et al. (2021)	Crude oil prices	2011-06 to 2019-08	Sentiment analysis-CNN; hybrid modeling-LSTM, GRU, and RNN	Text-based and big-data features enhance crude oil price forecasting.
J. Wang et al. (2020)	Daily natural gas price	2018-06 to 2019-05	IPSS, SVR, LSTM	Non-linear pattern modeling improves prediction accuracy.
Ding et al. (2022)	Crude oil	2018-03 to 2021-06	Random Forest, XGBoost, and LightGBM	Hybrid models enhance forecasting precision.
Lévesque et al. (2016)	Various datasets from the UCI Machine Learning Repository	N/A	KNN, SVM, decision trees, random forests, AdaBoost, Gaussian Naive Bayes, etc.	Ensemble optimization outperforms classical Bayesian methods.

Chapter 3

Data and methodology

The workflow and sequence of operations within the proposed model are depicted in Figure 3.1, which highlights the systematic approach to data processing, model training, and performance evaluation. After data cleaning and normalization, two primary categories of predictive methods are applied: deep learning models and ensemble models (tree-based).

In Step 2, the focus is on training and testing a set of deep learning models, including 1D CNNs, Uni/Bi-LSTMs, Uni/Bi-GRUs, and DFFNNs. Each of these models is tuned using BO to optimize their hyperparameters, ensuring that each model performs at its best. After training, the predictions from each base model are collected. A weighted ensemble method is then applied, where the final predictions are combined using a weighted average. The weights are determined by optimizing an objective function using the Sequential Least Squares Programming (SLSQP) algorithm. The goal is to find the set of weights that minimize the error metric (e.g., RMSE) and maximize the overall performance of the ensemble.

Step 3 introduces traditional ensemble models such as XGBoost and Random Forest. Similar to the deep learning models, these ensemble models are also optimized using BO.

Finally, the results from each model are evaluated using performance metrics such as RMSE, MAE, and MAPE. This step is crucial for comparing the predictive accuracy and robustness of different approaches.

In this study, we focus on exploring the advantages and disadvantages of each model. Specifically, we aim to determine whether combining multiple deep learning models in an ensemble can

improve overall performance and reduce overfitting, or if using standalone ensemble models like Random Forest or XGBoost can achieve similar results. The comparison between the weighted ensemble of deep learning models and the traditional ensemble models offers insights into which strategy is more effective for this specific predictive task. The details of each step and methodology will be discussed in the later sections.

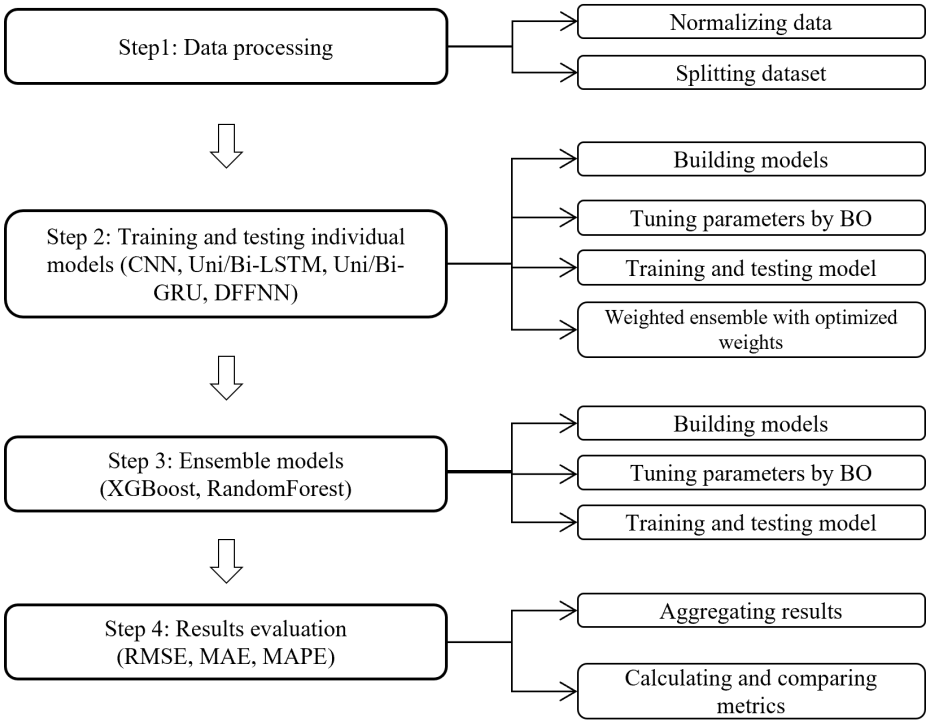


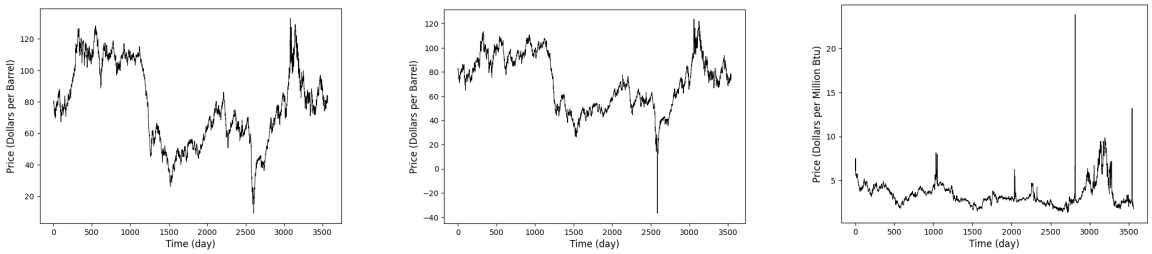
Figure 3.1: The overall workflow of the study

3.1 Data retrieval and processing

This study examined three major datasets—Brent crude oil spot price, WTI crude oil spot price, and Henry Hub Natural Gas spot price. These time series datasets span from January 4, 2010, to February 12, 2024, and feature daily observations. The data were sourced from the U.S. Energy Information Administration (EIA), which provides extensive energy statistics and can be accessed via its official website at https://www.eia.gov/. Figure 3.2 illustrates these trends graphically, while Table 3.1 presents the descriptive statistics of the daily spot prices for these datasets. Collecting over a decade of data enhances the deep learning models used in this research by providing a broad

spectrum of market conditions, which helps detect complex patterns and improve forecasts' accuracy. This extensive data collection enables the models to effectively generalize and predict future market behaviors, which is crucial for achieving robust performance in the volatile markets of oil and natural gas.

A normalization procedure is applied to scale the data between 0 and 1, ensuring consistency in data scales and facilitating efficient model training. The dataset is split into 80% for training and 20% for testing, allowing for robust training and performance evaluation on unseen data. A 30-day historical window is used to forecast one day ahead, capturing essential trends in markets like oil and natural gas. Each training instance uses a 30-day sequence to predict the next day's value, enhancing the model's ability to learn from recent behaviors. This method aligns with standard financial analysis practices and ensures reliable forecasting performance.



(a) Brent Spot Price (b) WTI Spot Price (c) Natural Gas Spot Price

Figure 3.2: Daily spot price for crude oil and natural gas

Table 3.1: The descriptive statistics of datasets

	Count	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
Brent	3571.0	77.83	25.47	9.12	57.36	75.77	103.47	133.18
WTI	3544.0	71.77	22.06	-36.98	52.62	72.48	91.19	123.64
Natural Gas	3563.0	3.42	1.36	1.33	2.60	3.02	3.99	23.86

3.2 Methodology

3.2.1 1D-CNNs

Convolutional Neural Networks (CNNs) are traditionally employed for image processing tasks due to their ability to leverage 2D input-data structures like image signals (Alzubaidi et al., 2021). However, these networks have been adapted to analyze sequential data efficiently by applying convolutions across the temporal dimension. One-dimensional Convolutional Neural Networks (1D-CNNs) utilize convolutional layers with learnable filters that are applied to one-dimensional data, allowing them to capture temporal features automatically without the need for manual feature engineering (Kim, Oh, Kim, & Choi, 2023). This adaptation enables 1D-CNNs to excel in various applications, including audio processing, natural language processing, and financial time-series analysis (Kiranyaz et al., 2021). Unlike conventional 2D-CNN architectures, which are limited to 2D inputs, 1D-CNNs are specifically designed to handle 1D signal prediction effectively.

The core operation within 1D-CNNs involves the convolution of filters across the time dimension of the input data (Srinivasamurthy, 2018):

$$f(x) = \text{ReLU}(w * x + b) \tag{1}$$

Here, x is the input sequence, w represents the weights of the convolutional filters, b is the bias, and $*$ denotes the convolution operation. The output is passed through a Rectified Linear Unit (ReLu) activation function to introduce non-linearity, enhancing the model's ability to learn complex patterns. The typical architecture is shown in Figure 3.3. First, the convolutional layers use filters that slide over the input data to extract local patterns and features, capturing relationships across the temporal dimension. After convolution, an activation function is applied to produce the initial output. This output is then processed by pooling layers, which downsample the feature maps by selecting the maximum or average values within pooling windows, thus reducing the dimensionality while retaining important information. Next, the flattened output from the convolutional and pooling layers forms a one-dimensional vector, which is fed into fully connected (dense) layers. These layers learn high-level representations of the features for classification or regression tasks. The

configuration of the output layer depends on the specific task: for time series forecasting, it may consist of a single neuron for predicting the next data point or multiple neurons for predicting multiple future values. For anomaly detection, the output layer may perform binary classification (normal vs. anomaly) or multiclass classification to categorize each time step.

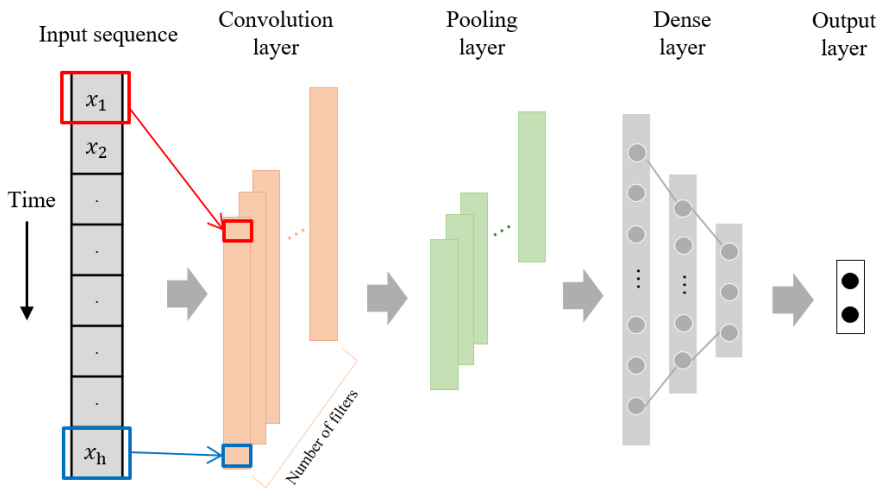


Figure 3.3: Detailed architecture of the 1D-CNN model

1D-CNNs exhibit several advantages in time series analysis, including efficient feature extraction, translation invariance, and computational speed, making them suitable for large datasets. However, these models face challenges with long-term dependencies and are sensitive to hyperparameter settings. To address these challenges, careful model design and regularization are essential. In tasks requiring a better understanding of long-range temporal patterns, combining 1D-CNNs with other architectures or incorporating techniques such as dilated convolutions may be necessary.

3.2.2 LSTMs and BiLSTMs

Long Short-Term Memory Networks (LSTMs) are a crucial component in the deep learning landscape because of their capacity to capture long-term dependencies within sequential data. Initially proposed by Hochreiter and Schmidhuber (1997), numerous researchers have extensively improved and popularised these particular types of Recurrent Neural Networks (RNNs). The architecture of LSTMs is shown in Figure 3.4.

The cell state mechanism is the core of its design, composed of a symphony of gates: the input gate (i_t), the forget gate (f_t), and the output gate (o_t), which together manage the flow of information. Weights and biases (W and b) correspond to the weight matrices and bias parameters within the network.

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \tag{2}$$

Here, the forget gate (f_t) regulates the flow of information from the current input and the previous hidden state (h_{t-1}) by applying a sigmoid activation function, selectively keeps or discarding previous information, guaranteeing that the cell state (C_t) is a dynamic representation of learned data over time.

The following step is to use the input gate (i_t) to manage the patterns of data flowing from the current input and the previous hidden states. Cell State (C_t) updates with contributions from the input gate, forget gate, previous cell state (C_{t-1}), and the transformed input using the tanh function.

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{4}$$

Finally, the output gate (O_t) determines the necessary information for the activation function of the output value (h_t).

$$O_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \tag{5}$$

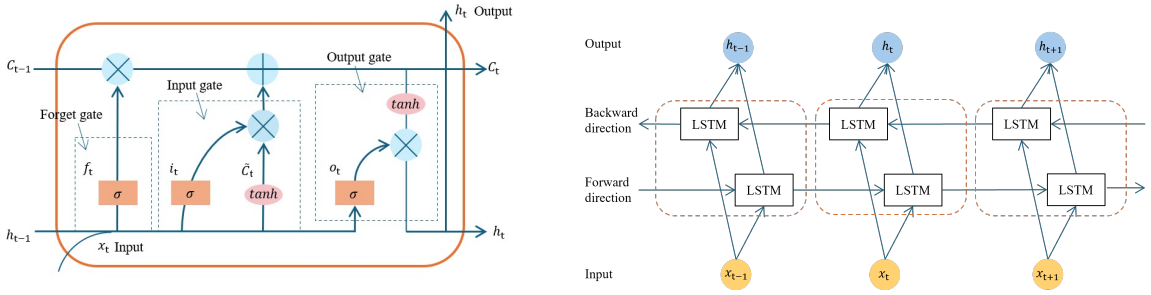
$$h_t = O_t \tanh(C_t) \tag{6}$$

Enhancing the LSTM's capability, BiLSTMs employ a forward and backward pass through the data. This approach allows the network to capture information from both past and future contexts. This bidirectional sweep is pivotal for predictions where understanding the entire temporal context is beneficial. The BiLSTM modifies the LSTM's structure by introducing two separate hidden states for each time step:

$$\overrightarrow{h_t} = \overrightarrow{\text{LSTM}}(x_t, \overrightarrow{h_{t-1}}) \tag{7}$$

$$\overleftarrow{h_t} = \overleftarrow{\text{LSTM}}(x_t, \overleftarrow{h_{t+1}}) \tag{8}$$

Subsequently, the model joins these states, $\overleftarrow{h_t}$ and $\overrightarrow{h_t}$, to provide a comprehensive output that captures data in both directions, improving the model's predicted accuracy. The BiLSTMs model's structure is shown in Figure 3.4.



(a) LSTM (b) BiLSTM

Figure 3.4: Framework of the LSTM and BiLSTM models

3.2.3 GRUs and BiGRUs

Gated Recurrent Units (GRUs) and their bidirectional counterparts (BiGRUs) are integral components of modern sequence modeling within deep learning frameworks. Shen, Tan, Zhang, Zeng, and Xu (2018) demonstrated the superior performance of GRU-based models in financial time series predictions. The GRU layer, like the LSTM layer, is a type of recurrent neural network that addresses the vanishing gradient problem and can learn long sequences of data (Faraji, Sadrossadat, Na, Feng, & Zhang, 2023). However, the GRU has fewer parameters than the LSTM, leading to better model accuracy (Faraji et al., 2023). This makes GRUs suitable for use in systems where computing performance is crucial.

The primary distinction between GRU and LSTM lies in their approach to managing memory cell state. In LSTM, the memory cell state is maintained separately from the hidden state and is updated through three gates: the input gate, the output gate, and the forget gate. Conversely, GRU simplifies this by using Candidate Activation Vector ($\tilde{h}_t$) instead of a distinct memory cell state and updates it with two gates: the reset gate (r_t) and the update gate (z_t). The reset gate controls how much of the previous hidden state should be forgotten, while the update gate determines the

extent to which the candidate activation vector should be incorporated into the new hidden state. The equations are represented as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{9}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{10}$$

$$\tilde{h}_t = tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t]) \tag{11}$$

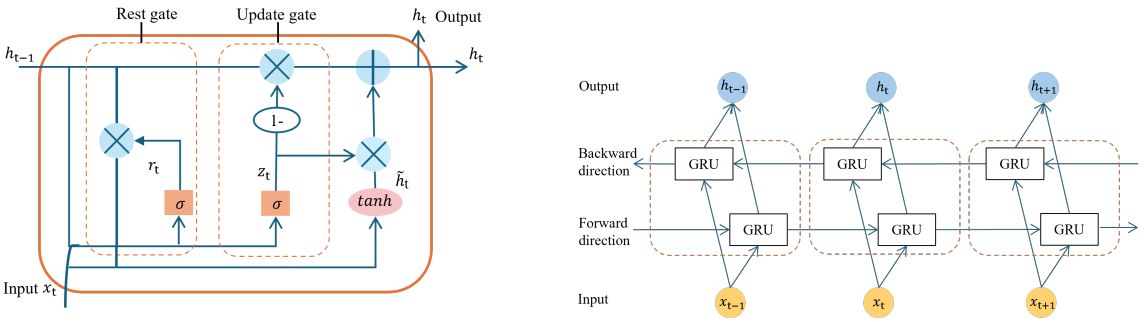
$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \tag{12}$$

Here, the update gate (z_t) determines how much the unit's activation or state should be updated. The associated reset gate (r_t) controls the influence of the previous state on the current computation, allowing GRUs to dynamically discard or retain information across time steps. The candidate activation vector ($\tilde{h}_t$) is computed from the current input (x_t) and a modified version of the previous hidden state that is adjusted by the reset gate. The new hidden state (h_t) is then derived by combining the candidate activation vector with the previous hidden state (h_{t-1}), with the contribution of each weighted by the update gate.

BiGRUs enhance the GRU's functionality by processing data in both forward and backward directions, encapsulating a broader temporal context. By combining the forward and backward, BiGRUs enhance the GRU's functionality by processing data in both forward and backward directions, encapsulating a broader temporal context, as shown in Figure 3.5. The formulas for the BiGRU model are as follows:

$$\overrightarrow{h_t} = \overrightarrow{\text{GRU}}(x_t, \overrightarrow{h_{t-1}}) \tag{13}$$

$$\overleftarrow{h_t} = \overleftarrow{\text{GRU}}(x_t, \overleftarrow{h_{t+1}}) \tag{14}$$



(a) GRU (b) BiGRU

Figure 3.5: Framework of the GRU and BiGRU models

3.2.4 DFFNNs

Deep feedforward neural networks (DFFNNs), also known as deep neural networks (DNNs), are a widely used type of artificial neural networks (ANNs) with multiple hidden layers, designed to model complex functions by increasing network depth (Lau, Phang, & Lim, 2019). They are useful in many different applications, including image classification (Lau et al., 2019), speech synthesis (Z. Wu, Valentini-Botinhao, Watts, & King, 2015), and financial market prediction (Dixon, Klabjan, & Bang, 2015). DFFNNs typically consist of multiple hidden layers, each containing numerous neuron nodes. The architecture ensures that each neuron in one layer is connected to every neuron in the next layer. This structure helps the network learn the best weights for representing the data, which is essential for tasks such as classification or regression. However, the performance of DFFNNs can be influenced by various hyper-parameters, such as the activation function, dropout regularization, and network architecture (Koutsoukas, Monaghan, Li, & Huan, 2017). This study tests three activation functions in the BO process: rectified linear unit (relu) function, sigmoid function, and hyperbolic tangent (tanh) function.

The DFFNNs model consists of an input layer, multiple hidden layers ($k \geq 2$), and an output layer. This structure, depicted in Figure 3.6, showcases how the DFFNN model processes input data through interconnected layers, adjusting weights to learn and predict outcomes effectively.

The input layer receives the initial data, represented by the input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$,

where n denotes the number of input features. The data is then passed through a series of transformations in the hidden layers. For each hidden layer k, the input undergoes a linear transformation followed by a nonlinear activation function. Specifically, the linear transformation involves multiplying the input by a weight matrix $\mathbf{W}_k$ and adding a bias vector $\mathbf{b}_k$. This results in the pre-activation value $\mathbf{z}_k$. The pre-activation value is then passed through an activation function f to produce the activation output $\mathbf{h}_k$. This process can be summarized by the equations:

$$\mathbf{z}_k = \mathbf{W}_k \mathbf{h}_{k-1} + \mathbf{b}_k \tag{15}$$

$$\mathbf{h}_k = f(\mathbf{z}_k) \tag{16}$$

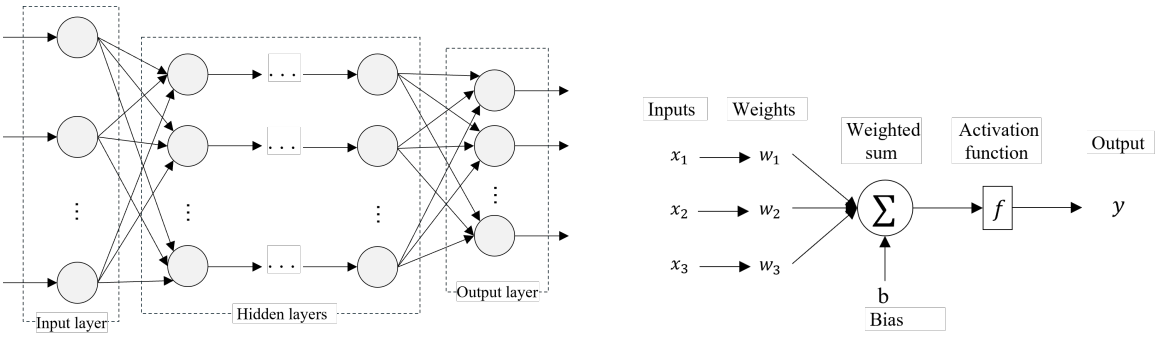
Where $\mathbf{h}_{k-1}$ is the output from the previous layer (or the input vector $\mathbf{x}$ for the first hidden layer).

Finally, the output layer takes the output from the last hidden layer $\mathbf{h}_{L-1}$ as its input and applies a similar transformation. The pre-activation value for the output layer is computed as:

$$\mathbf{z}_L = \mathbf{W}_L \mathbf{h}_{L-1} + \mathbf{b}_L \tag{17}$$

This pre-activation value is then passed through an activation function g to produce the final output $\mathbf{y}$:

$$\mathbf{y} = g(\mathbf{z}_L) \tag{18}$$



(a) (b)

Figure 3.6: Architecture of the DFFNN network

3.2.5 XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced machine learning technique that improves the performance of decision trees through a boosting framework. It improves upon traditional Gradient Boosting Decision Tree (GBDT) approaches. XGBoost and GBDT possess the same basic algorithmic structure, with XGBoost being an implementation of GBDT (Zhang, Li, & Jiao, 2019), designed to improve both the speed and accuracy of predictions. XGBoost has emerged as a powerful tool for predictive tasks, particularly in handling large-scale datasets. Its effectiveness has been demonstrated in various domains, making it a powerful tool for predictive tasks like price forecasting (Ding et al., 2022; Gono, Napitupulu, & Firdaniza, 2023). XGBoost consistently outperforms traditional regression models in terms of accuracy and efficiency(dairu & Zhang, 2021). Its scalability allows it to process billions of examples using fewer resources than existing systems(Chen & Guestrin, 2016).

XGBoost builds an ensemble of "weak learners" through an iterative process. Each new decision tree is trained to correct the errors (residuals) made by the previous trees, thus improving the model's accuracy (Ding et al., 2022). One of the key features of XGBoost is its regularized objective function, which combines a loss function that measures prediction error with a regularization term to limit model complexity. By controlling the number of leaves in each tree and adjusting the weights of the leaf nodes, XGBoost helps prevent overfitting and improves the model's ability to generalize (Chen & Guestrin, 2016).

The algorithm starts by making an initial prediction, usually the average of the target variable. It then calculates the residuals, which are the differences between the actual and predicted values, and constructs decision trees to minimize these residuals. As shown in Figure 3.7, XGBoost sequentially builds multiple decision trees. The final prediction is the weighted sum of the predictions from all the trees, as shown in the following equation:

$$\hat{y} = \sum_{k=1}^{K} f_k(x_i) \tag{19}$$

where $\hat{y}$ is the final prediction, K is the number of trees, and $f_k(x_i)$ is the prediction from the k-th tree for input x_i. To prevent overfitting, XGBoost incorporates a regularization term into the

objective function:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{20}$$

Where $l(y_i, \hat{y}_i)$ represents the loss function measuring the difference between the actual value y_i and predicted value $\hat{y}_i$, and $\Omega(f_k)$ is the regularization term controlling the complexity of the k-th tree.

The regularization term helps prevent overfitting and is defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{21}$$

Where T is the number of leaf nodes in the tree, w_j is the weight of the j-th leaf, γ controls the penalty on the number of leaves, and λ is the L2 regularization term that penalizes large leaf weights.

This regularization encourages simpler trees and reduces the risk of overfitting, especially when dealing with noisy or high-dimensional data. By incorporating both the loss function and regularization, XGBoost ensures a balance between model accuracy and generalization.

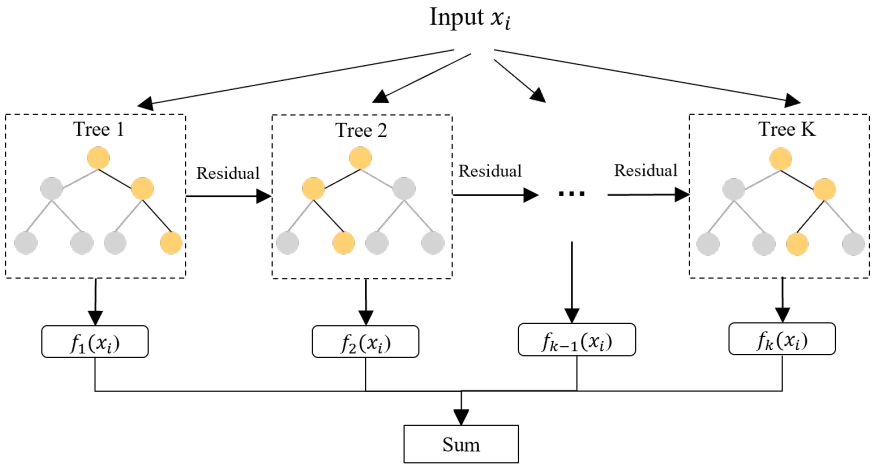


Figure 3.7: Overview of the XGBoost regression tree model

3.2.6 Random Forest

Random Forest is a bagging-based ensemble method that builds multiple decision trees using bootstrap sampling and random feature selection (Liaw, 2002). It combines the predictions of individual trees, and the final prediction is made by averaging the predictions from all trees (for regression tasks) or by majority vote (for classification tasks) (Talekar, 2020), as shown in Figure 3.8. This approach reduces overfitting and improves prediction accuracy compared to single decision trees (Liaw, 2002).

It introduces randomness by training each tree on a bootstrap sample of the data and selecting a random subset of features for splitting at each node. For regression problems, the final prediction is the average of all the trees:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_t(x) \tag{22}$$

where T is the number of trees, and $f_t(x)$ is the prediction from the t-th tree.

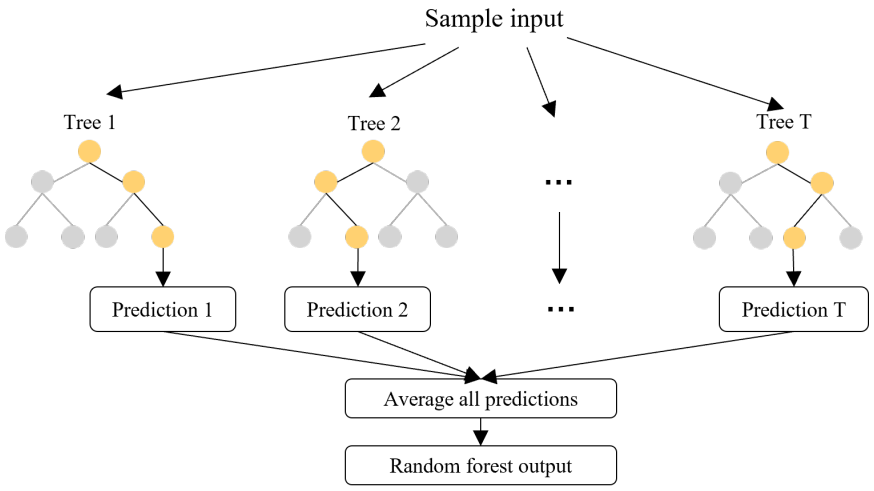


Figure 3.8: Overview of the random forest algorithm

3.2.7 Parameter tuning using Bayesian optimization (BO)

Bayesian optimization (BO) has emerged as a compelling strategy for hyperparameter tuning in machine learning, proving to be highly beneficial for optimizing complex deep neural networks. As evidenced in the findings of the Black-Box Optimization Challenge 2020 (Turner et al., 2021),

BO demonstrates a superior capability for tuning hyperparameters across various machine-learning models, outperforming random search, showing better efficiency and effectiveness in navigating complex hyperparameter spaces.

BO employs a sophisticated probabilistic model based on Gaussian Processes (Rasmussen, 2003), utilizing Bayes' theorem (Brochu, Cora, & De Freitas, 2010) to navigate and optimize the hyperparameter search space systematically. This search space typically includes crucial hyperparameters, such as the number of neurons, learning rates, and network depth, which depend on the structure of machine learning models.

At the core of BO's methodology is the application of Bayes' theorem, which is expressed mathematically as:

$$p(f \mid D) = \frac{p(D \mid f) \cdot p(f)}{p(D)} \tag{23}$$

Here, $p(f \mid D)$ represents the posterior probability of the function f after observing the data D, $p(D \mid f)$ is the likelihood of observing the data given the function, $p(f)$ denotes the prior belief about the distribution of function f, and $p(D)$ serves as the normalizing constant.

BO employs Bayes' theorem to refine its model of hyperparameter performance iteratively, integrating data from new evaluations. This probabilistic updating process enables BO to recalibrate the search space exploration dynamically. It adeptly balances the exploration of unexplored parameter configurations with the exploitation of regions known to produce favorable outcomes. This strategic balance is critical as it directs BO's computational efforts towards the most promising areas of the search space, thereby increasing sample efficiency.

Moreover, the strategic utilization of BO has been demonstrated to significantly reduce the Mean Square Error (MSE) and enhance the overall performance of models across various settings (Victoria & Maragatham, 2021). L. Wang, Dernoncourt, and Bui (2020) elaborated on a unified Bayesian optimization framework designed to optimize machine learning models for both predictive effectiveness and training efficiency. This framework underscores how sophisticated optimization techniques like BO can revolutionize the field of machine learning by boosting efficiency and diminishing the resources required for model training.

3.3 Experimental setting

We applied BO to explore the best hyper-parameters for each model. Table 3.2 displays the hyperparameters with the range values for each model studied in this paper.

Table 3.2: Experimental parameter values for BO by exploring different ranges

Model	**Parameters**	**Value range and step**
Conv1D	Filters	(16, 256, 16)
	Kernel size	[3, 5]
	Dense units	(8, 128, 8)
	Learning rate	(0.0001, 0.01, Log-uniform)
Uni/BiLSTM	Units	(16, 256, 16)
	Learning rate	(0.0001, 0.01, Log-uniform)
Uni/BiGRU	Units	(16, 256, 16)
	Learning rate	(0.0001, 0.01, Log-uniform)
DFFNN	Hidden layers	(1, 5)
	Network neurons	(10, 100)
	Activation	(0, 2)
XGBoost	Max depth	(3, 10)
	Learning rate	(0.01, 0.3, Log-uniform)
	N estimators	(100, 1000)
	Min child weight	(1, 10)
	Subsample	(0.5, 1.0, Uniform)
	Colsample bytree	(0.5, 1.0, Uniform)
RandomForest	N estimators	(100, 1000)
	Max depth	(3, 20)
	Min samples split	(2, 10)
	Min samples leaf	(1, 4)
	Max features	['sqrt', 'log2', None]
	Bootstrap	[True, False]

Note: activation (0, 2) - Index for activation functions; maps to relu, sigmoid, tanh.

3.4 Evaluation criteria

A range of evaluation metrics are used for time series forecasting, with Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) being the most common (Astatkie, 2006; Mahmoud & Pegels, 1990). These metrics, of which MAPE is a relative measure while RMSE and MAE are absolute values, are used to compare the performance of various models.

- Root Mean Square Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{24}$$

- Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{25}$$

- Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{26}$$

In these equations, y_i is the actual observed value for the i -th observation; $\hat{y}_i$ is the predicted value for the i-th observation.

Chapter 4

Experimental results

In this chapter, we evaluate and compare the forecasting performance of nine different models before and after applying BO. These include six individual deep learning models (i.e., 1D-CNN, LSTM, BiLSTM, GRU, BiGRU, and DFFNN), two classic tree-based ensemble models (i.e., XGBoost and Random Forest), and one weighted ensemble model that combines all the deep learning models. We also provide a comparative analysis of the training times for each model, both before and after the application of BO. To demonstrate the effectiveness of these methods, we present detailed graphs depicting the actual versus predicted data for three distinct datasets.

4.1 Performance evaluation

Table 4.1 presents the forecasting performance of various models for Brent Oil using three evaluation metrics: MAE, RMSE, and MAPE, before and after Bayesian Optimization (BO). Overall, most models improved all three metrics after BO, indicating that the optimization enhanced their predictive accuracy.

Looking at RMSE, which reflects the average prediction error, several models experienced notable improvements. The models that showed the most improvement were LSTM, BiLSTM, 1D-CNN, GRU, and XGBoost. For example, the RMSE of LSTM dropped from 3.726 to 2.504, a significant reduction of about 32.8%.

After BO, the Weighted Ensemble model achieved the best performance, with the lowest RMSE

of 2.289 and the shortest blue bars among all models, as shown in Figure 4.1. In this step, the Sequential Least Squares Programming (SLSQP) algorithm was used to minimize the mean squared error (MSE) between the actual values and the weighted sum of model predictions. The algorithm adjusted the weights of 1D-CNN, LSTM, BiLSTM, GRU, BiGRU, and DFFNN, with the constraint that the weights sum to 1. The final optimized weights were LSTM: 0.2701, BiLSTM: 0.1595, and GRU: 0.5704. This suggests that combining multiple models (ensemble) helped capture complex patterns better than individual models, resulting in lower prediction errors. The next best individual model was GRU, with an RMSE of 2.299. XGBoost performed better than Random Forest, with an RMSE of 2.317. XGBoost is known for its strong predictive performance, especially with complex and nonlinear data, making it a solid choice for this task. On the other hand, the worst-performing model post-BO was DFFNN with an RMSE of 3.836, indicating that this model struggled to improve despite optimization. This suggests that DFFNN may not be as effective for this type of time-series data or that its structure requires further refinement.

Table 4.1: Forecasting performance comparison of Brent Oil

Models performance	**Pre-BO**			**Post-BO**		
	RMSE	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
1D-CNN	3.754	2.720	3.050%	2.905	1.981	2.218%
LSTM	3.726	2.666	3.010%	2.504	1.751	1.979%
BiLSTM	4.523	3.250	3.690%	3.059	2.480	2.805%
GRU	2.876	2.046	2.290%	2.299	1.655	1.851%
BiGRU	2.768	1.977	2.210%	2.606	1.846	2.090%
DFFNN	3.292	2.297	2.570%	3.836	2.740	3.072%
Weighted Ensemble	3.276	2.314	2.600%	**2.289**	**1.637**	**1.833%**
XGBoost	2.736	1.997	2.230%	2.317	1.693	1.898%
Random Forest	**2.377**	**1.737**	**1.950%**	2.356	1.713	1.925%

Notes: Bold numbers show the model with the best forecasting performance.

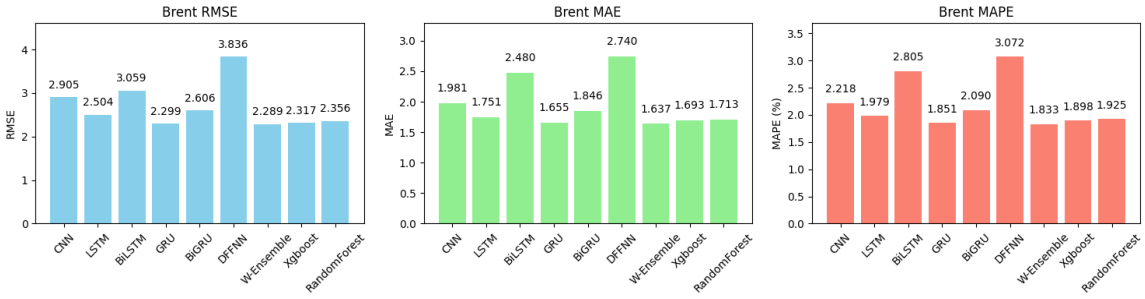


Figure 4.1: Model performance evaluation (post-BO) - Brent Oil

For WTI Oil (See Table 4.2 and Figure 4.2), before BO, the models exhibited varying performance. Random Forest had the lowest RMSE at 2.491, likely due to its ability to capture non-linear patterns through its ensemble structure. GRU also performed well with an RMSE of 2.720, reflecting its strength in handling sequential data. In contrast, more complex models like BiLSTM and BiGRU had higher RMSEs, possibly due to overfitting or the need for further tuning.

After applying BO, the Weighted Ensemble model once again achieved the best results, with the lowest RMSE of 2.207, confirming the value of ensemble methods in improving accuracy. The optimal weights for the weighted ensemble model were LSTM: 0.2916 and BiGRU: 0.7084, indicating a stronger contribution from BiGRU in the ensemble's overall prediction. DFFNN, however, continued to perform poorly, with an RMSE of 4.914, indicating it is not well-suited for this task without further refinement.

In summary, the analysis shows that simpler models like Random Forest and GRU were more robust and performed better pre-BO, while more complex models like BiLSTM and BiGRU needed optimization to reach their full potential. After BO, weighted ensemble models and GRU-based models excelled in the oil market predictions, while simpler neural networks like DFFNNs face difficulties in achieving competitive results.

Table 4.2: Forecasting performance comparison of WTI Oil

Models performance	**Pre-BO**			**Post-BO**		
	RMSE	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
1D-CNN	3.477	2.525	3.010%	4.061	3.054	3.720%
LSTM	4.139	3.105	3.680%	2.855	2.323	2.789%
BiLSTM	3.778	2.725	3.270%	3.133	2.261	2.729%
GRU	2.720	2.053	2.430%	2.363	1.804	2.157%
BiGRU	3.348	2.608	3.130%	2.266	1.610	1.923%
DFFNN	3.508	2.658	3.190%	4.914	4.010	5.013%
Weighted ensemble	3.307	2.452	2.930%	**2.207**	**1.604**	**1.905%**
XGBoost	3.112	2.118	2.440%	2.637	1.813	2.094%
Random Forest	**2.491**	**1.796**	**2.100%**	2.474	1.764	2.057%

Notes: Bold numbers show the model with the best forecasting performance.

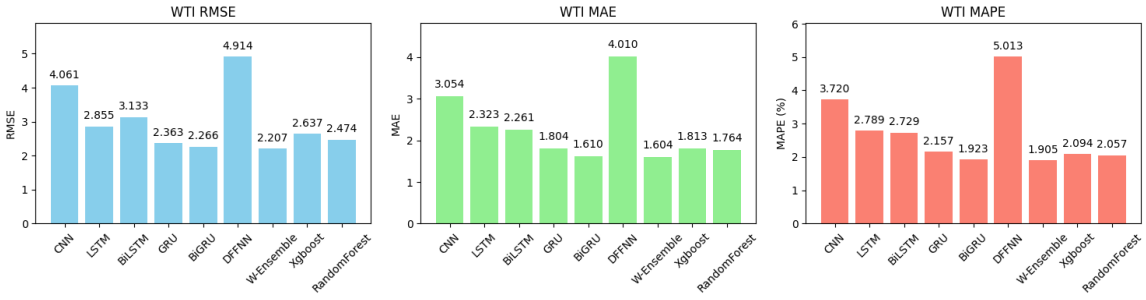


Figure 4.2: Model performance evaluation (post-BO) - WTI Oil

Table 4.3 and Figure 4.3 show the forecasting performance for the natural gas market. In the natural gas market, 1D-CNN stands out as the most effective model after BO, with the lowest RMSE at 0.627, indicating it adapted well to this market. XGBoost had a significant improvement, with its RMSE decreasing by 45.5%, although it did not outperform Random Forest. On the other hand, DFFNN performed exceedingly before BO, however, it experienced a decline in performance, with its RMSE increasing to 0.685 post-BO, suggesting it is not suitable for this task without further refinement.

Compared with the performance of machine learning models in the Brent and WTI Oil markets,

the natural gas market favored models like 1D-CNN, which performed moderately in the oil market but excelled post-BO for natural gas forecasting. On the other hand, models like BiGRU and Random Forest remained reliable across both markets, while DFFNN consistently underperformed, showing limitations in handling both tasks.

Shifting the focus to MAPE, this metric is particularly useful for comparing models across different datasets, as it normalizes errors relative to the true values, making it independent of scale. The results show that MAPE values are generally lower in the oil market compared to the natural gas market, suggesting that the oil market is more predictable and that the proposed models are better suited for forecasting oil prices. Conversely, the higher MAPE values in the natural gas market reflect greater complexity and volatility, making it more difficult to capture consistent patterns. This underscores the need for further tuning or alternative modeling approaches to better address the inherent complexities of natural gas price movements.

Table 4.3: Forecasting performance comparison of Natural Gas

Models performance	**Pre-BO**			**Post-BO**		
	RMSE	**MAE**	**MAPE**	**RMSE**	**MAE**	**MAPE**
1D-CNN	0.666	0.360	7.400%	**0.627**	**0.293**	**6.679%**
LSTM	0.755	0.440	8.840%	0.794	0.326	7.383%
BiLSTM	0.819	0.450	8.330%	0.778	0.298	6.124%
GRU	0.787	0.426	7.990%	0.702	0.268	5.523%
BiGRU	0.669	0.330	6.280%	0.662	0.261	5.644%
DFFNN	**0.590**	**0.261**	**6.200**%	0.685	0.371	7.653%
Weighted ensemble	0.669	0.350	6.810%	0.710	0.291	5.865%
XGBoost	1.404	0.663	10.910%	0.765	0.392	7.052%
Random Forest	0.962	0.427	7.530%	0.702	0.379	7.557%

Notes: Bold numbers show the model with the best forecasting performance.

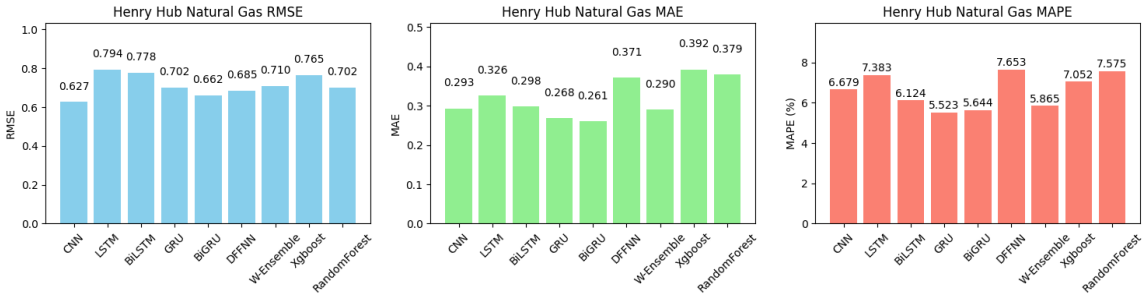


Figure 4.3: Model performance evaluation (post-BO) - Natural Gas

4.2 Training time analysis

In this study, the proposed models were trained using a Google Colab environment with an L4 GPU as the hardware accelerator. The training time for machine learning models, especially deep learning models can vary greatly depending on the hardware used.

Figure 4.4 compares the training time of different models across three datasets—Brent, WTI, and Henry Hub Natural Gas—both before and after BO. After BO, where 30 iterations of parameter combinations were tried. For most models, training time after BO is significantly longer than before BO, reflecting the computational cost of hyperparameters tuning. This is a common trade-off in model optimization—improved performance often comes at the cost of increased training time.

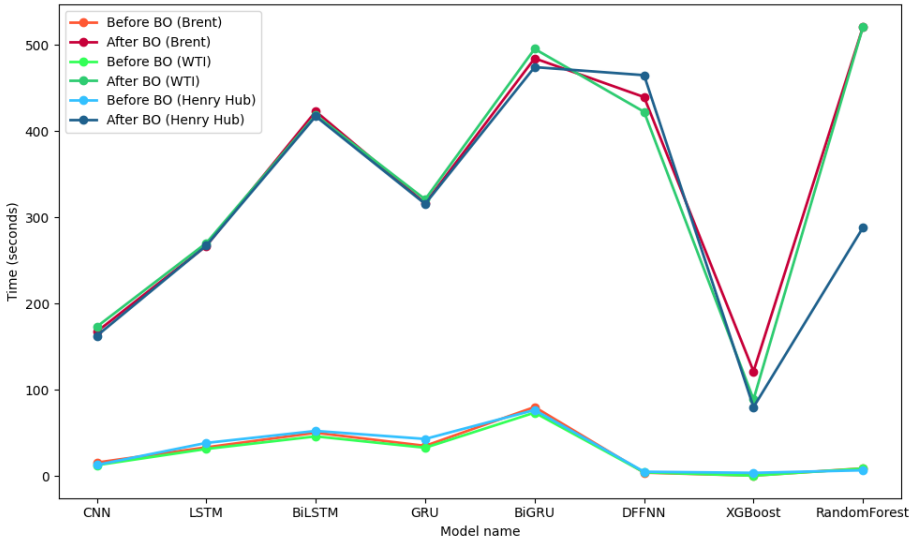


Figure 4.4: Training time comparison for Brent, WTI, and Henry Hub Natural Gas

Random Forest stands out with the highest training time after BO, exceeding 500 seconds in the oil market. The reason for this is that Random Forest, being an ensemble method with many trees, is computationally expensive, and optimizing hyperparameters such as the number of trees or the depth of trees further adds to this. BiGRU also exhibits a longer processing time, likely due to the complexity of the model. Bidirectional networks like BiGRU and BiLSTM require more computations, as they process information in both forward and backward directions.

In addition, DFFNN has a much longer training time after BO, nearly matching the time taken by BiGRU, as seen in the graph. The training process takes longer because the network can have many parameters to adjust, particularly when using multiple hidden layers with many neurons. During the BO process, every time the hyperparameters change, the model needs to be completely retrained, updating all these parameters through multiple layers. Additionally, since DFFNN has limitations in processing sequential data like energy prices, it might need more time to learn the patterns. These factors could explain why the training takes so long, even though there are not many hyperparameters to tune.

Models like LSTM, 1D-CNN, and GRU see relatively moderate increases in training time after BO. This suggests that while these models benefit from optimization, their computational complexity is not as high as more complex architectures like BiGRU or Random Forest. Hence, the time cost is more manageable.

XGBoost demonstrates impressive efficiency in this analysis. Despite having the shortest average training time of around 100 seconds post-BO, XGBoost delivers strong predictive performance. Its relatively low training time highlights its computational efficiency, a key advantage of tree-based models like XGBoost, especially when combined with boosting techniques that reduce errors without requiring excessive computational resources.

In summary, considering both error performance and training time, GRU stands out as the most cost-effective model because it offers a substantial reduction in error metrics while keeping training time relatively low compared to more complex models like BiGRU and Random Forest. LSTM and 1D-CNN also provide a reasonable balance, with moderate increases in training time and decent improvements in performance, making them good alternatives when time efficiency is crucial. BiGRU and Random Forest, while performing well post-BO, may not be as time-efficient due to their

longer training times. However, despite this increased training time, DFFNN's predictive performance post-BO remained poor, with high error rates (as discussed earlier). This suggests that the extended training time does not translate into improved performance, making DFFNN an inefficient choice both in terms of time and accuracy.

4.3 Prediction visualization

In this section, we present a comparison between the predicted and actual values for the test set across three key markets: Brent, WTI, and Natural Gas, as shown in Figures 4.5, 4.6, and 4.7. For each market, we highlight the prediction plots of the top three performing models. These plots illustrate how well the models align with the actual value patterns, showcasing their impressive predictive capabilities. The models consistently capture the trends and fluctuations in each market, reflecting their robustness and accuracy in handling complex time series data. This alignment with actual values underscores the effectiveness of these models in real-world market prediction.

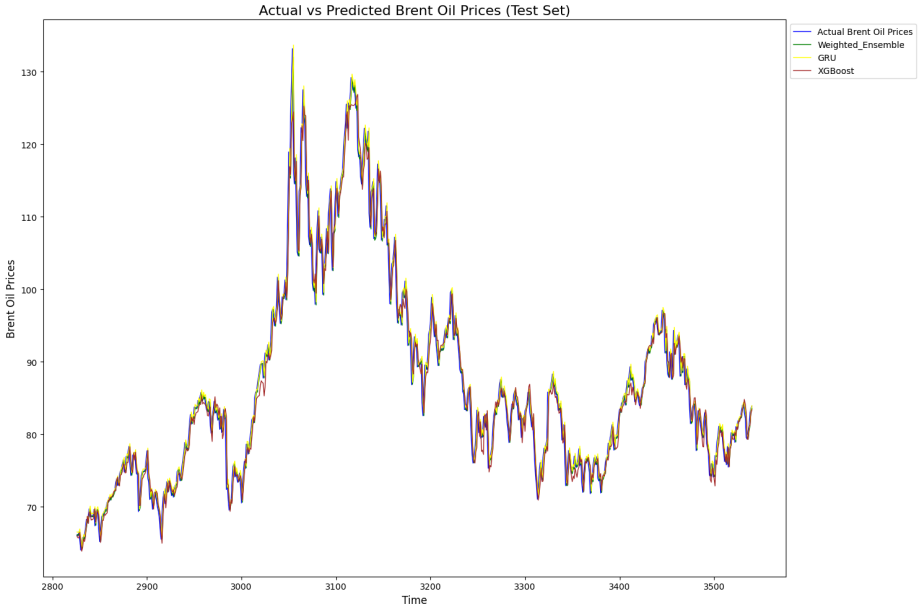


Figure 4.5: Actual and predicted graph of Brent Oil Prices

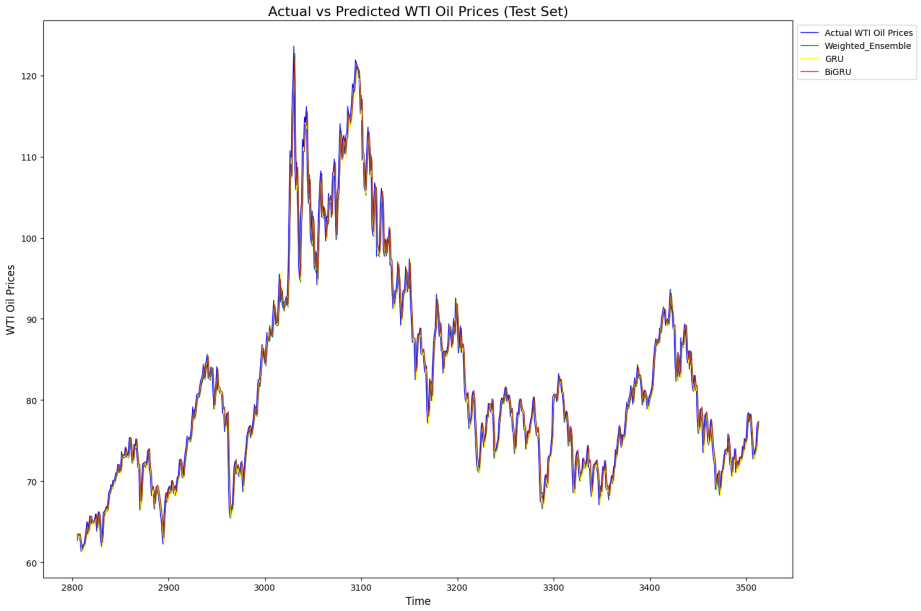


Figure 4.6: Actual and predicted graph of WTI Oil Prices

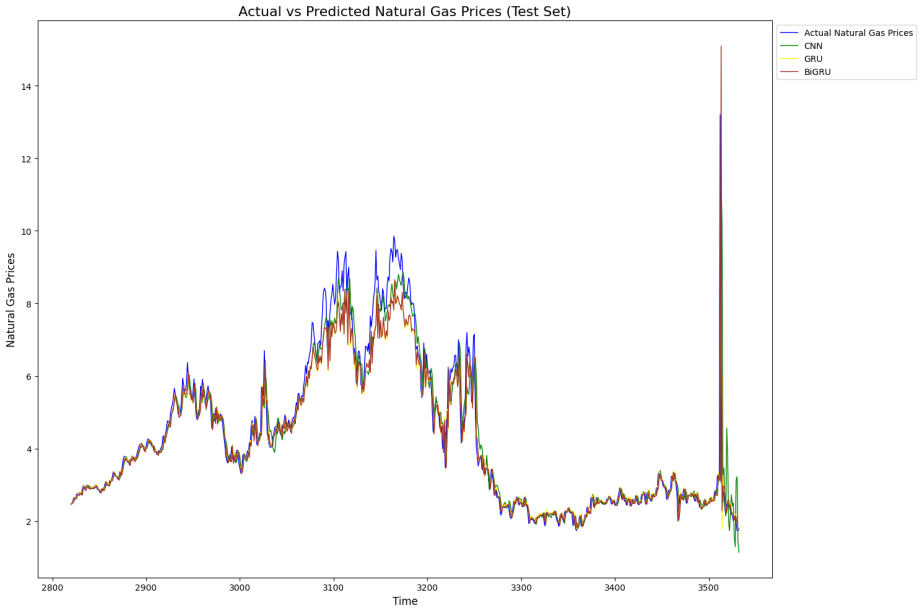


Figure 4.7: Actual and predicted graph of Natural Gas Prices

Chapter 5

Discussion

5.1 Findings analysis

This study tackles the challenge of forecasting prices in the highly volatile oil and natural gas markets, which play a crucial role in the global economy. Price fluctuations in these markets have significant impacts on national economies, energy policies, and investment decisions. The models developed in this research aim to enhance forecasting accuracy, offering valuable insights for risk management and informed decision-making.

In the oil market, the weighted ensemble model, combining LSTM and GRU, emerged as the top performer. This model excels in capturing both short- and long-term patterns in the data, effectively managing the volatility inherent in oil prices. LSTM's ability to handle sequential data, along with GRU's more efficient architecture, results in more reliable and accurate predictions, making it particularly well-suited for the complex price dynamics in the oil market. As a result, the LSTM-GRU combination is a strong option for decision-makers looking to minimize forecasting errors in this sector.

In the natural gas market, 1D-CNN showed the best prediction performance after BO, contrasting with the oil market. CNNs excel at capturing short-term fluctuations and sudden spikes in price data (Livieris, Pintelas, Kiriakidou, & Stavroyiannis, 2020). The ability of CNNs to identify local patterns and short-term dependencies makes them particularly suitable for financial time series forecasting, where temporal structures can be inconsistent and subject to short-term variability (Zeng et

al., 2023).

Conversely, the worst-performing model in both oil and gas markets was the standalone Deep Feedforward Neural Networks (DFFNNs). While DFFNNs can capture general trends, they struggle with the sequential and time-dependent nature of energy prices. DFFNNs are more difficult to train because of issues like overloading of hidden units and vanishing or exploding gradients (Glorot & Bengio, 2010). This limitation makes them less effective in volatile markets like oil and natural gas, where capturing dependencies over time is crucial.

Overall, Bayesian Optimization played a crucial role in enhancing the performance of deep learning models and ensemble models by systematically optimizing the hyperparameters. BO explores the hyperparameter space using probabilistic models to minimize errors effectively. In this study, BO led to significant improvements in model performance, especially for BiLSTM, GRU, Bi-GRU, and XGBoost. However, the optimization process incurs a time cost, as seen in the increased training times across all models after BO. Despite this, the overall benefit of reducing prediction errors outweighs the time investment, making BO a valuable tool for improving forecasting models in the energy market.

In summary, this research significantly contributes to the field by demonstrating the effectiveness of using advanced models and optimization techniques for forecasting energy prices. The results show that, in the oil market, the weighted ensemble model performs best, while in the natural gas market, 1D-CNN excels. These models, optimized through BO, offer practical solutions for managing the inherent risks and uncertainties of energy markets, providing stakeholders with more reliable forecasts for informed decision-making.

5.2 Managerial implications

This research presents a direct comparison between deep learning models (such as LSTM and GRU) and ensemble models (like XGBoost and Random Forest), an area that has been less thoroughly explored in the current literature. This analysis highlights the distinct advantages of each approach. While earlier studies often focus on either deep learning or ensemble models in isolation, our findings demonstrate that ensemble methods, particularly when fine-tuned using Bayesian

Optimization (BO), can match or even exceed the performance of deep learning techniques. For example, XGBoost displayed high predictive accuracy while requiring significantly less training time compared to more complex models, making it particularly effective for time-sensitive forecasting needs.

The findings from this research have important implications for professionals in the crude oil and natural gas industries. Accurate forecasting models are essential for energy companies, policymakers, and risk management professionals. GRU, for instance, excels at capturing long-term trends in time series data, which makes it a valuable tool for pricing policy development. By improving the precision of price forecasts, GRU helps companies in the energy sector minimize costly errors and make better-informed decisions.

Conversely, XGBoost stands out in situations where fast decision-making is critical. Its short training time and consistent predictive performance make it an excellent choice for managing risks, providing reliable forecasts that help businesses respond to market volatility. Additionally, weighted ensemble models that combine multiple deep learning approaches leverage the strengths of various methods, further enhancing both prediction accuracy and reliability, making them ideal for long-term strategic planning in the energy industry.

The practical applications of these results are far-reaching. More accurate forecasting models enable managers and traders to improve pricing strategies, inventory management, and hedging tactics, ultimately resulting in more profitable and data-driven decision-making. Governments and policymakers can use these models to design more effective energy policies and better manage supply in fluctuating markets. For investors, improved predictions of price trends present opportunities for strategic investments, reducing risk while maximizing returns in the energy market.

5.3 Limitations and future work

While this research has made significant progress in forecasting oil and natural gas prices, there are a few limitations. First, the models were tested only on oil and natural gas markets, leaving other energy markets like coal, gasoline, or propane unexplored. Since different energy markets may show different price patterns, as a result, the performance of certain models could vary, and other models

may perform better in these markets. Second, due to limited computational resources, the Bayesian Hyperparameter Optimization (BO) process was simplified. Although BO improved model performance, the search for the best hyperparameters was restricted. With more powerful computational resources, future studies could explore a wider range of hyperparameter combinations for better optimization. Additionally, this study covers a data period from 2010 to 2024. While this is a robust dataset, future work could benefit from testing the models on longer time frames or different periods to capture more market conditions and trends.

While this study shows the benefits of using deep learning, ensemble methods, and BO, future research could apply these models to other energy commodities, like coal or renewable energy. This would help test if the models work well in different market conditions. Another direction is to add external factors such as macroeconomic data, geopolitical events, or climate information. Including these could make the models better at capturing a wider range of influences on energy prices. With better computational resources, future studies could also try more combinations of hyperparameters to improve the model's accuracy. Finally, making the models easier to understand would help users see what drives the predictions, making the results clearer and more useful for decision-makers.

Chapter 6

Conclusion

The energy market, especially the oil and natural gas sectors, plays a crucial role in the global economy, influencing a wide range of activities, from industrial production to household consumption. Price fluctuations in these markets have extensive consequences, affecting national economies, government policies, and the strategic decisions of businesses and investors. In such volatile markets, where supply and demand can change suddenly, accurate price forecasting becomes crucial for risk management, pricing strategies, and policy-making.

This research addresses the need for better forecasting models in the energy sector by employing a combination of deep learning techniques, ensemble methods, and BO to improve the accuracy of oil and natural gas price predictions. Through this study, we demonstrate that integrating multiple models and fine-tuning their parameters using BO significantly enhances forecasting performance, providing a valuable tool for decision-makers in the energy industry.

The key contributions of this research are threefold. First, we implemented a variety of forecasting models, including deep learning and tree-based ensemble methods, to predict both oil and natural gas prices. Second, by applying BO, we reduced forecasting errors—measured by RMSE, MAE, and MAPE—across both markets, optimizing the performance of each model. Third, we conducted a thorough comparison of model performance and training times, highlighting the strengths and trade-offs of different models in various market contexts.

Our results show clear distinctions in model performance across the two markets. In the oil market, the Weighted Ensemble model outperformed other models by effectively combining the

strengths of LSTM and GRU, both of which also showed strong individual performance. This suggests that ensemble methods, when optimized, can significantly improve predictive accuracy by leveraging the strengths of different models. In contrast, for the natural gas market, 1D-CNN was the best performer after BO optimization, showing its ability to handle the volatility and trends unique to this market. Additionally, XGBoost proved to be a strong performer in both markets, offering an optimal balance between predictive accuracy and shorter training times, making it a practical choice when both accuracy and computational efficiency are important. As a result, the model presented in this study provides valuable and practical insights for policymakers, companies, and investors, enabling them to choose models that align with their specific priorities, whether focused on accuracy or training efficiency.

Appendix A

My Appendix

A.1 The proposed model code

This appendix provides the code developed for the study. The sample code for the Brent Oil market demonstrates the key models used in the research, including 1D-CNNs, LSTMs, BiLSTMs, GRUs, BiGRUs, DFFNNs, weighted ensemble models, XGBoost, and Random Forest, alongside the application of Bayesian optimization techniques across 30 trials.

The code can be accessed via the following link:
`https://colab.research.google.com/drive/1o8yOFdfxVLokyL5B1Kh2T76UJbStmRia?usp=sharing`

References

Abdollahi, H. (2020). A novel hybrid model for forecasting crude oil price based on time series decomposition. *Applied energy*, *267*, 115035.

Alam, M. S., Murshed, M., Manigandan, P., Pachiyappan, D., & Abduvaxitovna, S. Z. (2023). Forecasting oil, coal, and natural gas prices in the pre-and post-covid scenarios: Contextual evidence from india using time series forecasting tools. *Resources Policy*, *81*, 103342.

Al-Roubaie, A. (2010). An assessment of international liquidity and higher oil prices. *World Journal of Entrepreneurship, Management and Sustainable Development*, *6*, 161-180. Retrieved from `https://api.semanticscholar.org/CorpusID:153390740`

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... Farhan, L. (2021). Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, *8*, 1–74.

Astatkie, T. (2006). Absolute and relative measures for evaluating the forecasting performance of time series models for daily streamflows. *Hydrology Research*, *37*(3), 205–215.

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

Busari, G. A., & Lim, D. H. (2021). Crude oil price prediction: A comparison between adaboost-lstm and adaboost-gru for improving forecasting performance. *Computers & Chemical Engineering*, *155*, 107513.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Retrieved from https://api.semanticscholar.org/CorpusID:4650265

dairu, X., & Zhang, S. (2021). Machine learning model for sales forecasting by using xgboost. *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 480-483. Retrieved from https://api.semanticscholar.org/CorpusID:231851078

Deng, S., Zhu, Y., Duan, S., Yu, Y., Fu, Z., Liu, J., ... Liu, Z. (2023). High-frequency forecasting of the crude oil futures price with multiple timeframe predictions fusion. *Expert Systems with Applications*, *217*, 119580.

Ding, X., Fu, L., Ding, Y., & Wang, Y. (2022). A novel hybrid method for oil price forecasting with ensemble thought. *Energy Reports*, *8*, 15365–15376.

Dixon, M., Klabjan, D., & Bang, J. H. (2015). Implementing deep neural networks for financial market prediction on the intel xeon phi. In *Proceedings of the 8th workshop on high performance computational finance* (pp. 1–6).

Dritsaki, C. (2018). The performance of hybrid arima-garch modeling and forecasting oil price. *International Journal of Energy Economics and Policy*, *8*, 14-21. Retrieved from https://api.semanticscholar.org/CorpusID:158243017

Faraji, A., Sadrossadat, S. A., Na, W., Feng, F., & Zhang, Q.-J. (2023). A new macromodeling method based on deep gated recurrent unit regularized with gaussian dropout for nonlinear circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics.* Retrieved from https://api.semanticscholar.org/CorpusID:5575601

Gono, D. N., Napitupulu, H., & Firdaniza. (2023). Silver price forecasting using extreme gradient boosting (xgboost) method. *Mathematics*. Retrieved from https://api.semanticscholar.org/CorpusID:261571768

Gupta, R., Pierdzioch, C., & Salisu, A. A. (2022). Oil-price uncertainty and the uk unemployment rate: A forecasting experiment with random forests using 150 years of data. *Resources Policy*, *77*, 102662.

Gupta, S., & Pahwa, M. S. (2011). Analysis of price volatility in energy commodities. *FIIB*

Business Review, *1*, 68 - 74. Retrieved from https://api.semanticscholar.org/CorpusID:154825238

Haque, M. I., & Shaik, A. R. (2021). Predicting crude oil prices during a pandemic: A comparison of arima and garch models. *Montenegrin Journal of Economics*, *17*(1), 197–207.

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

Huang, Y., & Deng, Y. (2021). A new crude oil price forecasting model based on variational mode decomposition. *Knowledge-Based Systems*, *213*, 106669.

Kim, J., Oh, S., Kim, H., & Choi, W. (2023). Tutorial on time series prediction using 1d-cnn and bilstm: A case example of peak electricity demand and system marginal price prediction. *Engineering Applications of Artificial Intelligence*, *126*, 106817.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, *151*, 107398.

Kotek, P., Selei, A., Tóth, B. T., & Felsmann, B. (2023). What can the eu do to address the high natural gas prices? *Energy Policy*. Retrieved from https://api.semanticscholar.org/CorpusID:254623686

Koutsoukas, A., Monaghan, K. J., Li, X., & Huan, J. (2017). Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of cheminformatics*, *9*, 1–13.

Kristjanpoller, W., & Minutolo, M. C. (2016). Forecasting volatility of oil price using an artificial neural network-garch model. *Expert Systems with Applications*, *65*, 233–241.

Lahmiri, S. (2024). Fossil energy market price prediction by using machine learning with optimal hyper-parameters: A comparative study. *Resources Policy*, *92*, 105008.

Lau, M. M., Phang, J. T. S., & Lim, K. H. (2019). Convolutional deep feedforward network for image classification. In *2019 7th international conference on smart computing & communications (icscc)* (pp. 1–4).

Lee, J. (2020). *How negative oil prices revealed the dangers of the futures market.* https://www.cnbc.com/2020/06/16/how-negative-oil-prices

-revealed-the-dangers-of-futures-trading.html. (Accessed: 2024-04-13)

Lévesque, J.-C., Gagné, C., & Sabourin, R. (2016). Bayesian hyperparameter optimization for ensemble learning. *arXiv preprint arXiv:1605.06394*.

Li, J., Zhu, S., & Wu, Q. (2019). Monthly crude oil spot price forecasting using variational mode decomposition. *Energy Economics*, *83*, 240–253.

Liaw, A. (2002). Classification and regression by randomforest. *R news*.

Livieris, I. E., Pintelas, E. G., Kiriakidou, N., & Stavroyiannis, S. (2020). An advanced deep learning model for short-term forecasting u.s. natural gas price and movement. *Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops*, *585*, 165 - 176. Retrieved from https://api.semanticscholar.org/CorpusID:218980710

Mahmoud, E., & Pegels, C. C. (1990). An approach for selecting times series forecasting models. *International Journal of Operations & Production Management*, *10*(3), 50–60.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning* (pp. 63–71). Springer.

Shen, G., Tan, Q., Zhang, H., Zeng, P., & Xu, J. (2018). Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia computer science*, *131*, 895–903.

Srinivasamurthy, R. S. (2018). *Understanding 1d convolutional neural networks using multiclass time-varying signalss* (Unpublished doctoral dissertation). Clemson University.

Talekar, B. (2020). A detailed review on decision tree and random forest. *Bioscience Biotechnology Research Communications*. Retrieved from https://api.semanticscholar.org/CorpusID:234423825

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *Neurips 2020 competition and demonstration track* (pp. 3–26).

Victoria, A. H., & Maragatham, G. (2021). Automatic tuning of hyperparameters using bayesian optimization. *Evolving Systems*, *12*(1), 217–223.

Viviani, E., Di Persio, L., & Ehrhardt, M. (2021). Energy markets forecasting. from inferential statistics to machine learning: The german case. *Energies*, *14*(2), 364.

Wang, H., Lei, Z., Zhang, X., Zhou, B., & Peng, J. (2019). A review of deep learning for renewable energy forecasting. *Energy Conversion and Management*, *198*, 111799.

Wang, J., Lei, C., & Guo, M. (2020). Daily natural gas price forecasting by a weighted hybrid data-driven model. *Journal of Petroleum Science and Engineering*, *192*, 107240.

Wang, L., Dernoncourt, F., & Bui, T. (2020). Bayesian optimization for selecting efficient machine learning models. *arXiv preprint arXiv:2008.00386*.

Wu, B., Wang, L., Lv, S.-X., & Zeng, Y.-R. (2021). Effective crude oil price forecasting using new text-based and big-data-driven model. *Measurement*, *168*, 108468.

Wu, Z., Valentini-Botinhao, C., Watts, O., & King, S. (2015). Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *2015 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 4460–4464).

Zeng, Z., Kaur, R., Siddagangappa, S., Rahimi, S., Balch, T. H., & Veloso, M. (2023). Financial time series forecasting using cnn and transformer. *ArXiv*, *abs/2304.04912*. Retrieved from `https://api.semanticscholar.org/CorpusID:258060145`

Zhang, R., Li, B., & Jiao, B. (2019). Application of xgboost algorithm in bearing fault diagnosis. In *Iop conference series: Materials science and engineering* (Vol. 490, p. 072062).