arXiv:2407.12062v2 [cs.LG] 14 Dec 2024

Multistep Brent Oil Price Forecasting with a Multi-Aspect Meta-heuristic Optimization and Ensemble Deep Learning Model

Mohammed Alruqimi[1*] and Luca Di Persio[1]

[1*]Department of Computer Science, University of Verona, Verona, Italy.

*Corresponding author(s). E-mail(s): mohammed.alruqimi@univr.it;
Contributing authors: luca.dipersio@univr.it;

Abstract

Accurate crude oil price forecasting is crucial for various economic activities, including energy trading, risk management, and investment planning. Although deep learning models have emerged as powerful tools for crude oil price forecasting, achieving accurate forecasts remains challenging. Deep learning models' performance is heavily influenced by hyperparameters tuning, and they are expected to perform differently under various circumstances. Furthermore, price volatility is also sensitive to external factors such as world events. To address these limitations, we propose a hybrid approach that integrates metaheuristic optimization with an ensemble of five widely used neural network architectures for time series forecasting. Unlike existing methods that apply metaheuristics to optimise hyperparameters within the neural network architecture, we exploit the GWO metaheuristic optimiser at four levels: feature selection, data preparation, model training, and forecast blending. The proposed approach has been evaluated for forecasting three-ahead days using real-world Brent crude oil price data, and the obtained results demonstrate that the proposed approach improves the forecasting performance measured using various benchmarks, achieving 0.000127 of MSE.

Keywords: Crude oil price forecasting, Brent oil analysis, time series forecasting, Ensemble learning, Grey Wolf Optimizer

1 Introduction

Time series forecasting plays a crucial role in various industries, from finance to logistics to energy, as it provides valuable insights into future trends and helps businesses make informed decisions. However, crude oil forecasting is particularly challenging due to the complex dynamics of the oil market and the non-linear and volatile nature of crude oil prices. In recent years, deep learning techniques have gained significant recognition for their ability to capture complex patterns in time series data. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have become popular choices for modeling temporal dependencies in forecasting tasks [1, 2]. Despite the emergence of more recent deep learning models such as transformers and generative models, LSTM and GRU remain popular choices for time series forecasting. LSTM and GRU networks excel in tasks with clear temporal sequences [3, 4]. Moreover, they are more accurate and quicker to train on small datasets, making them well-suited for daily frequency datasets such as daily oil prices [3, 5, 6]. To further enhance their performance for time series forecasting, researchers have increasingly integrated additional components, such as CNN-LSTM architectures, combining convolutional neural networks with LSTM to capture spatiotemporal patterns [7]. Nonetheless, achieving accurate predictions remains a challenge. Individual deep-learning models typically struggle to capture all data patterns, and their performance is heavily dependent on the training configuration. The effectiveness of individual models varies depending on the given dataset and conditions. Furthermore, many external factors influence crude oil prices, including geopolitical events and global economic conditions [8–11]. In other words, the capacity to enhance the capture of dataset patterns, identify external factors influencing the formation of these patterns, and select the appropriate network architecture while configuring it optimally are critical factors that significantly impact forecasting performance. The optimal choice often varies depending on the dataset's characteristics and the specific forecasting problem at hand. To address these challenges, we propose a hybrid approach to combine the Grey Wolf Optimizer (GWO) with an ensemble of five deep learning networks- LSTM, GRU, CNN-LSTM, CNN-LSTM-attention, and an encoder-decoder-LSTM. GWO is a bio-inspired metaheuristic optimisation algorithm which simulates the leadership hierarchy and grey wolf hunting mechanism in nature. The objective is to make a significant contribution to the advancement of research in the field of crude oil forecasting, building upon the recognition of the challenges mentioned above and the advantages of meta-heuristic optimisation and conditional forecast combination. This approach can be particularly beneficial in forecasting tasks where different models of diverse structures and data assumptions are expected to perform differently under various conditions. For example, one neural network can perform better than another in a specific dataset and vice-versa.

Metaheuristic optimisation techniques present a powerful solution for optimal hyperparameters tuning [12, 13]. They excels in reaching near-optimal solutions with limited information and computational resources. These algorithms generally give better results and converge faster than blind approaches like Exhaustive Grid Search. This makes them particularly valuable in optimising deep learning methods for time series forecasting. While some researchers, such as [14], have already employed meta-heuristic optimization algorithms for hyperparameter tuning in deep learning models for crude oil forecasting, we extend this application by utilizing the Grey Wolf Optimizer (GWO) algorithm to optimise the entire process including parameters outside the ML network structures such as feature selection, sliding window, and to fuse multiple forecasts.

The main contributions of this work are summarised as follows:

- An ensemble model to enhance the prediction accuracy of crude oil price.
- Evaluating five different neural network (NN) models with various architectures for multi-step crude oil forecasting.
- Identifying an optimal configuration for each model using the GWO meta-heuristic algorithm.
- Introducing a novel method to implement weighted ensemble learning through a meta heuristic optimisation algorithm.

2 Litrature Review

Initially, statistical models such as autoregressive integrated moving average (ARIMA) [15] were applied to forecast oil prices. As statistical methods relies on mathematical assumptions, they struggle to capture the nonlinear and intricate dynamic of oil prices. Recent years have witnessed strong growth in adopting deep learning for time-series forecasting, driven by its remarkable ability to capture more complex patterns within the data. Researchers have employed various deep learning networks, including long short-term memory (LSTM) networks [1, 2, 16, 17], gated recurrent units (GRUs) [17–19], convolutional neural networks (CNNs) [20, 21], and transformer models [22], to forecast crude oil prices accurately based on historical price data and additional data sources such as sentiment analysis, supply and demand data [23, 24]. A popular approach combines various neural networks layers to improve crude oil forecasting. For instance, many researchers have merged LSTM networks with CNN and self-attention mechanisms to capture temporal, local, and long-term dependencies in historical price data [18–20, 25–28]. Among these deep learning networks, GRU and LSTM stand out as the most widely applied methods for crude oil price forecasting. These methods excel in capturing temporal dependencies. Additionally, they are well-suited for smaller datasets and are computationally efficient compared to the more complex architectures of Transformers. Readers can refer to [4, 6, 7] for further details on this topic. [3] conducted an empirical analysis of various deep learning architectures, including LSTM, GRU, CNN-LSTM, and Transformer-based models, for oil price forecasting, demonstrating the superiority of GRU and LSTM. Since tuning model hyperparameters is crucial for enhancing prediction performance, metaheuristic algorithms have recently gained popularity for optimizing the training of machine learning models [12, 29]. These algorithms have been employed to some extent for optimising ML models for crude oil forecasting, i.e. [14], leading to improved prediction accuracy. In this work, we present a comprehensive approach to addressing the aforementioned limitations from multiple perspectives by developing an optimized ensemble model based on established models in the literature.

3 Preliminaries

3.1 Metaheuristic optimisation for machine learning models

At the core of machine learning models is the process of optimisation, wherein algorithms are trained to perform functions with optimal efficiency. Successful hyperparameter optimisation is essential for achieving precision in a model. The thoughtful selection of model configurations profoundly impacts accuracy and proficiency in handling specific tasks. However, the optimisation of hyperparameters can present a challenging undertaking. Metaheuristics have become powerful optimisation tools, thanks to their simplicity and capability to effectively address complex NP-hard problems with practical computational resources [30, 31]. Metaheuristic optimisation algorithms are a family of mathematical-concepts-based methods used to find good solutions to complex optimisation problems. This type of algorithm provides efficient solutions in scenarios where traditional methods may struggle due to complex and dynamic environments. They are designed to explore the search space in a way that is likely to find good solutions, even if they do not always find the optimal solution. Metaheuristics are not bound by specific problem structures, do not need gradient information and can adapt to various optimisation challenges. Their efficiency in solving real-world problems with limited resources and uncertainties makes metaheuristics one of the most notable achievements of the last two decades in operations research [29, 32, 33]. Metaheuristic algorithms can be grouped into different families based on the natural phenomena they use to guide the search process, such as, for example, evolution or ant behaviour [32]. Over the past few years, population-based methods have been successfully employed to address a diverse range of real-world challenges, including COVID-19 surveillance and forecasting, cloud-edge computing optimisation, energy-efficient cloud computing, feature selection, global optimisation, credit card fraud detection, air pollution forecasting, network intrusion detection, and optimisation of various machine learning models [12]. However, the choice of an algorithm largely depends on the optimisation problem at hand. There is no agreed guideline for large-scale nonlinear global optimisation problems for how to choose and what to choose [32].

3.2 Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) is a bio-inspired swarm intelligence optimisation algorithm inspired by grey wolves, which has been introduced in [34] in 2014. This algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature as illustrated in Figure 1. To simulate the leadership hierarchy in the GWO, four types of grey wolves—alpha (α), beta (β), delta (δ), and omega (ω)—are employed. These categories simulate the leadership hierarchy, where the optimisation process is guided by three wolves (α, β, and δ), while the omega (ω) wolves follow. As observed in the hunting process, grey wolves tend to encircle their prey, a phenomenon mathematically modelled by equations (1) and (2).

$$\vec{D} = |\vec{C}.\vec{X}_p(t) - \vec{X}(t)| \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{2}$$

Where t is the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_p$ is the vector of the prey position, and $\vec{X}$ indicates the vector of the grey wolf position. $\vec{A}$ and $\vec{C}$ vectors can be calculated as illustrated in equations (3) and (4).

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a} \tag{3}$$

$$\vec{C} = 2.\vec{r_2} \tag{4}$$

The elements of vector a undergo a linear decrease from 2 to 0 across iterations, with r_1 and r_2 being random vectors within the [0, 1] range. In emulating the hunting behaviour of grey wolves, it is assumed that α (the top candidate solution), β, and δ possess superior knowledge regarding the potential prey positions. Consequently, the three best solutions obtained thus far are preserved, compelling other search agents (including ω) to adjust their positions based on the optimal search agent positions. Further insights into the GWO algorithm can be found in [34, 35].

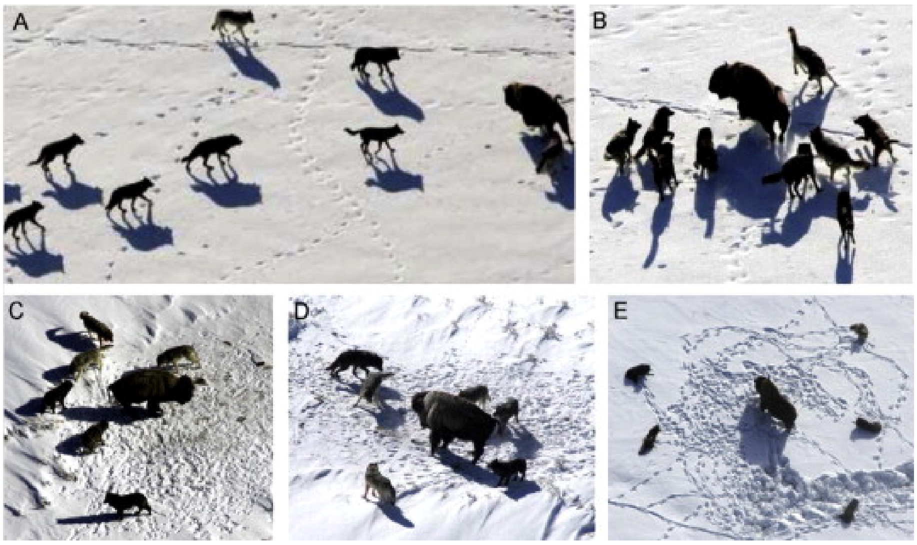


Fig. 1: Typical wolf-pack hunting behaviours: (A) chasing, approaching, and tracking prey (B–D) pursuing, harassing, and encircling (E) stationary situation and attack [36].

4 Proposed approach

In this work, we propose a deep learning ensemble forecasting approach, denoted as GWO-ensemble, to enhance multi-step Brent oil price forecasting. Since hyperparameter tuning is a crucial factor affecting model performance, we integrate the well-known Grey Wolf Optimizer (GWO). Unlike existing models that focus solely on neural network hyperparameter tuning, we apply GWO throughout all stages of model development, including feature selection, sequence data preparation, training individual models, and blending forecasts to generate the final output.

The proposed approach is implemented in two main stages, as shown in the figure 2. In the first stage, we train five deep neural networks (described in Section 4.1), proven for their proficiency in handling time series prediction. In this stage, GWO is used for optimisation of selecting sliding windows for building data sequences, feature selection, and tuning neural network hyperparameters.

The second stage involves blending the forecasts of the individual models using a weight-variant technique. In this stage, GWO is employed to optimize the weights for the forecast blending process.

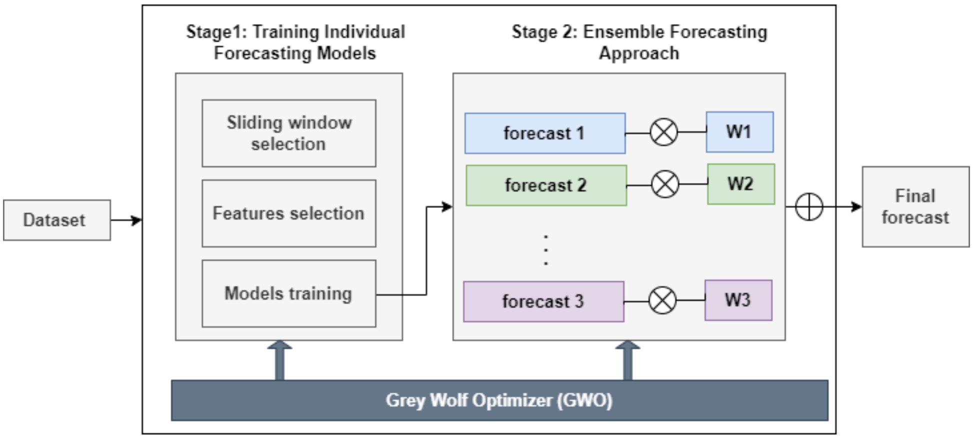


Fig. 2: Proposed Approach

To implement the proposed model, we build our code upon mealpy library [33], where users can select from a wide range of advanced population-based metaheuristic algorithms.

In Subsection 4.1, we describe the individual models used as the bases for the ensemble learning. Then, in the next subsection 4.2, we explain the ensemble learning approach. In Subsection 4.3, we outline aspects and stages in which the GWO optimiser has been employed.

4.1 Individual models

Our ensemble forecasting model is developed using blended ensemble learning that combines the following five established deep learning architectures. First, we trained each model individually, and then the individual forecasts obtained by each model were blended using a weighted-variant method, as illustrated in Figure 2.

1. **Bi-LSTM**, which is a bidirectional Long Short-Term Memory (Bi-LSTM) model; one LSTM layer followed by two full connected layers with relu activation function.
2. **Bi-GRU**, is a bidirectional Gated Recurrent Unit (Bi-GRU) model; one GRU layer followed by two fully connected layers with a relu activation function.
3. **CNN-Bi-LSTM**, This model combines two convolutional neural networks (CNNs) to extract input features and two LSTM layers for learning temporal dependencies in the data, followed by a fully connected layer with relu activation function.
4. **CNN-Bi-LSTM-Attention**, which adds an attention technique for the previous one.
5. **Encoder-decoder-Bi-LSTM** is an encoder-decoder architecture with two LSTMs with 128 units for the encoder and another LSTM with 200 units for the decoder.

4.2 GWO ensemble model

Inspired by the advantages of blending forecasts using conditional methods, we explore a weighted ensemble method combined with a meta-heuristic optimiser to fuse forecasts obtained from individual models. The weighted ensemble method in forecasting involves a mathematical approach to combine the predictions of multiple individual models to improve overall accuracy and robustness. Each model in the ensemble contributes to the final forecast with a certain weight. The final ensemble forecast is then obtained by summing the products of each model's forecast with its corresponding optimal weight. Mathematically, this can be represented as the weighted sum:

$$\text{Weighted ensemble prediction} = \sum_{i=1}^{n} M_i W_i \tag{10}$$

Here, W_i represents an optimal weight assigned to $model_i$, where $0 \leq W_i \leq 1$, and $\sum W_i = 1$. While M_i represents $model_i$ forecasts. In this study, a dynamic weighted ensemble method was employed to blend the predictions of the top five forecasts obtained from the SENT-Bi-GRU, SENT-Bi-LSTM, SENT-CNN-Bi-LSTM, SENT-CNN-Bi-LSTM-att, and SENT-USD-Encoder-Decoder-LSTM. The weights metric W was optimised using the Grey Wolf Optimization (GWO) metaheuristic algorithm.

4.3 GWO optimisation

The Grey Wolf Optimization (GWO) algorithm is employed at four different levels to optimise crucial aspects of the ensemble forecasting process: feature selection, sliding window size selection, model hyperparameter tuning, and final forecast blending. Generally, implementation of metaheuristic optimisations involves defining the main following procedures: problem preparation, objective function definition, population initialisation, and optimisation process. Firstly, the problem preparation involves specifying parameters to be calibrated and establishing their respective ranges (for instance, the range of sliding windows, the list of input features, and the number of epochs, see Table 1). The objective function representing the metric that the metaheuristic aims to either maximise or minimise during the optimisation process. This function typically encapsulates the performance of the deep learning model, such as accuracy or loss. The population in the context of metaheuristic optimisation refers to a collection of candidate solutions simultaneously. The metaheuristic explores and refines this population throughout the optimisation iterations to iteratively improve the model's performance. A larger population generally provides more diversity and allows for a more thorough search space exploration, but it also increases computational cost.

4.3.1 Feature selection

The first parameter to be calibrated for each individual model using the GWO is the selection of input features. The range of possible values for this parameter includes four options: [None], [USDX], [SENT], and [USDX, SENT]. This allows the model to be evaluated using only historical Brent oil price data, USDX alone, SENT alone, or a combination of both features.

4.3.2 Sliding window size

This parameter represents the temporal scope of historical data used to train a model. Selecting an optimal sliding window size is a delicate balance between incorporating sufficient historical information for meaningful pattern recognition and avoiding overfitting or loss of relevance. Thus, selection suitable sliding window size is crucial to ensure the trained model learns from past data effectively. The defined range for calibrating this parameter spans from 3 to 30, allowing for exploration of the optimal value within this interval.

4.3.3 Neural networks hyperparameters tuning

Hyperparameter tuning is essential for achieving optimal performance in deep learning models, as it significantly affects training efficiency and accuracy. Finding the right hyperparameters, such as learning rate, optimizer, and hidden units, is challenging due to the complex interplay of parameters and the high-dimensional search space. We applied the GWO algorithm to find the optimal configurations for all the models described in Section 4.1. Table 1 presents the optimised hyperparameters and their corresponding range values.

4.3.4 Blending of final forecasts

Ensemble forecast methods can enhance accuracy by combining forecasts of diverse models that capture different aspects of the data. This often leads to better generalisation and robustness. However, the success of the ensemble methods depends on the individual models' diversity and quality and the nature of the data. Thus, variant-Weight technique is a well-suited method for ensemble forecasting, in which multiple models are combined by assigning distinct weights to their respective predictions [25, 26]. To achieve this, we utilized the GWO algorithm to optimize the weight distribution, thereby determining the contribution of individual models to the final ensemble, as detailed in Section 4.2.

5 Experiments

This section outlines the dataset, details the experimental setup, and delves into the obtained results.

5.1 Dataset and input features

The dataset used for the experiments consists of the daily observations of the Brent crude oil price for the period from (January 3, 2012, to April 1, 2021) including the period where the COVID-19 pandemic has affected energy and stock markets. It also includes Two external factors: the daily USD closing price (USDX) and a daily cumulative sentimental score (SENT).

Crude oil prices movement is heavily impacted by external factors that shape market dynamics. In our previous work [3], we analyzed the correlation between several external variables and Brent oil prices, including stocks, commodities, and sentiment analysis indicators. Our goal was to identify the most valuable variables for improving prediction accuracy while maintaining simplicity and efficiency. Accordingly, In this work, we integrate the U.S. dollar index (USDX) and sentimental index (SENT), individually and combined. The evident inverse relationship between Brent crude, which is traded in U.S. dollars, and the U.S. Dollar Index is a valuable factor in modeling Brent prices and is widely utilised in the literature [37]. While oil sentiment analysis shows substantial relevance to major events that impact the oil market. Thus, the sentimental index can reflect many other major factors such as supply and demand dynamics.

Historical data of Brent oil and USDX rate can be obtained from many public online sources such as Yahoo Finance. While SENT index is generated by the CrudeBERT model described in [38]. CrudeBERT is a variant of FinBERT that has been fine-tuned towards assessing the impact of market events on crude oil prices, focusing on frequently occurring market events and their impact on market prices according to Adam Smith's theory of supply and demand. Mainly, CrudeBERT dataset used headlines originating from 1034 unique news sources, published on the Dow Jones newswires (approx. 21,200), Reuters (approx. 3,000), Bloomberg (approx. 1,100), and Platts (approx. 870). The sentimental score data is described and publicly available in [38]. The period from 2012 to 2021 was chosen for analysis due to the availability of sentiment scores during this time frame. Figure 3 displays the time-series plot of the daily closing price of Brent for this period.

Figure 4 illustrates the trends of Brent price, USD index (USDX), and the sentiment score (SENT) for the period from 2012 to 2021.

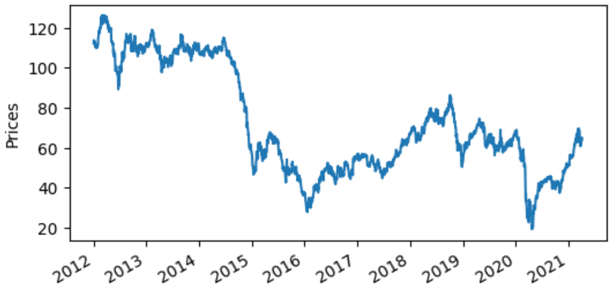


Fig. 3: Brent oil price trend from 2012 to 2021

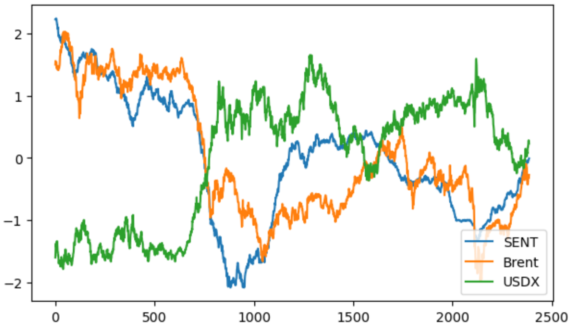


Fig. 4: Historical data of Brent oil price, US dollar index (USDX), and the sentimental score (SENT)

5.2 Experimental setup

As mentioned earlier, our experiments are conducted in two stages. In the first stage, the objective is to identify optimal values for hyperparameters of each model (optimal configuration). The individual models are described in Section 4.1, and the targeted hyperparameters are described in Table 1. Subsequently, we train and evaluate each model individually using its corresponding optimal configuration. The forecasts obtained from each model serve as inputs for the second stage. In the second stage, we aim to optimally blend the forecasts obtained individually from each model in the first stage to produce the final forecast as illustrated in Figure 2.

In order to determine the optimal configuration for each model, the Grey Wolf Optimization (GWO) has been executed through five independent runs for each network. The objective function is specifically designed to minimize Mean Squared Error (MSE) loss, aiming to attain the best global solution. The best global solution is simply a vector containing the optimal values identified for the targeted hyperparameters, corresponding to the smallest Mean Squared Error (MSE) at these values.

Table 1: The list of hyperparameters to be optimised by the GWO algorithm for each network, combined with the external features and the window size. The 'range' column shows the bound of the search space defined for the GWO.

Parameter	Range	Description
learning rate	(0.0001 - 0.1)	this parameter, crucial for controlling the step size during model training
hidden unites	(2^n, where $1 \leq n \leq 8$)	the number of hidden units
optimiser	7 options	The optimisation algorithm employed during training, the seven options are (SGD, RMSprop, Adagrad, Adadelta, AdamW, Adam, and Adamax)
dropout	(0.2 - 0.5)	Dropout is a regularisation technique that randomly drops nodes from the network during training to prevent model overfitting.
sliding window size	(3-30)	The number of timesteps used to represent the input data is crucial for capturing the temporal dynamics of the series.
external factors	4 options	to examine the impact of integrating external factors. The range includes the following options (None, USDX, SENT, and USDX + SENT)

The hyperparameters and their respective ranges (bounds or search space) are as shown in Table 1.

To reduce the problem dimension for the sake of alleviating resources consumption during the running of the GWO, we selected the most essential hyperparameters. The batch size and epoch number have also been excluded from the GWO problem dimensions. Instead, the batch size has been experimentally set to 16, and the early-stop technique is used to terminate epochs automatically.

5.3 Evaluation metrics

For each model, the GWO has been run multiple independent runs with an objective function to minimise the Mean square error (MSE), which can be calculated according to the equation (6). However, to evaluate the final ensemble and individual models, we used the following six common evaluation metrics:

- Mean Absolute Error (MAE) measures the average absolute difference between the predicted and actual values.

$$\text{MAE}(y,\hat{y}) = \frac{\sum_i^n |y_i - \hat{y}_i|}{n} \tag{5}$$

- Mean Squared Error (MSE) for calculating the average squared difference between the predicted and actual values.

$$\text{MSE}(y,\hat{y}) = \frac{\sum_i^n (y_i - \hat{y}_i)^2}{N} \tag{6}$$

- Root Mean Squared Error (RMSE), providing a measure of the average magnitude of the error.

$$\text{RMSE}(y,\hat{y}) = \sqrt{\frac{\sum_i^n (y_i - \hat{y}_i)^2}{n}} \tag{7}$$

- Mean Squared Prediction Error (MSPE) is a statistical measure used to evaluate the accuracy of a predictive model. It quantifies the average squared difference between the predicted values and the actual observed values.

$$MSPE(y,\hat{y}) = \frac{1}{n}\sum_i^n (y_i - \hat{y}_i)^2 \tag{8}$$

- Coefficient of Determination (R^2 or R-squared) is a statistical measure of how well the regression predictions approximate the actual data points. It is given by the following equation:

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \tag{9}$$

These evaluation metrics allow for quantifying the accuracy and precision of the model's predictions. Lower MSE, MAE, MSPE, and RMSE values indicate better performance, While a high R^2 value indicates better performance.

6 Results and discussion

In this section, we first present the optimal configurations identified by the GWO algorithm for each model. We then compare the accuracy of the ensemble forecasts with that of the individual model forecasts.

6.1 Models optimal configurations

Table 2 shows the best solutions identified by GWO for each model. The first column shows the MSE value achieved by the model using the corresponding configuration values, as determined for each model by the GWO optimizer, which are displayed in the subsequent columns.

The results obtained from the GWO-optimised five benchmark models, in Table 2, showed that the best-performing model is based on the Bi-GRU architecture when

Table 2: Best solutions found by GWO (models optimal configuration). TS = time steps, and h is the number of hidden units

Model	MSE	TS	LR	Dropout	h	Optimiser
SENT-Bi-GRU	0.000137	5	0.0031	0.3992	2	AdamW
SENT-Bi-LSTM	0.000194	17	0.0032	0.3205	12	Adam
SENT-CNN-Bi-LSTM	0.000280	5	0.01	0.3439	8	Adagrad
SENT-CNN-Bi-LSTM-Att	0.000300	6	0.0045	0.4399	8	Adam
SENT-USD-CNN-Bi-LSTM-Att	0.000349	10	0.0043	0.3140	4	RMSprop
SENT-encoder-decoder-Bi-LSTM	0.000358	10	0.0060	0.2573	64	Adam
SENT-USD-encoder-decoder-LSTM	0.000344	17	0.0080	0.2747	64	AdamW

incorporating the sentimental score (SENT-Bi-GRU), with the best MSE of 0.000137. The corresponding optimal configuration identified in this model is AdamW as the optimiser with a sliding window of 5 lagged steps, 0.003154571 as the learning rate, 0.0399269013 as the dropout value, and 2 hidden units. Integrating the sentiment score index (SENT) generally provides greater accuracy compared to the USDX index. However, combining both factors improves performance exclusively with the encoder-decoder-LSTM architecture. The optimal values for sliding windows are (5, 6, 10, 17). These figures align consistently with typical workdays, as 5 corresponds to the number of working days a week.

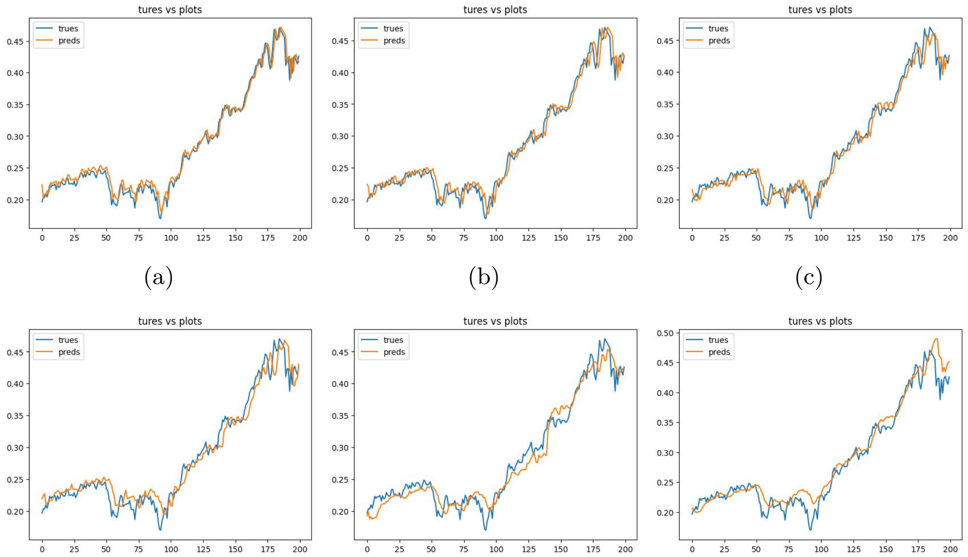
Conversely, Figure 6 illustrates global objective and run-time charts. The global objective chart shows the movement curve towards the optimal solution to minimise the MSE for a single model. Like manual hyperparameter tuning, the GWO algorithm takes longer to find a near-optimal solution with the encoder-decoder-LSTM and CNN-LSTM-att models. At the same time, finding a near-optimal solution with LSTM and GRU is faster. Experimentally, GWO reaches a near-optimal solution during the first 30 epochs.

6.2 GWO-ensemble model evaluation

Table 3: GWO-ensemble model optimal weights

Model	Optimal weight
SENT-Bi-GRU	0.0860
SENT-Bi-LSTM	0.6266
SENT-Bi-CNN-LSTM-att	0.0889
SENT-encoder-decoder-LSTM	0.1057
SENT-USDX-encoder-decoder-LSTM	0.0927

The experiments showed that this metaheuristic-calibration-based weighted ensemble method achieves better results than traditional method or other naive ensemble methods and that the ensemble model outperforms the individual models, achieving a mean square error of (0.000127) as shown in Table 4. The best final forecast has been acquired by blending the forecasts obtained from SENT-Bi-GRU, SENT-Bi-LSTM, SENT-Bi-CNN-LSTM-attention, SENT-encoder-decoder-LSTM and SENT-USDX-encoder-decoder-LSTM models. The optimal weights assigned for each model are illustrated in Table 3.



(a) (b) (c)

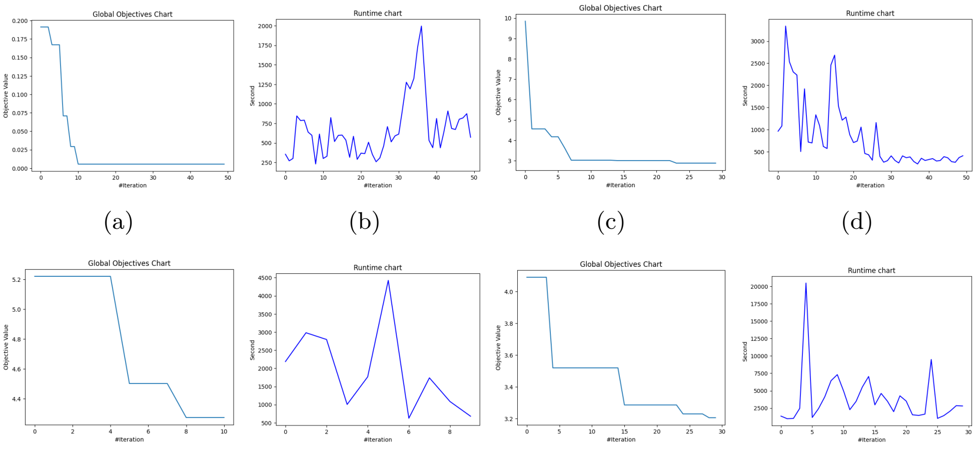
(d) (e) (f)

Fig. 5: Actual vs. predicted results of the examined models: (a). Ensemble model, (b). SENT-Bi-GRU, (c). SENT-Bi-LSTM, (d). SENT-CNN-Bi-LSTM, (e). SENT-CNN-Bi-LSTM-att, (f). SENT-USD-Encoder-Decoder-Bi-LSTM

Figure 5 illustrates the forecasting performances for all individual models. The figure shows the plot chart of Actual vs. predicted results. Table 4 shows the overall comparison of the GWO-ensemble model with the other benchmark models.

Table 4: Forecasting accuracy evaluation

Model	MAE	MSE	RMSE	MSPE	MAPE	R^2
GWO-Ensemble	0.007414	**0.000127**	0.011269	0.001649	0.026401	0.981353
SENT-BI-GRU	0.008300	0.000137	0.011704	0.002089	0.031694	0.980519
SENT-BI-LSTM	0.009800	0.000194	0.013928	0.002722	0.036012	0.972414
SENT-CNN-BI-LSTM	0.012650	0.000280	0.016733	0.001633	0.031219	0.853016
SENT-CNN-BI-LSTM-att	0.013550	0.000300	0.017306	0.004602	0.050919	0.957414
SENT-USD-CNN-BI-LSTM-att	0.014400	0.000349	0.018682	0.006660	0.057875	0.950375
SENT-Encoder-Decoder-BI-LSTM	0.012550	0.000359	0.018934	0.005175	0.046804	0.949024
SENT-USD-Encoder-Decoder-BI-LSTM	0.012900	0.000344	0.018547	0.005353	0.049209	0.951086



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6: Objective function and runtime charts of a single GWO run for various models: (a,b): LSTM, (c,d): GRU, (e,f): CNN-LSTM-att, (g,h): Encoder-Decoder.

7 Limitations

Although metaheuristic optimisation algorithms effectively tune deep learning models, they require significant computational resources. This limitation has forced us to restrict our hyperparameter search to a limited number of parameters. However, expanding the problem dimension to include a broader range of parameters could improve forecasting performance. Suppose we need to run GWO 20 independent runs for the Bi-LSTM model with 30 epochs and a population size of 10. Each Bi-LSTM model will be trained 6,000 times (20 runs * 30 epochs * 10 population size).

8 Conclusions

In this study, we demonstrated how the accuracy of Brent oil price forecasting can be enhanced through the use of ensemble models and advanced optimization techniques. We have introduced a hybrid approach that combines established deep learning networks with Grey Wolf Optimizer (GWO) to enhance the forecasting of Brent crude oil prices. The main idea is to optimise configurations at various stages of constructing a forecasting solution, alongside the power of combining forecasts with multiple deep learning models. The GWO algorithm was employed to optimise feature selection, dataset preparation, tuning hyperparameters for five individual deep learning networks, and the fusion of their forecasts. The results demonstrated that this approach improved the accuracy for each model and that the presented ensemble model outperforms benchmark models and achieves an MSE of 0.000127.

Declarations

- Funding: No external funding was received for this research.
- Conflict of interest/Competing interests: Not applicable.
- Ethics approval and consent to participate: Not applicable.
- Data availability: The dataset used in this paper can be accessed through this GitHub repos
- Code availability: Will be available upon request.
- Corresponding author: Mohammed Alruqimi

References

[1] Jovanovic, L., Jovanovic, D., Bacanin, N., Jovancai Stakic, A., Antonijevic, M., Magd, H., Thirumalaisamy, R., Zivkovic, M.: Multi-step crude oil price prediction based on lstm approach tuned by salp swarm algorithm with disputation operator. Sustainability **14**(21) (2022) https://doi.org/10.3390/su142114616

[2] Cen, Z., Wang, J.: Crude oil price prediction model with long short term memory deep learning based on prior knowledge data transfer. Energy **169** (2018) https://doi.org/10.1016/j.energy.2018.12.016

[3] Alruqimi, M., Di Persio, L.: Enhancing multi-step brent oil price forecasting with ensemble multi-scenario Bi-GRU networks. International Journal of Computational Intelligence Systems **17**(1), 225 (2024)

[4] Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are Transformers Effective for Time Series Forecasting? (2022)

[5] Ezen-Can, A.: A comparison of LSTM and BERT for small corpus. CoRR **abs/2009.05451** (2020) 2009.05451

[6] Buestán-Andrade, P.-A., Santos, M., Sierra-García, J.-E., Pazmiño-Piedra, J.-P.: Comparison of lstm, gru and transformer neural network architecture for prediction of wind turbine variables. In: García Bringas, P., Pérez García, H., Pisón, F.J., Martínez Álvarez, F., Troncoso Lora, A., Herrero, Á., Calvo Rolle, J.L., Quintián, H., Corchado, E. (eds.) 18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023), pp. 334–343. Springer, Cham (2023)

[7] Al-Selwi, S.M., Hassan, M.F., Abdulkadir, S.J., Muneer, A., Sumiea, E.H., Alqushaibi, A., Ragab, M.G.: Rnn-lstm: From applications to modeling techniques and beyond—systematic review. Journal of King Saud University - Computer and Information Sciences **36**(5), 102068 (2024) https://doi.org/10.1016/j.jksuci.2024.102068

[8] Li, J., Xu, Z., Xu, H., Tang, L., Yu, L.: Forecasting oil price trends with sentiment of online news articles. Asia-Pacific Journal of Operational Research **34**(02), 1740019 (2017) https://doi.org/10.1142/S021759591740019X

[9] Zhao, L.-T., Xing, Y.-Y., Zhao, Q.-R., Chen, X.-H.: Dynamic impacts of online investor sentiment on international crude oil prices. Resources Policy **82**, 103506 (2023) https://doi.org/10.1016/j.resourpol.2023.103506

[10] Miao, H., Ramchander, S., Wang, T., Yang, D.: Influential factors in crude oil price forecasting. Energy Economics **68**, 77–88 (2017) https://doi.org/10.1016/j.eneco.2017.09.010

[11] Fang, Y., Wang, W., Wu, P., Zhao, Y.: A sentiment-enhanced hybrid model for crude oil price forecasting. Expert Systems with Applications **215**, 119329 (2023) https://doi.org/10.1016/j.eswa.2022.119329

[12] Kaveh, M., Mesgari, M.S.: Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review. Neural Processing Letters **55**(4), 4519–4622 (2023) https://doi.org/10.1007/s11063-022-11055-6

[13] Stoean, C., Zivkovic, M., Bozovic, A., Bacanin, N., Strulak-Wójcikiewicz, R., Antonijevic, M., Stoean, R.: Metaheuristic-based hyperparameter tuning for

recurrent deep learning: Application to the prediction of solar energy generation. Axioms **12**(3) (2023) https://doi.org/10.3390/axioms12030266

[14] Zhang, T., Tang, Z., Wu, J., Du, X., Chen, K.: Multi-step-ahead crude oil price forecasting based on two-layer decomposition technique and extreme learning machine optimized by the particle swarm optimization algorithm. Energy **229**, 120797 (2021) https://doi.org/10.1016/j.energy.2021.120797

[15] Xiang, Y., Zhuang, X.H.: Application of arima model in short-term prediction of international crude oil price. In: Advances in Applied Science and IndFotios Petropoulostrial Technology. Advanced Materials Research, vol. 798, pp. 979–982. Trans Tech Publications Ltd, ??? (2013). https://doi.org/10.4028/www.scientific.net/AMR.798-799.979

[16] Hu, Zhenda: Crude oil price prediction using ceemdan and lstm-attention with news sentiment index. Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles **76**, 28 (2021) https://doi.org/10.2516/ogst/2021010

[17] Guo, L., Huang, X., Li, Y., Li, H.: Forecasting crude oil futures price using machine learning methods: Evidence from china. Energy Economics **127**, 107089 (2023) https://doi.org/10.1016/j.eneco.2023.107089

[18] Zhang, S., Luo, J., Wang, S., Liu, F.: Oil price forecasting: A hybrid gru neural network based on decomposition–reconstruction methods. Expert Systems with Applications **218**, 119617 (2023) https://doi.org/10.1016/j.eswa.2023.119617

[19] Lin, H., Sun, Q.: Crude oil prices forecasting: An approach of using ceemdan-based multi-layer gated recurrent unit networks. Energies **13**(7) (2020)

[20] Lin, Y., Chen, K., Zhang, X., Tan, B., Lu, Q.: Forecasting crude oil futures prices using bilstm-attention-cnn model with wavelet transform. Applied Soft Computing **130**, 109723 (2022) https://doi.org/10.1016/j.asoc.2022.109723

[21] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. In: International Conference on Learning Representations (2023)

[22] Liu, C., Ruan, K., Ma, X.: Dmeformer: A newly designed dynamic model ensemble transformer for crude oil futures prediction. Heliyon **9**(6), 16715 (2023) https://doi.org/10.1016/j.heliyon.2023.e16715

[23] Beckmann, J., Czudaj, R.L., Arora, V.: The relationship between oil prices and exchange rates: Revisiting theory and evidence. Energy Economics **88**, 104772 (2020) https://doi.org/10.1016/j.eneco.2020.104772

[24] Shang, J., Hamori, S.: Do crude oil prices and the sentiment index influence foreign exchange rates differently in oil-importing and oil-exporting countries?

a dynamic connectedness analysis. Resources Policy **74**, 102400 (2021) https://doi.org/10.1016/j.resourpol.2021.102400

[25] Wu, H., Levinson, D.: The ensemble approach to forecasting: A review and synthesis. Transportation Research Part C: Emerging Technologies **132**, 103357 (2021) https://doi.org/10.1016/j.trc.2021.103357

[26] Wei, X., Sun, X., Sun, J., Yin, J., Sun, J., Liu, C.: A comparative study of multi-model ensemble forecasting accuracy between equal- and variant-weight techniques. Atmosphere **13**(4) (2022) https://doi.org/10.3390/atmos13040526

[27] Qin, Q., Huang, Z., Zhou, Z., Chen, C., Liu, R.: Crude oil price forecasting with machine learning and google search data: An accuracy comparison of single-model versus multiple-model. Engineering Applications of Artificial Intelligence **123**, 106266 (2023) https://doi.org/10.1016/j.engappai.2023.106266

[28] Yang, Y., Guo, J., Sun, S., Li, Y.: Forecasting crude oil price with a new hybrid approach and multi-source data. Engineering Applications of Artificial Intelligence **101**, 104217 (2021) https://doi.org/10.1016/j.engappai.2021.104217

[29] Abdel-Basset, M., Abdel-Fatah, L., Sangaiah, A.K.: Chapter 10 - metaheuristic algorithms: A comprehensive review. In: Sangaiah, A.K., Sheng, M., Zhang, Z. (eds.) Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications. Intelligent Data-Centric Systems, pp. 185–231. Academic Press, ??? (2018). https://doi.org/10.1016/B978-0-12-813314-9.00010-4 . https://www.sciencedirect.com/science/article/pii/B9780128133149000104

[30] Oliva, D., Houssein, E.H., Hinojosa, S.: Metaheuristics in Machine Learning: Theory and Applications. Studies in Computational Intelligence. Springer, ??? (2021). https://books.google.it/books?id=Zlw4EAAAQBAJ

[31] Calvet, L., Armas, J., Masip, D., Juan, A.A.: Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. Open Mathematics **15**(1), 261–280 (2017) https://doi.org/10.1515/math-2017-0029

[32] Van Thieu, N., Barma, S.D., Van Lam, T., Kisi, O., Mahesha, A.: Groundwater level modeling using augmented artificial ecosystem optimization. Journal of Hydrology **617**, 129034 (2023) https://doi.org/10.1016/j.jhydrol.2022.129034

[33] Van Thieu, N., Mirjalili, S.: Mealpy: An open-source library for latest metaheuristic algorithms in python. Journal of Systems Architecture (2023) https://doi.org/10.1016/j.sysarc.2023.102871

[34] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Advances in Engineering Software **69**, 46–61 (2014) https://doi.org/10.1016/j.advengsoft.2013.12.007

[35] Elhariri, E., El-Bendary, N., Hassanien, A.E.: Bio-inspired optimization for feature set dimensionality reduction. In: 2016 3rd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA), pp. 184–189 (2016). https://doi.org/10.1109/ACTEA.2016.7560136

[36] Muro, C., Escobedo, R., Spector, L., Coppinger, R.P.: Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations. Behavioural Processes **88**(3), 192–197 (2011) https://doi.org/10.1016/j.beproc.2011.09.006

[37] Zhou, J., Sun, M., Han, D., Gao, C.: Analysis of oil price fluctuation under the influence of crude oil stocks and us dollar index — based on time series network model. Physica A: Statistical Mechanics and its Applications **582**, 126218 (2021) https://doi.org/10.1016/j.physa.2021.126218

[38] Kaplan, H., Weichselbraun, A., Braşoveanu, A.M.P.: Integrating economic theory, domain knowledge, and social knowledge into hybrid sentiment models for predicting crude oil markets. Cognitive Computation (2023) https://doi.org/10.1007/s12559-023-10129-4