Crude Oil Price Forecasting Using LSTM and GRU Feature Extractor and Machine Learning Regressor

Rifdah Amelia * and Lili Ayu Wulandhari

Department of Computer Science, BINUS Graduate Program-Master of Computer Science,
Bina Nusantara University, Jakarta, Indonesia
Email: rifdah.amelia@binus.ac.id (R.A.); lili.wulandhari@binus.ac.id (L.A.W.)
*Corresponding author

Abstract—**The crude oil market is distinguished by significant volatility, primarily due to its responsiveness to economic and geopolitical events and the intricate factors that drive price fluctuations. Accurate forecasting of crude oil prices is essential for mitigating adverse impacts on national economic stability and growth. This study examines the prediction of Brent crude oil prices utilizing a 20-year time series dataset encompassing 2004 to 2024. A hybrid modelling framework integrates Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models as feature extractors, followed by regression analyses utilizing SVR (Support Vector Regression), Random Forest, and Extreme Gradient Boosting algorithms. The proposed architecture results in the development of six distinct models, evaluated across two different window sizes, specifically 1 and 5, to assess the impact of temporal granularity on predictive accuracy, which may help identify optimal configurations for enhancing model performance in various forecasting scenarios. Model performance is quantified using Mean Absolute Error (MAE) and R^2 Score (Coefficient of Determination) metrics. The experimental findings indicate that the LSTM-SVR model operating with a window size of 2 exhibits superior performance, achieving an MAE of 1.050 and an R^2 Score of 0.997 on the training dataset and an MAE of 1.558 with an R^2 Score of 0.973 on the test dataset. The optimal configuration for the LSTM feature extractor comprises 10 units, with a dropout rate of 0.05, 10 epochs, and a batch size 32. The SVR regressor utilizes an RBF (Radial Basis Function) kernel with parameters C = 100, epsilon = 0.01, and auto gamma.**

Keywords—**brent crude oil prices, time series, Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Support Vector Regression (SVR), Random Forest (RF), Extreme Gradient Boosting (XGB)**

I. Introduction

Crude oil, also known as petroleum, is a blend of different hydrocarbons that can be refined to create various fuels [1]. These processes may produce natural gas, methane, liquefied petroleum gas, petroleum ether, kerosene, heating oil, gas oil, diesel, etc. [1]. Crude oil as a raw material for manufacturing derivative fuels is often subject to price changes at any time. An increase in the price of crude oil can inhibit a country's economic growth, which results in inflationary pressure. Otherwise, oil-exporting countries may be negatively affected if oil prices fall drastically [2]. Fluctuations and shocks that occur in crude oil prices have a significant impact on the real economy and virtual economy [3–5] and have an impact on import and export trade activities [6]. In addition to causing changes in inflation, changes in crude oil prices can affect the exchange rate through the transmission of supply and demand mechanisms [7].

Changes in crude oil prices are influenced by several factors, such as economic and political developments and other hidden factors, making crude oil price forecasting more challenging to study. Some other factors are the Organization of the Petroleum Exporting Countries (OPEC) market, supply and demand factors, announcements or information published by OPEC, and the U.S. Strategic Petroleum Reserve for the futures market [8, 9]. Various factors that affect the price of crude oil cause the process of forecasting crude oil prices to be more complex, and there is a need for a unique and appropriate method to predict with such data characteristics. Other challenges in crude oil price prediction case studies are high noise, non-linearity, and nonstationarity, which determine the complexity and difficulty of crude oil futures forecasting [10].

Predicting crude oil prices is important because changes in these prices can impact a country's economy, and traders widely buy and sell this commodity in the stock market. From an industrial point of view, changes in crude oil prices can directly affect the price of the products produced. Manufacturing requires fuel, other industrial raw materials, and crude oil derivatives [6]. Moreover, academics, investors, and the government widely use predicting crude oil prices to help control the risks that may arise later [7]. Hasan *et al.* [2] highlights the urgent need for accurate predictions of crude oil prices, which is crucial for researchers, businesses, industries, and governments. Additionally, Akil *et al.* [11] and Foroutan

Manuscript received November 13, 2024; revised January 10, 2025; accepted January 13, 2025; published August 8, 2025.

and Lahmiri [12] have proposed the use of various machine learning models, including deep learning, to enhance the prediction of crude oil prices. With the various reasons for the need to predict the price of crude oil, this research was conducted as one of the renewals of science, complementing the shortcomings in previous studies and trying to find a more accurate model from previous studies.

Previous research used machine learning algorithms and artificial neural networks to predict crude oil prices. Some of the algorithms that have been used in previous studies are random forest [2], support vector machine or support vector regression [2, 7], artificial neural network [13], long short-term memory [6, 7], gated recurrent unit [7], convolutional neural network [7], extreme gradient boosting [14], and several other algorithms that will be described in Section II.

Prior research has employed Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms to forecast crude oil prices. These sophisticated machine learning techniques are especially effective in analyzing time series data because they capture complex temporal dependencies. This research indicates that both LSTM and GRU models are adept at identifying underlying patterns and trends within historical price data, thereby facilitating more accurate forecasting. The models accurately predict crude oil prices with low error rates. LSTM and GRU algorithms can overcome vanishing gradients. Furthermore, the model implements the gate concept, allowing it to retain important information and eliminate less significant data, ultimately enhancing the quality of the features. So, we decided to use the LSTM and GRU models in this research.

Guo *et al.* [7] conducted a comprehensive review that strongly supports the selection of this model by providing insightful findings on the predictive performance of various models for crude oil prices. The review highlights that Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models demonstrate a remarkable ability to forecast crude oil prices with significantly lower error rates than existing models. Researchers observe this enhanced accuracy regardless of whether the models incorporate dynamic variable factors known to influence crude oil prices or operate independently of these variables.

The implications of this finding suggest that LSTM and GRU models offer a more reliable approach for stakeholders interested in forecasting crude oil price movements in an increasingly volatile market. This research proves this statement by examining the performance of models with and without influencing factors, showing that the resulting error value has a relatively small difference. The results of his research prove this statement by examining the performance of models with and without influencing factors, showing that the resulting error value has a relatively small difference. The results of this study prove the results of the review conducted. In addition, research conducted by Busari and Lim [15] shows that the hybrid model can provide a smaller error value than the single model. So, after evaluating several aspects, we decided to use the LSTM and GRU models as feature extractors for this study. The reliability of the LSTM and GRU algorithms is proven in other cases, such as COVID-19 prediction [16], stock price prediction [17], emerging stock market prediction [18], livestock product price forecasting [19], and remaining useful prediction for lithium-ion batteries [20].

We selected the regressor model for this study based on a review of prior research in the literature. This study showed that the Extreme Gradient Boosting (XGB), Random Forest (RF), and Support Vector Regression (SVR) models worked well in predicting crude oil prices in previous studies. In addition, these regressor algorithms have never used this method in previous studies.

The main objectives of this research are:

- Performing feature extraction on crude oil price time series data using the LSTM and GRU algorithm approaches.
- Perform regression modeling with machine learning to predict crude oil prices with features extracted from LSTM and GRU.
- Comparing the performance evaluation of models with LSTM-machine learning and GRU-machine learning and finding the best model to predict crude oil prices.

This paper consists of five sections: Section I discusses an introduction to crude oil, the purpose of the research, and why this research is essential; Section II will review the literature study done in previous research; Section III will discuss the algorithm's basic theory and the research method; Section IV will discuss the results of this research in detail; and Section V will discuss the conclusion of the research that has been done.

II. LITERATUR REVIEW

Before conducting this research, a literature review was conducted on previous research relevant to the topic of this research, which is crude oil price forecasting. The literature review was conducted to determine what problems have been discovered in previous studies and was a consideration in determining the method to be used in this study.

A study carried out by Busari and Lim [13] compared the AdaBoost-LSTM, AdaBoost-GRU, single LSTM, and single GRU models to predict crude oil prices. The dataset used in this study is the daily price of crude oil from October 2009 to June 2021. Daily data is used in this study because it is possible to predict oil prices daily using daily price data. The results of this study show that the AdaBoost-GRU model has the best performance compared to other models, with an MAE value of 1.4164.

Zhang and Hong [6] and Maulana *et al.* [21] tested the LSTM algorithm to predict crude oil prices. In Ref. [6], the LSTM algorithm outperformed the ARIMA (Autoregressive Integrated Moving Average) and ANN (Artificial Neural Network) models for testing on short-term, medium-term, and long-term data. Likewise, the research results of Maulana *et al.* [21] show that the LSTM model created can predict crude oil prices, as evidenced by the small error value produced.

Gulati *et al.* [13] predicted crude oil prices using a hybrid artificial neural network model and particle swarm optimization. This algorithm was chosen because the ANN algorithm can deal with large amounts of dynamic, non-linear, and noisy data. Meanwhile, the PSO (Particle Swarm Optimization) model was chosen to find the optimal weight and bias for the ANN model. The performance of the proposed model in this study is compared with the performance of a single ANN model. Both models are evaluated using the MAE value, where it is concluded that the ANN-PSO model performs better than a single ANN with an RMSE (Root Mean Square Error) value of 1.76.

Other algorithms used in Ding *et al.* [22] are random forest, XGBoost, and LightGBM. This research proposes an ensemble model using the three types of tree-based algorithms. The results of this study show that the proposed hybrid model has the best performance compared to the XGB model. Furthermore, Yang *et al.* [14] also applied the XGB algorithm to predict crude oil prices. In this study, the performance of the models was compared with Autoregressive Integrated Moving Average (ARIMA), Autoregressive Integrated Moving Average Exogenous (ARIMAX), Random Forest, and XGB models. The results of this study show that the XGB algorithm can predict crude oil prices well compared to other models for data with lags 0, 7, and 10.

Research conducted by He *et al.* [23] proposed the Time Delay Embedding-Convolutional Neural Network (TDE-CNN) method to predict crude oil prices. This study uses TDE to transform data into a two-dimensional phase space. On the other hand, the CNN model is used to extract the most substantial features of the data by utilizing this algorithm’s hierarchical feature extraction capabilities. Researchers compared the proposed model with random walk and ARMA models, where the proposed model performs better than the other two models.

Jahandoost *et al.* [24] applied the LSTM-AM (Long Short-Term Memory-Attentional Mechanism), GRU-AM (Gated Recurrent Unit-Attentional Mechanism), CNN-LSTM-AM (Convolutional Neural Network-Long Short-Term Memory-Attentional Mechanism), and CNN-GRU-AM (Convolutional Neural Network-Gated Recurrent Unit-Attentional Mechanism) methods. In this study, 39 other features were used to predict crude oil prices with the proposed hybrid model. The results show that the CNN-GRU-AM model is better than the other three. Aldbagh *et al.* [18] researched crude oil price prediction using several algorithms, including CNN-LSTM, LSTM, CNN, SVM (Support Vector Machine), and ARIMA. The research scheme is to test for one-step and multi-step models. The study shows that the CNN-LSTM model predicts oil prices better than other models for one-step and multi-step testing.

Guo *et al.* [7] using RNN (Recurrent Neural Network), LSTM, GRU, SVR, MLP (Multi-Layer Perceptron), CNN, and BP (Backpropagation) in this study. The dataset used in this study is the price of Chinese crude oil from March 2018 to February 2023. In addition to this data, the South China commodity index data, related crude oil futures consist of West Texas Intermediate (WTI) crude oil, Brent crude oil, Shanghai Composite Index, and China Securities Index (CSI) Energy Index. The results of this study show that the GRU model has a minimum prediction error and an estimated value close to the actual value.

Sen and Choudhury [25] conducted research related to crude oil price prediction using a deep learning approach. The algorithms used in this research are Long Short-Term Memory and Gated Recurrent Unit, where the hyperparameters of the two algorithms are optimized using Particle Swarm Optimization. This research uses quantitative methods to study time series data and build mathematical models. The results of this study show that the proposed model, namely GRU using Particle Swarm Optimization, has the most minor error value compared to several models from other researchers. The model in this study was evaluated using RMSE, where the best model could provide an error value of 1.23.

Hasan *et al.* [2] conducted research to predict crude oil prices using ensemble learning models. The algorithms used in this research are lasso regression, bagging lasso regression, boosting, random forest, and support vector regression. The researcher proposed a model named LKDSR, which involves linear regression, KNN (K-Nearest Neighbors), SVR, decision tree regression, and ridge regression algorithms. The results of this study indicate that the proposed model has good performance. Research conducted by Akil *et al.* [11] compared SVM, SGD (Stochastic Gradient Descent)-based SVM, and SMO (Sequential Minimal Optimization)-based SVM algorithms to predict daily, weekly, and monthly crude oil prices. The results of this study show that the SMO-based SVM model performs best compared to other models.

According to the literature review findings, researchers have demonstrated that the LSTM and GRU algorithms effectively forecast crude oil prices with a relatively low error margin close to the actual prices [6, 7, 15, 21, 25]. The LSTM and GRU algorithms benefit models by mitigating the vanishing gradient problem, allowing them to retain information over extended periods [26]. This capability proves especially useful for analyzing historical trends in crude oil prices. Because the algorithm can store long-term valuable information for analyzing trends in the dataset, and previous research demonstrates the efficacy of LSTM and GRU algorithms in predicting these prices, we have chosen to use these algorithms as feature extractors in this study.

Additionally, based on the literature, we identified Extreme Gradient Boosting (XGB), Random Forest (RF), and Support Vector Regression (SVR) as regressors, as past models have shown that they can successfully predict crude oil prices. We will evaluate the models developed in this study to determine which one performs the best. In addition, the model proposed in this study has never been applied in previous studies. To facilitate a compelling comparison between the proposed model and the most successful model identified in previous research, we have provided comprehensive details in Table I. This table presents the types of model utilized and the performance metrics associated with the top-performing models.

TABLE I. PERFORMANCE COMPARISON FROM PREVIOUS RESEARCH

Reference	Dataset and Time Period	Proposed Method	Result
[15]	Crude oil price from 2009 to 2021.	• AdaBoost-LSTM • AdaBoost-GRU	The best model is AdaBoost-GRU with MAE 1.4164 and RMSE 2.4602.
[6]	Brent and WTI Crude oil price from 1986 to 2021	• ANN • ARIMA • LSTM	The best model for long-term prediction is LSTM. The MSE score for the WTI price forecasting model is 0.122 and 0.420 for MAE, while the MSE score for the Brent price forecasting model is 0.086 and 0.284 for MAE.
[21]	Brent crude oil price from 1987 to 2021	LSTM	The best model uses 50 LSTM units, 2 lookbacks, a batch size of 104, and an epoch of 432 times. The RMSE test value is 1.27055, and the MAE value is 0.92827.
[13]	Crude oil price from 2019 to 2022	• ANN • ANN-PSO	The best model is ANN-PSO, with a value of 1.79 for the RMSE and 0.017 for the MAPE.
[22]	Crude oil price from 2018 to 2021	• ARIMA-BP • Random forest • XGBoost • LightGBM • SVR • RF-XGB-LGBM	The best model is RF-XGB-LGBM, with 13.7417 MAE on the test set.
[14]	China's Shanghai crude oil from 2018 to 2023	• ARIMA • ARIMAX • Random Forest • XGBoost	The best model had a 10-day lag. This model gives 16.6583.
[23]	Crude oil price	• TDE-CNN • ARMA • RW	The best model was TDE-CNN, with an MAE value of 3.8928×10^{-4}. We found that the model's performance is sensitive to the chosen hyperparameter.
[24]	WTI crude oil price from 2008 to 2023	• LSTM-AM • GRU-AM • CNN-LSTM-AM • CNN-GRU-AM	The best model is CNN-GRU-AM, with a test MAE score of 1.5534.
[27]	WTI crude oil price from 2013 to 2022	• CNN-LSTM • CNN • SVM • ARIMA	The best model is CNN-LSTM, with an RMSE score of 2.18.
[7]	China crude oil price from 2018 to 2023	• RNN • LSTM • GRU • SVR • MLP • CNN • BP	The best model for forecasting crude oil prices with 5- and 10-day steps is GRU. The MAE for the 5-day step is 2.3599, and for the 10-day step, it is 2.3961.
[25]	Crude oil price from 1983 to 2021	• LSTM • GRU • PSO-LSTM • PSO-GRU	The best model from this research was PSO-GRU, which forecasted the crude oil price with a 4-day lag, MAE 0.91, 1.23 for the RMSE score, and R-squared 0.9939.
[2]	Brent and WTI crude oil price from 1987 to 2022	LKDSR	The model with the best performance for predicting the daily price is the one that gives 0.9954 for MAE, 0.00010 for MSE, and 0.99 for R-squared.
[11]	WTI crude oil price from 1986 to 2023, and Brent crude oil price from 1987 to 2023	• SVM • SMO based SVM • SGD based SVM	The model performs better using Brent crude oil price, and the best model was for predicting the daily dataset. The best model was SGD-based SVM with an MSE value of 4.546300 and R-squared of 0.98921.

III. MATERIALS AND METHOD

A. Algorithm Basic Theory

1) Long Short-Term Memory (LSTM)

The LSTM algorithm is a development of the RNN algorithm that aims to overcome the long-term dependency experienced by the RNN algorithm [28]. Fig. 1 shows the chained architecture of the LSTM algorithm. This algorithm consists of three types of gates: forget gate, input gate, and output gate. The other components are hidden state, cell state candidate, and cell state.

The core of the LSTM algorithm is the cell state, a component that acts as a global memory or an aggregate of the LSTM network across time steps. The first step in the LSTM algorithm is to determine what information will be discarded from the cell state [28]. This determination is made based on calculations derived from Eq. (1). If the value is 0, then the information will not be used entirely; otherwise, if it is 1, it will be used entirely.

$$f_t = \sigma\left(W_f \, . [h_{t-1}, x_t] + b_f\right) \tag{1}$$

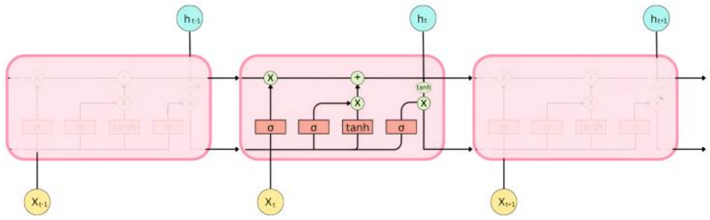

Fig. 1. Long short-term memory architecture [28].

Next, it will determine the information stored in the cell state. This stage consists of two processes. The first process is at the input gate (i_t), where the value will be updated, as described in Eq. (2). Meanwhile, the second process is determining the candidate ($\tilde{C}_t$) that will be stored in the cell state, as described in Eq. (3).

$$i_t = \sigma \left(W_i \,.\, [h_{t-1}\,, x_t] + b_i\right) \tag{2}$$

$$\tilde{C}_t = \tanh\left(\sigma \left(W_c \,.\, [h_{t-1}\,, x_t] + b_c\right)\right) \tag{3}$$

After calculating the forget gate, input gate, and cell state candidate values, the next step is to update the cell state (C_t) using Eq. (4). The previous information is added to the new information.

$$C_t = f_t \odot C_{t-1} + \tilde{C}_t \odot i_t \tag{4}$$

The last stage in this algorithm is to determine the output gate value (o_t) and calculate the hidden state (h_t). Eq. (5) shows the operation on the output gate, and Eq. (6) shows the operation of determining the hidden state value.

$$o_t = \sigma \left(W_o \times [h_{t-1}\,, x_t] + b_o\right) \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

2) Gated Recurrent Unit (GRU)

The GRU algorithm is proposed to improve the performance of the RNN algorithm, which experiences vanishing gradient problems [26]. In this algorithm, the forget gate and input gate are converted into one gate, the update gate [28]. The GRU algorithm replaces the cell state with a candidate activation vector that will be updated using the reset gate and update gate. The architecture of the GRU algorithm is shown in Fig. 2.

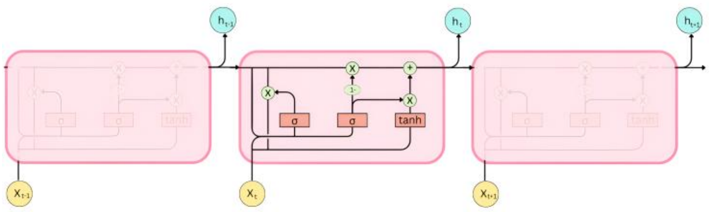

Fig. 2. Gated recurrent unit architecture [28].

The update gate controls how much information from the previous hidden state is brought to the current hidden state [29, 30]. This algorithm uses several equations, namely Eq. (7) to calculate the update gate, Eq. (8) to calculate the reset gate, Eq. (9) to determine the candidate hidden state, and Eq. (10) to calculate the hidden state.

$$z_t = \sigma \left(W_z x_t + U_z h_{(t-1)}\right) \tag{7}$$

$$r_t = \sigma \left(W_r x_t + U_r h_{(t-1)}\right) \tag{8}$$

$$\tilde{h}_t = tanh \left(W x_t + r_t \odot U h_{(t-1)}\right) \tag{9}$$

$$h_t = (1 - z_t) h_{(t-1)} + z_t \tilde{h}_t \tag{10}$$

3) Extreme Gradient Boosting (XGB)

Extreme gradient boosting is an ensemble learning method consisting of n decision trees. The new tree is updated iteratively using the gradient algorithm and residuals from the previous tree to provide the most accurate results and reduce residuals [14].

4) Random forest

Random forest is an ensemble machine learning method that uses a decision tree framework [14]. The random forest comprises a bagging algorithm and CART trees [14]. Theoretically, each tree is independent, so it does not depend on other trees, and each tree can be trained simultaneously without taking much time [22]. The trees in the model will have different internal structures and information. The forecasting results will be obtained based on the voting results so that the forecasting results can reach the optimal number [22].

5) Support vector regression

The support vector regression algorithm is a development algorithm of the support vector machine, originally used to classify binary objects. This algorithm was developed to predict numerical values [31]. The goal of this algorithm is to find the best similarity that can accurately predict the target variable while reducing the level of complexity to avoid overfitting results. The dataset will be aggregated into a hyperplane to reduce the potential for overfitting and improve the machine's generalization ability [11].

B. Research Method

The research flow is described in Fig. 3. This research begins by conducting a literature study to identify gaps in previous research and understand the research to be carried out. Furthermore, problem formulation will be carried out to determine the objectives and make the scope of research. Furthermore, the dataset collection process is carried out, namely collecting Brent crude oil price data from January 2004 to April 2024. Further explanation of the dataset is described in subsection 1).

After the dataset is collected, a preprocessing stage is carried out to ensure the data quality used. The data ready for use is then used to build a crude oil price forecasting model. The model-building process is divided into LSTM-machine learning and GRU-machine learning; each flow will be described more clearly in subsections 2) to 3). In this study, the LSTM and GRU algorithms served as feature extractors. Machine learning algorithms used as regressors in this research are Extreme Gradient Boosting (XGB), Random Forest (RF), and Support Vector Regression (SVR). The model will be assessed using Mean Absolute Error (MAE) and R-squared (R^2) Score. The last stage of this research is to compile the research results into a paper to explain the results of the research that has been done.

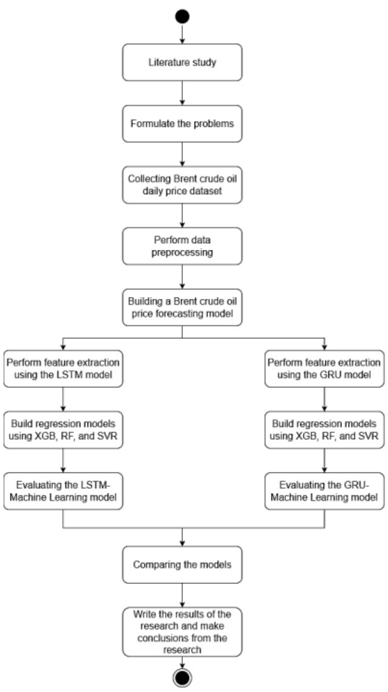

Fig. 3. Research flow for predicting the price of crude oil using LSTM-machine learning and GRU-machine learning.

1) Dataset and descriptive statistic

In this study, the Brent daily crude oil price dataset for the period 1 January 2004 to 30 April 2024 is used. This dataset consists of two variables, namely date variables and price variables. In this research, the dataset will be divided into training data, validation data, and testing data with a proportion of 80:10:10. Table II shows the proportion of division in the dataset in this study. In this study, window sizes with sizes 1 and 5 will be used; this refers to research conducted by Maulana *et al.* [21], where it is stated that the smaller the number of window sizes or window sizes used, the better the resulting model will be.

TABLE II. DATASET SHARING PROPORTION

Dataset Split	Train set	Val set	Test set
Amount and	4243	530	530
Percentage	80%	10%	10%
Period	02-01-2024 to 06-04-2020	07-04-2002 to 18-04-2022	19-04-2022 to 30-04-2024

The purposive sampling method is applied to determine the dataset period used in this study. In 2003, the invasion of Iraq led to the depletion of crude oil reserves for production. This event resulted in reduced oil reserve capacity, which triggered increased crude oil prices. Therefore, the initial data used in this study is the oil price data in 2004, before the significant increase or change in the price of crude oil. The price change is presented in Fig. 4, utilizing the price of crude oil as of April 2024 as the data endpoint for the study.

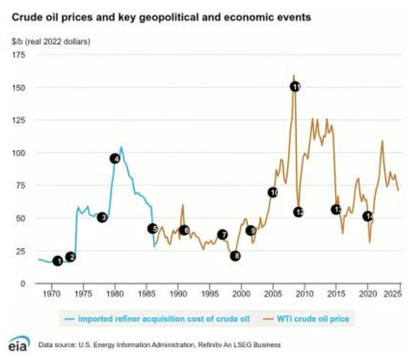

Fig. 4. Geopolitical and economic events that triggered crude oil price changes from 1968 to 2024 [32].

Between 2004 and 2024, the price of Brent crude oil reached its lowest point on April 21, 2020, at 9.1200 USD per barrel and attained its highest value on July 3, 2008, at 143.9500 USD per barrel. The standard deviation of this dataset is recorded at 25.3477 USD per barrel, while the average price of crude oil during this period is 74.0264 USD per barrel. Table III provides a detailed overview of the descriptive statistics for this dataset.

TABLE III. STATISTICS ON BRENT CRUDE OIL PRICE 2004–2024

Statistic Descriptions	Values
Count	5153
Mean	74.0264
Standard Deviation	25.3477
Minimum Value	9.1200
25%	54.5300
50%	70.7200
75%	93.5200
Maximum Value	143.9500

2) Preprocessing

The preprocessing stage conducted in this study effectively removes extraneous noise from the dataset. The noise identified consists of dot punctuation, which is present in 150 rows of data. We employ a systematic row elimination approach to mitigate this issue and effectively remove the identified noise.

TABLE IV. ADF TEST ON BRENT CRUDE OIL PRICE FROM 2004–2024

Metrics	Values
ADF Statistic score	−2.4770
p-value	0.1211
Critical Values 1%	−3.4316
Critical Values 5%	−2.8621
Critical Values 10%	−2.5671

Using the Augmented Dickey-Fuller (ADF) test, we conducted stationarity testing on the dataset. The results yielded an ADF statistic of −2.4770 and a *p*-value of 0.1211. Additionally, we present the critical values at the 1%, 5%, and 10% significance levels in Table IV. The discoveries indicate that the ADF statistic exceeds the critical value, leading us to accept the null hypothesis. Therefore, we conclude that the dataset exhibits non-stationary behaviour, as the *p*-value supports this conclusion by remaining above the significance threshold of 0.05.

To assess stationarity, we evaluate the Augmented Dickey-Fuller (ADF) statistic and examine the autocorrelation using the Autocorrelation Function (ACF) plot in Fig. 5. Additionally, we analyze the Partial Autocorrelation Function (PACF) plot shown in Fig. 6. These analyses aim to identify potential trends or seasonal patterns within the data. The ACF plot in Fig. 5 reveals a statistically significant positive correlation at the initial lag, where all data points exceed the established confidence intervals. Furthermore, the gradual decline in ACF values suggests a diminishing correlation over time, indicating a prevailing trend within the dataset. This slow attenuation also implies that historical crude oil prices influence the time series over an extended period. Additionally, the absence of periodic spikes at specific intervals in Fig. 5 supports the conclusion that the crude oil price dataset from 2004 to 2024 lacks any seasonal patterns. The insights derived from the PACF graph in Fig. 6 further affirm that this dataset exhibits a non-stationary trend while simultaneously confirming the absence of seasonal characteristics.

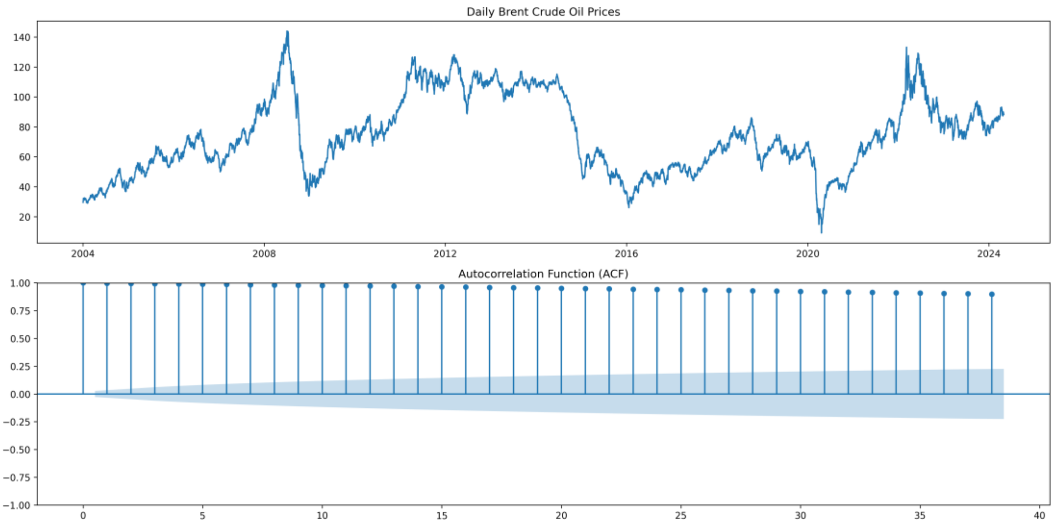

Fig. 5. Autocorrelation test for the price of brent crude oil from 2004–2024.

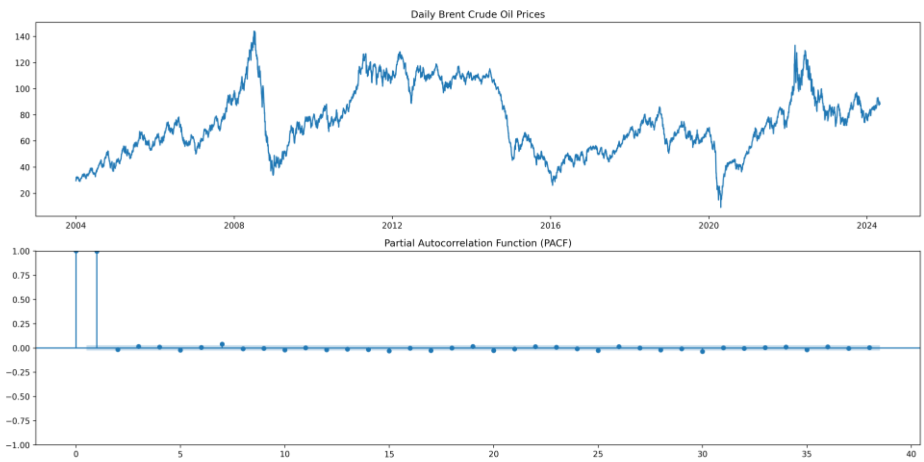

Fig. 6. Partial autocorrelation test for the price of Brent crude oil from 2004–2024.

To assess the presence of noise in the data, a box plot visualization, as illustrated in Fig. 7, was utilized. The results of this visualization indicate that there is no noise in the data. Therefore, it can be concluded that the loss metrics to be considered in this study are the Mean Absolute Error (MAE) and the R-squared (R^2) Score. Additionally, the normalization process is performed using the Min-Max Scaler algorithm, as shown in Eq. (11).

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{11}$$

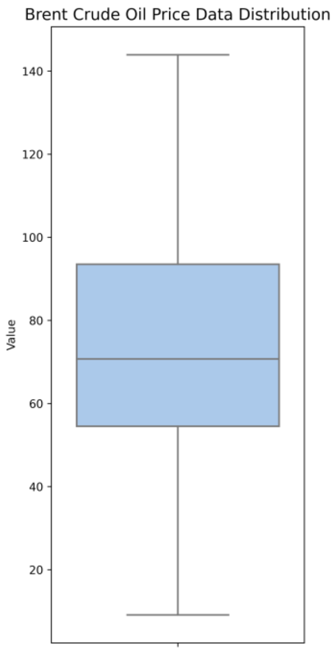

Fig. 7. Visualization of the distribution of Brent crude oil price data from 2004–2024.

3) LSTM-machine learning

Data that has undergone preprocessing will be used to build the LSTM machine learning model. Fig. 8 shows the flow of the model experiment. The testing phase begins with training the feature extraction model and ends with the model evaluation process using test data.

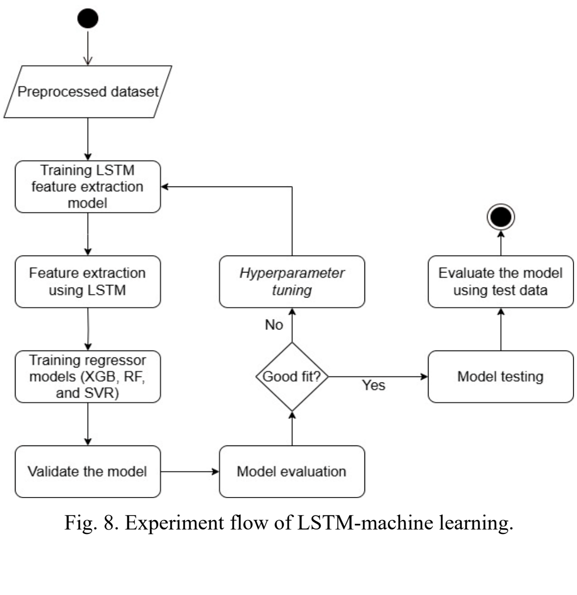

Fig. 8. Experiment flow of LSTM-machine learning.

This research will test the LSTM feature extraction model using the hyperparameters listed in Table V. These hyperparameters will be tuned until the most appropriate feature extraction model is found. In this study, we actively tune the number of LSTM units to capture the temporal dependencies in the input data effectively. We also adjust the dropout rates to prevent excessive co-adaptation within the model. A correlation exists between the number of LSTM units and dropout; as the neuron count increases, the model's capacity expands, raising the risk of overfitting. Therefore, increasing the dropout rate is a strategy to mitigate this risk.

Additionally, we focus on fine-tuning the epoch hyperparameter to determine the optimal number of complete passes through the training data during the training process. We also adjust the batch size to define the number of training samples processed before updating the model's weights. Utilizing a smaller batch size can lead to noisier updates. These considerations drive the selection of hyperparameters for tuning in both LSTM and GRU models, aiming to identify the most effective feature extractor model. The LSTM hyperparameters tuned to build the feature extractor model in this study are listed in Table V.

TABLE V. THE LSTM HYPERPARAMETERS ARE TO BE TUNED

Hyperparameters	Values
Unit LSTM	[5, 10]
Dropout	[0.05, 0.01, 0.1, 0.2]
Epochs	[5, 10]
Batch Size	[32, 64]

Next, the training will be carried out on regressor models: extreme gradient boosting, random forest, and support vector regression. Table VI lists the hyperparameters that will be tested on each regressor algorithm. The XGBoost (XGB) model uses the hyperparameter known as the number of estimators, which determines how many decision trees the model constructs during the modelling process. A suboptimal number of estimators may result in insufficient boosting rounds to capture complex patterns within the data adequately. In contrast, an excessively high number may lead to overfitting. Therefore, it is essential to conduct hyperparameter tuning to identify the optimal value for n_estimators.

In the Random Forest model, the hyperparameters that undergo tuning include n_estimators, criterion, maximum depth, and maximum features. The rationale for tuning n_estimators in the Random Forest model parallels that of the XGB model, as both utilize a tree-based approach. Adjusting these parameters can minimize the Mean Absolute Error (MAE) during the training and validation phases. The maximum depth parameter limits the tree depth permitted by the model. A too large depth may cause the model to capture noise and specific patterns unique to the training dataset. The maximum features parameter indicates the upper limit on the number of features the algorithm considers when dividing tree segments.

TABLE VI. THE REGRESSOR HYPERPARAMETERS ARE TO BE TUNED

Hyperparameter	XGB	RF	SVR
N_estimators	[10, 25, 50, 100, 150]	[10, 25, 50]	N/A
Criterion	N/A	[Absolute error]	N/A
Max depth	N/A	[None, 10, 25, 50]	N/A
Max features	N/A	[auto, sqrt, log2]	N/A
C	N/A	N/A	[0.01, 0.1, 1, 10, 100]
Epsilon	N/A	N/A	[0.01, 0.1, 1, 10, 100]
Gamma	N/A	N/A	[scale, auto]
Kernel	N/A	N/A	[rbf]

In the Support Vector Regression (SVR) model, the hyperparameters C, epsilon, and gamma require tuning. The tuning of the C parameter is critical for balancing the minimization of error values in the training dataset while ensuring a smooth regression function. Epsilon represents the margin of tolerance surrounding the actual target value, within which predictions incur no penalty. Conversely, the gamma parameter governs the influence of individual training examples, with a higher gamma leading to a more localized area of influence for each data point.

The hyperparameter value utilized in this study's feature extractor model is determined by analyzing the results from experiments conducted with single Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models. Meanwhile, the hyperparameters employed in the regressor model are selected based on the outcomes of hyperparameter tuning via the grid search algorithm. In this study, the researchers implement early stopping to prevent overfitting.

4) GRU-machine learning

We will carry out this procedure using the preprocessed dataset. We will train the feature extractor model to create a GRU feature extraction model. Now, we will train the regressor model. We will validate this model before evaluating it. If the model achieves satisfactory results, we will proceed with model testing; if not, we will perform hyperparameter tuning. The last stage of this scheme is to evaluate the model using test data. The process of building the GRU-machine learning model is depicted in Fig. 9. We will record the results from this test to compare them with other models and identify the best one.

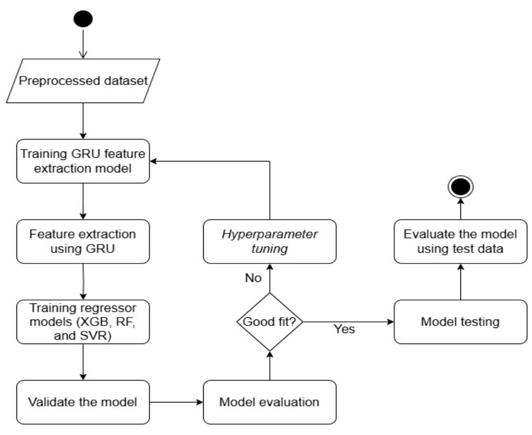

Fig. 9. Experiment flow of GRU-machine learning.

In this study, the GRU model will be trained with the hyperparameters listed in Table VII. After training the GRU model, the extracted data features will be used to train the regressor model. The parameters used for training the regressor model are listed in Table VI.

TABLE VII. THE GRU HYPERPARAMETERS ARE TO BE TUNED

Hyperparameters	Values
Unit GRU	[5, 10]
Dropout	[0.05, 0.01, 0.1, 0.2]
Epochs	[5, 10]
Batch Size	[32, 64]

5) Model evaluation

The model that has been made in this research will be evaluated using the Mean Absolute Error (MAE) and R^2 Score loss functions. The reason for using MAE to evaluate the results of this study refers to research conducted by Jadon *et al.* [33] that data that has many outliers is more suitable to be evaluated using Mean Squared Error, while data that has few outliers is more suitable to be evaluated using Mean Absolute Error.

After checking the outlier data, it is stated that this data does not have outliers, so the loss function used is Mean Absolute Error. Eq. (12) is used to calculate the MAE value, while Eq. (13) is used to calculate the R^2 Score value.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{12}$$

$$R^2\ Score = 1 - \frac{\sum(y_i - \hat{y_i})^2}{\sum(y_i - \bar{y_i})^2} \tag{13}$$

IV. RESULT AND DISCUSSION

A. LSTM-Machine Learning

The research results for the LSTM machine learning method will be divided into three subsections: LSTM-XGB, LSTM-RF, and LSTM-SVR. Each model will consist of two schemes for testing with window sizes 1 and 5.

Before forecasting using the LSTM-machine learning or GRU-machine learning methods, researchers first tested the LSTM and GRU models to predict crude oil prices. From the results of this experiment, the best hyperparameters were obtained to be applied to the LSTM feature extractor model. The best hyperparameters are listed in Table V. In the table, two models will be used as a comparison to obtain the best model when the LSTM-machine learning model is built.

TABLE VIII. HYPERPARAMETERS FOR THE LSTM FEATURE EXTRACTOR

Window size	Model	Unit	Dropout	Epochs	Batch Size
1	1	10	0.01	10	32
	2	10	0.05	10	32
5	1	5	0.01	10	32
	2	10	0.01	10	32

1) LSTM-XGB

a) Model experiment with window size 1

The feature extraction model was built using the hyperparameters listed in Table VIII to perform this experiment. The regressor model utilizes a hyperparameter, the number of estimators, configured to 25. In this study, five experiments were carried out on each model, and the best model was obtained with the MAE error value and R^2 Score at the training and testing stages. We also considered the time execution, which is all the time needed for feature extraction and regression execution for each model. All these criteria are listed in Table IX.

Based on the error value and R^2 Score, these two models have the same error value, so the execution time of the model is considered. The execution time needed by LSTM-XGB 2 is longer than that of LSTM-XGB 1, which only takes 0.22127 seconds using CPU (Central Processing Unit), and 0.12286 seconds using GPU (Graphics Processing Unit). Based on the results of this analysis, it is decided that the LSTM-XGB 1 is the best model that can be generated in the LSTM-XGB research for a window size value of 1.

TABLE IX. MODEL EVALUATION LSTM-XGB FOR WINDOW SIZE 1

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-XGB 1	1.053	24.2947	0.997	1.605	23.7427	0.971	0.23060	0.14958
LSTM-XGB 2	1.053	24.2947	0.997	1.605	23.7427	0.971	0.22127	0.12286

b) Model experiment using window size 5

The hyperparameters to be used in the feature extractor model are listed in Table VIII. As for the hyperparameters used in the regressor model, there are n-estimators of 25. The best test results from this experiment are listed in Table X.

TABLE X. MODEL EVALUATION LSTM-XGB FOR WINDOW SIZE 5

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-XGB 3	1.486	23.8617	0.994	2.162	23.1857	0.950	0.15691	0.15234
LSTM-XGB 4	1.558	23.7897	0.994	2.229	23.1187	0.947	0.28134	0.20633

The hyperparameters to be used in the feature extractor model are listed in Table V. As for the hyperparameters used in the regressor model, there are n-estimators of 25. The best test results from this experiment are listed in Table XX. Based on Table X, we have seen that the MAE error value for LSTM-XGB 3 was lower than the model LSTM-XGB 4. This means that model LSTM-XGB 3 performs better in predicting crude oil prices in this scheme. We can also see that model LSTM-XGB 3 has the shortest execution time. So, for this scheme, we can conclude that model LSTM-XGB 3 was the best model.

2) LSTM-RF

a) Model experiment with window size 1

In Table VIII, you can find a comprehensive list of the hyperparameters used in developing the feature extractor model. This table presents the values and settings we selected to enhance the model's performance. Similarly, Table XI presents the hyperparameters utilized for the regressor model, detailing the configurations contributing to its predictive accuracy. Each table is a valuable reference for understanding the distinct parameters influencing the respective models.

The LSTM-RF model tested in this study will use the best hyperparameters of the feature extractor and regressor. Based on Table XII, the execution time required by the LSTM-RF 1 model is less than that of the LSTM-RF 2. The MAE value produced by the LSTM-RF 2 for the training and testing process is smaller than that of the LSTM-RF 1 model. This shows that the LSTM-RF 2 has better performance compared to the LSTM-RF 1 model.

TABLE XI. HYPERPARAMETERS OF THE RF REGRESSOR FOR WINDOW SIZE 1

Model	Criterion	Max depth	Max features	n-estimators
RF 1	Absolute error	10	Auto	25
RF 2	Absolute error	10	Auto	50

TABLE XII. MODEL EVALUATION LSTM-RF FOR WINDOW SIZE 1

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-RF 1	0.836	24.5117	0.998	1.709	23.6387	0.969	0.14247	0.14827
LSTM-RF 2	0.831	24.5167	0.998	1.684	23.6637	0.969	0.15743	0.15390

b) Model experiment with window size 5

The content of Table XIII provides a detailed overview of the hyperparameters used during the model experimentation process, particularly for configurations with a window size of 5. We list each hyperparameter and its corresponding settings, which we carefully selected to optimize model performance for this specific window size. Table XIII lists the feature extractor models that will be used, with the hyperparameters used being models 1 and 2 with a window size of 5. We will pair LSTM model 1 with regressor model RF 3 as LSTM-RF 3 and LSTM model 2 with regressor model RF 4 as LSTM-RF 4.

The evaluation results of this best model are listed in Table XIV. Based on the error value obtained, the LSTM-RF 4 model has the most minor error value both in the training and testing stages. There is a slight difference in MAE value between LSTM-RF 3 and LSTM-RF 4, valued at 0.038. This shows that the LSTM-RF 4 model can predict crude oil prices well because this model's predictions are closer to the actual values. In terms of execution time using CPU, the LSTM-RF 3 model has a shorter execution time than the LSTM-RF 4 model. However, the LSTM-RF 4, has the shortest GPU execution time, which only takes 0.13131 seconds. In this research, we focused on generating the best model that can predict the crude oil prices to be close to the actual ones. So, we chose the LSTM-RF 4 as the best model for this scheme.

TABLE XIII. HYPERPARAMETERS OF THE RF REGRESSOR FOR WINDOW SIZE 5

Model	Criterion	Max depth	Max features	n-estimators
RF 3	Absolute error	10	Log2	50
RF 4	Absolute error	10	Auto	50

TABLE XIV. MODEL EVALUATION LSTM-RF FOR WINDOW SIZE 5

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-RF 3	1.215	24.1327	0.996	2.264	23.0837	0.946	0.17041	0.15100
LSTM-RF 4	1.177	24.1707	0.996	2.189	23.1587	0.948	0.27384	0.13131

3) LSTM-SVR

a) Model experiment with window size 1

The SVR regressor hyperparameters used in the study for the LSTM-SVR 1 and LSTM-SVR 2 models are hyperparameters of the same value and are listed in Table XV. The hyperparameters that will be used in the LSTM feature extractor model are listed in Table VIII.

Table XVI summarizes the evaluation results for the best model obtained from the testing scheme. The test results indicate that the LSTM-SVR 1 has the shortest execution time while using the CPU. However, the LSTM-SVR has the shortest execution time while using GPU. This process only takes 0.16645 seconds. However, upon examining the Mean Absolute Error (MAE) during testing, it becomes clear that the LSTM-SVR 2 model performs better. The experiments demonstrate that the LSTM-SVR 2 model is the superior choice.

TABLE XV. HYPERPARAMETERS OF THE SVR REGRESSOR FOR WINDOW SIZE 1

Model	Kernel	C	Epsilon	Gamma
SVR	RBF	100	0.01	Auto

TABLE XVI. MODEL EVALUATION LSTM-SVR FOR WINDOW SIZE 1

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-SVR 1	1.051	24.2967	0.997	1.560	23.7877	0.972	0.18722	0.16872
LSTM-SVR 2	1.050	24.2977	0.997	1.558	23.7897	0.973	0.24096	0.16645

b) Model experiment with window size 5

This scheme provides a detailed overview of the hyperparameters specified in Table XVII. Notably, the regressor hyperparameter values utilized in constructing the LSTM-SVR 3 and LSTM-SVR 4 models are identical, ensuring consistency across these models. Furthermore, Table XVIII displays the outcomes of the model evaluation, highlighting the performance metrics and findings associated with each model configuration.

The experiment results show that the MAE error value at the training and testing for LSTM-SVR 4 is smaller than the error value produced by LSTM-SVR 3. This means that LSTM-SVR 4 has the best predictions that are generally close to the actual price. The R^2 Score value for this model means that the model can explain 95.0% of the variance in the target variable. This means the model captures most of the data patterns. Also, this model has the shortest execution time. So, we conclude that LSTM-SVR 4 is the best model for this scheme.

TABLE XVII. HYPERPARAMETERS OF THE SVR REGRESSOR FOR WINDOW SIZE 5

Model	Kernel	C	Epsilon	Gamma
SVR	RBF	10	0.01	Auto

TABLE XVIII. MODEL EVALUATION LSTM-SVR FOR WINDOW SIZE 5

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-SVR 3	1.622	23.7257	0.993	2.248	23.0997	0.947	0.62126	0.26949
LSTM-SVR 4	1.561	23.7867	0.994	2.186	23.1617	0.950	0.37492	0.20602

From the experimental results of the LSTM-machine learning model for both window sizes 1 and 5, the best model was found, namely the LSTM-SVR 2 model, where this model is a model with window size 1. Although the MAE training value produced by the LSTM-RF 2 model is smaller, the MAE testing value for the LSTM-SVR 2 model is lower than that of the LSTM-RF 2 model. This indicates that while the LSTM-RF 2 model performs better during training, the LSTM-SVR 2 model demonstrates superior performance during testing. The slight difference between the training and testing error values in the LSTM-SVR 2 model compared to the LSTM-RF 2 model led us to conclude that the LSTM-SVR 2 model is the best choice for the LSTM-machine learning research scheme.

The hyperparameters used to build the LSTM feature extraction model consist of 10 LSTM units, with a dropout of 0.05, 10 epochs, and a batch size 32. The hyperparameters used to build the regressor model are SVR models with RBF kernels using C of 100, epsilon of 0.01, and gamma set to auto.

The LSTM-SVR 2 model has an MAE error value of 1.050 and an R^2 Score of 0.997 at the training stage. The LSTM-SVR 2 model has demonstrated the ability to capture the crude oil price patterns during the training stage using the loss function. Therefore, it is ready to proceed with the validation and testing processes. In the testing phase, we found that the LSTM-SVR 2 model delivered the lowest MAE value among all the models evaluated. This shows that this model can capture the patterns in the dataset and has good generalization capabilities. The MAE error value obtained by this model at the testing stage is 1.558, and the R^2 Score value on this model is 0.973. The slight difference in the MAE error value and R^2 Score at the training and testing stages shows that this model has good performance where the model can explain 97.3% of the variance in the testing data, and this model has good stability.

B. GRU-Machine Learning

The research results for the GRU-machine learning model will be explained in subsection 1) for the GRU-XGB model, subsection 2) for the GRU-RF model, and subsection 3) for the GRU-SVR model. Each of these test schemes will consist of two parts to explain the results of testing the model with windows 1 and 5.

To determine the hyperparameters that will be used for feature extraction, a trial was previously conducted on the single GRU model, which obtained the best hyperparameters that provided the smallest error value at that time. The hyperparameters that will be used in this study are written in Table XIX. Furthermore, the hyperparameters will be used to build feature extractor models at window sizes 1 and 5.

TABLE XIX. HYPERPARAMETERS FOR THE GRU FEATURE EXTRACTOR

Window size	Model	Unit	Dropout	Epochs	Batch Size
1	1	10	0.01	10	32
	2	10	0.05	10	32
5	1	5	0.05	10	32
	2	10	0.01	10	32

1) GRU-XGB

a) Model experiment with window size 1

To build the GRU-XGB model, tests were conducted to find the correct number of n-estimators to obtain a model with the best loss value and performance. The experiment results show that the best hyperparameter is n-estimators worth 25. The results of the evaluation of the loss value and R^2 Score of the GRU-XGB model are listed in Table XX.

TABLE XX. MODEL EVALUATION GRU-XGB FOR WINDOW SIZE 1

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
GRU-XGB 1	1.053	24.2947	0.997	1.605	23.7427	0.971	0.14190	0.13769
GRU-XGB 2	1.053	24.2947	0.997	1.605	23.7427	0.971	0.13969	0.14535

Both models exhibit the same values for the Mean Absolute Error (MAE) and R^2 Score, indicating equal performance in terms of accuracy. As shown in Table XX, we will focus on execution time to identify the superior model. The GRU-XGB 2 model has the fastest execution time, completing its run in just 0.13969 seconds using the CPU. The execution times of the two models differ by only 0.00933 seconds, reflecting a negligible gap. Despite this minor difference, it is clear that GRU-XGB 2 is the better choice based on its efficiency.

b) Model experiment with window size 5

The results of the experiment with this model are listed in Table XXI. From the table, it is known that the MAE training and testing values on the GRU-XGB 3 model are smaller than the GRU-XGB 3 model, while the R^2 Score value of the GRU-XGB 3 model is greater than the GRU-XGB 4 model. The GRU-XGB 3 model shows an ability to accurately predict crude oil prices. However, if you look at the overall execution time, the GRU-XGB 4 model is faster than the GRU-XGB 3 model, so it can be concluded that GRU-XGB 3 was the best model.

TABLE XXI. MODEL EVALUATION GRU-XGB FOR WINDOW SIZE 5

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
GRU-XGB 3	1.191	24.1567	0.996	1.763	23.5847	0.965	0.18558	0.13538
GRU-XGB 4	1.283	24.0647	0.996	1.842	23.5057	0.962	0.12516	0.14762

2) GRU-RF

a) Model experiment with window size 1

A detailed overview of the hyperparameters used in the regressor model is presented in Table XXII, emphasizing the specific configurations that affect its performance. We use several key metrics to evaluate the GRU-RF model: the Mean Absolute Error (MAE) to assess prediction accuracy, the R^2 Score to determine the proportion of variance the model explains, and the execution time for both the feature extraction and regression processes. These metrics are thoroughly presented in Table XXIII, facilitating an in-depth comparison of the models' performance.

The experiment's results show that the GRU-RF 2 model requires faster execution time than the GRU-RF 1 model. In addition, the GRU-RF 2 model has a smaller MAE value than the GRU-RF 1 model's MAE value both in the training and testing stages. Based on the loss value and the resulting execution time, the GRU-RF 2 model is the best model in this scheme.

TABLE XXII. HYPERPARAMETERS OF THE RF REGRESSOR FOR WINDOW SIZE 1

Model	Criterion	Max depth	Max features	n-estimators
RF 1	Absolute error	10	Log2	50
RF 2	Absolute error	10	Auto	50

TABLE XXIII. MODEL EVALUATION GRU-RF FOR WINDOW SIZE 1

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
GRU-RF 1	0.835	24.5127	0.998	1.682	23.6657	0.969	0.23767	0.29022
GRU-RF 2	0.833	24.5147	0.998	1.674	23.6737	0.970	0.13970	0.21146

b) Model experiment with window size 5

A detailed summary of the random forest hyperparameters used in the development of this model is provided in Table XXIV. In addition, Table XXV summarizes the experiments' results, explicitly showcasing the two optimal models characterized by the lowest loss values and the shortest execution times observed across five trials.

Experimental results show that the GRU-RF 4 model requires a longer execution time when using the CPU but is very fast when using the GPU. For both the training and testing stages, the GRU-RF 4 model's MAE value is smaller than that of the GRU-RF 3 model. This shows that the GRU-RF 4 model is good at predicting crude oil prices but requires a longer time.

TABLE XXIV. HYPERPARAMETERS OF THE RF REGRESSOR FOR WINDOW SIZE 5

Model	Criterion	Max depth	Max features	n-estimators
RF 3	Absolute error	10	Sqrt	50
RF 4	Absolute error	10	Auto	50

TABLE XXV. MODEL EVALUATION GRU-RF FOR WINDOW SIZE 5

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
GRU-RF 3	0.908	24.4397	0.998	1.762	23.5857	0.965	0.18196	0.17584
GRU-RF 4	0.832	24.5157	0.998	1.686	23.6617	0.969	0.23974	0.17117

3) GRU-SVR

a) Model experiment with window size 1

The SVR regressor hyperparameters used in this scheme are listed in Table XV. It can be seen in Table XXVI that the MAE and R^2 Score values at the training stage of the GRU-SVR 1 and GRU-SVR 2 models have the same value. In the testing stage, the same MAE value is obtained. The R^2 Score value owned by the GRU-SVR 2 model is excellent, which means that this model has a better generalization ability. The GRU-SVR 2 captures slightly more of the data patterns than GRU-SVR 1. However, a drawback of this model is that GRU-SVR 2 has a longer execution time than GRU-SVR 1. So, we conclude that GRU-SVR 2 was the best model for this scheme.

TABLE XXVI. MODEL EVALUATION GRU-SVR FOR WINDOW SIZE 1

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
GRU-SVR 1	1.050	24.2977	0.997	1.559	23.7887	0.972	0.33088	0.15664
GRU-SVR 2	1.050	24.2977	0.997	1.559	23.7887	0.973	0.52961	0.18173

b) Model experiment with window size 5

The hyperparameters of the regressor model used in this scheme are the same as those listed in Table XV. The results of this scheme's experiment are written in Table XXVII. The MAE value produced by the GRU-SVR 4 model is smaller than the GRU-SVR 3 model. In comparison, the R^2 Score value of the GRU-SVR 4 model is excellent in both the training and testing stages. Based on the R^2 Score value, we can conclude that this model has a good generalization ability. Regarding to the CPU execution time, the GRU-SVR 4 model has the shortest execution time, so it can be concluded that the best model for this scheme is the GRU-SVR 4 model.

TABLE XXVII. MODEL EVALUATION GRU-SVR FOR WINDOW SIZE 5

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
GRU-SVR 3	1.350	23.9977	0.995	1.894	23.4537	0.961	0.23162	0.18063
GRU-SVR 4	1.296	24.0517	0.996	1.831	23.5167	0.964	0.22139	0.19519

In this scheme, six GRU-machine learning models for window size one and six models for window size five have been tested. From all the experiments conducted for the GRU-machine learning model scheme for window sizes 1 and 5, the best model will be determined, and the focus will be on the error value generated by the model.

The results of this study show that the GRU-RF 4 and GRU-RF 2 models have the slightest error value at the training stage, with an R^2 Score value of 0.998. However, the evaluation results at the testing stage show that the GRU-SVR 2 model in the GRU-machine learning scheme has the smallest MAE value and the most significant R^2 Score value. The GRU-SVR 2 model has an MAE error value at the training stage of 1.05, with an R^2 Score of 0.997.

This shows that the model can capture price patterns in the dataset. Then, the MAE value at the testing stage is 1.559, with the largest R^2 Score value among other models, 0.973. The difference in MAE error values at the training and testing stages of this model shows that this model has a small train-test value difference. This indicates that the model has good performance and stability.

The analysis results indicate that the GRU-SVR 2 is the most successful model produced by the GRU-machine learning framework. The hyperparameter used to build this model is one GRU layer containing 10 GRU units, a dropout of 0.05, epochs performed as many as 10, and a batch size 32. Meanwhile, the hyperparameter of the regressor model used is an RBF kernel type, with C 100, epsilon 0.01, and gamma, which is auto.

C. Evaluation of LSTM-Machine Learning and GRU-Machine Learning Models

1) Model evaluation with window size 1

This study identifies the most effective model for each analyzed scheme, as detailed in Table XXV. The evaluation encompasses Mean Absolute Error (MAE) and R^2 scores for both the training and testing phases, alongside considerations of execution time. Initially, the assessment of the training set's loss value and R^2 score reveals that the model with the lowest MAE value is LSTM-RF 2. Furthermore, the models that exhibit the highest R^2 scores are LSTM-SVR 2 and GRU-RF 2. The testing phase results indicate that the LSTM-SVR 2 model demonstrates the smallest MAE value compared to the other models. The R^2 scores further substantiate that LSTM-SVR 2 and GRU-SVR 2 achieve the highest values.

TABLE XXVIII. MODEL EVALUATION FOR BEST MODEL IN WINDOW SIZE 1

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-XGB 2	1.053	24.2947	0.997	1.605	23.7427	0.971	0.22127	0.12286
LSTM-RF 2	0.831	24.5167	0.998	1.684	23.6637	0.969	0.15743	0.15390
LSTM-SVR 2	1.050	24.2977	0.997	1.558	23.7897	0.973	0.24096	0.16645
GRU-XGB 2	1.053	24.2947	0.997	1.605	23.7427	0.971	0.13969	0.14535
GRU-RF 2	0.833	24.5147	0.998	1.674	23.7427	0.970	0.13970	0.21146
GRU-SVR 2	1.050	24.2977	0.997	1.559	23.7887	0.973	0.52961	0.18173

Based on the MAE and R^2 Score values obtained by the model at the training and testing stages, it is known that the LSTM-RF 2 model is a model that has the potential to be the best model because it has the smallest training loss value and the highest R^2 Score value, but for the LSTM-SVR 2 training process which has the most minor error value and the highest R^2 Score value.

The data in Table XXVIII analyzes the performance of the LSTM-RF 2 model in predicting crude oil prices. The model's predictions closely align with actual prices, although the Mean Absolute Error (MAE) increased by 0.853 during testing, indicating a slight increase in prediction error. The R^2 score was nearly 1 in both the training and testing phases, showcasing the model's ability to capture and explain data variability effectively. Additionally, the model's processing time for generating predictions was efficient at 0.15743 seconds using CPU, and 0.15390 seconds using GPU, which is crucial for timely decision-making in crude oil trading. These results highlight the LSTM-RF 2 model's effectiveness in price prediction.

The LSTM-SVR 2 model has a slightly higher error value than the LSTM-RF 2 model in the training stage. However, this model has the smallest MAE value at the testing stage. The MAE value at the testing stage of this model has increased by 0.508. The increase in the error value experienced by this model is slightly lower than the LSTM-RF 2 model. This indicates that the model performs well and has strong generalization abilities, enabling it to predict testing data with a low margin of error accurately. In the training phase, it is known that the model can explain 99.7% of data variability and capture data patterns.

Furthermore, at the testing stage, the R^2 Score value decreased to 97.3%, but the decrease was still within normal limits. Apart from the MAE value, the R^2 Score value also shows that the model has good generalization ability to new data. So, based on the results of this analysis, it is concluded that the best model in this study is LSTM-SVR 2.

During the training phase, a comprehensive comparative graph illustrates the relationship between actual crude oil prices and the predictions generated by the LSTM-SVR 2 model, as depicted in Fig. 10. In this graph, the blue line effectively represents the actual price fluctuations of Brent crude oil, capturing its temporal variations. Conversely, the dashed pink line indicates the forecasted prices produced by the model, visually representing its predictive accuracy. Significantly, the pink line closely adheres to the trajectory of the blue line, indicating a substantial correlation between the model's predictions and the observed actual prices. This visual comparison not only underscores the model's efficacy but also elucidates the error values generated during the training process, offering valuable insights into the model's performance and reliability.

The illustration presented in Fig. 11 comprehensively compares the actual crude oil prices and the forecasts produced by the LSTM-SVR 2 model during the testing phase. Notably, the pink line, which signifies the model's predictions, closely aligns with the blue line, representing the actual prices. This strong correspondence indicates that the model has effectively captured and predicted the variations in crude oil prices, demonstrating its efficacy in light of the preceding analytical work.

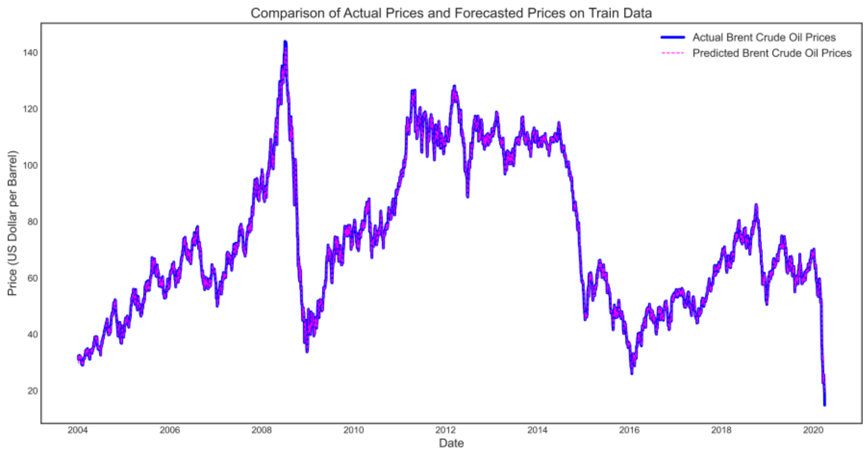

Fig. 10. Comparison of actual prices and predictions of the LSTM-SVR 2 model at the training stage.

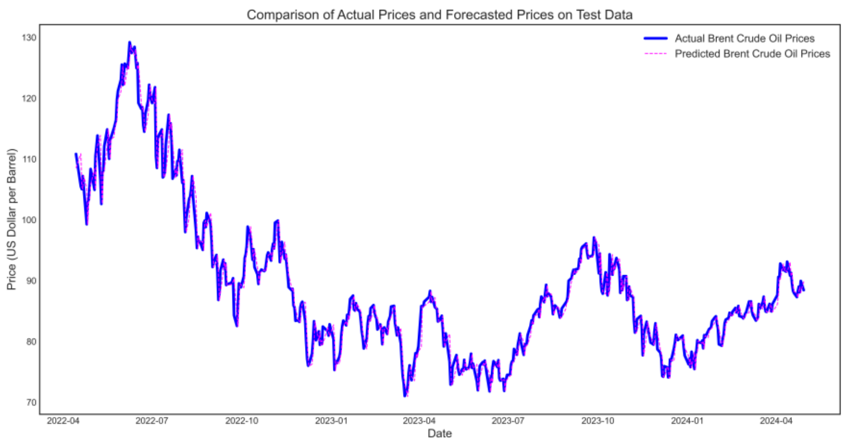

Fig. 11. Comparison of actual prices and predictions of the LSTM-SVR 2 model at the testing stage.

2) Model evaluation with window size 5

Table XXVI displays the optimal evaluation results from various LSTM-machine learning and GRU-machine learning model schemes with a window size 5. The findings reveal that the GRU-RF 4 model is the best option in this category. It has the lowest Mean Absolute Error (MAE) values for both the training and testing phases and the highest R^2 Score values for each phase.

According to the MAE and R^2 Score values, it can be inferred that the GRU-RF 4 model can forecast crude oil prices that closely align with the actual prices and demonstrate strong generalization performance. Fig. 12 compares the actual Brent crude oil price with the predicted price from the best model, GRU-RF 4. The graph shows that there is almost no gap between the actual price and the predicted price, which is supported by the MAE evaluation results in Table XXIX.

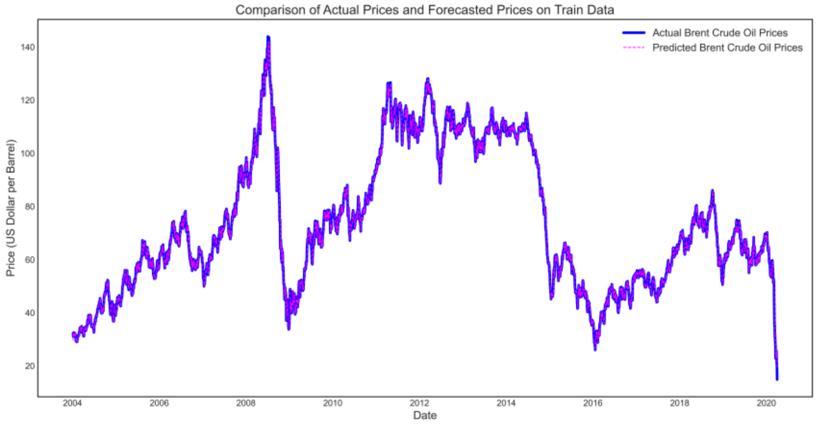

Fig. 12. Comparison of actual prices and predictions of the GRU-RF 4 model at the training stage.

TABLE XXIX. MODEL EVALUATION FOR BEST MODEL IN WINDOW SIZE 5

Model	Train			Test			Time Execution using CPU (s)	Time Execution using GPU (s)
	MAE	Difference of Standard Deviation with MAE	R^2 Score	MAE	Difference of Standard Deviation with MAE	R^2 Score		
LSTM-XGB 3	1.486	23.8617	0.994	2.162	23.186	0.950	0.15691	0.15234
LSTM-RF 4	1.177	24.1707	0.996	2.189	23.1587	0.948	0.27384	0.13131
LSTM-SVR 4	1.561	23.7867	0.994	2.186	23.1617	0.950	0.37492	0.20602
GRU-XGB 3	1.191	24.1567	0.996	1.763	23.5847	0.965	0.18558	0.13538
GRU-RF 4	0.832	24.5157	0.998	1.686	23.6617	0.969	0.23974	0.17117
GRU-SVR 4	1.296	24.0517	0.996	1.831	23.5167	0.964	0.22139	0.19519

The comparison graph presented in Fig. 13 delineates the relationship between actual prices and the predicted values generated by the GRU-RF 4 model. The close alignment of the two lines indicates the model's notable efficacy in generalizing and accurately capturing underlying data patterns, resulting in price predictions that are in strong concordance with the observed values. This minor disparity between the predicted and actual prices underscores the model's robustness and reliability in forecasting.

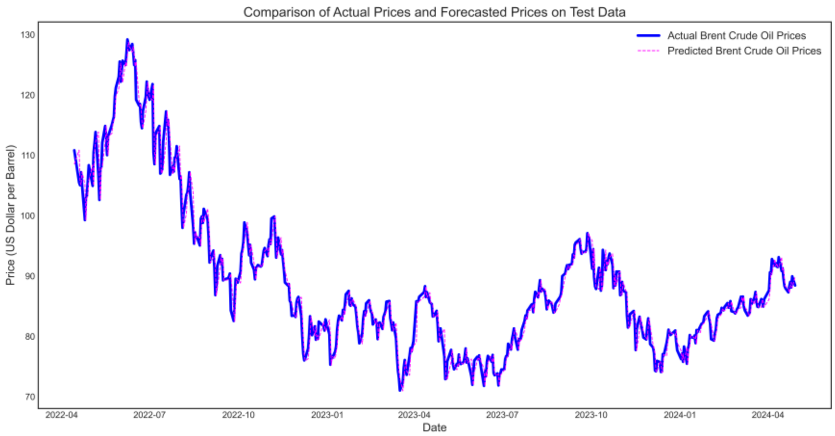

Fig. 13. Comparison of actual prices and predictions of the GRU-RF 4 model at the testing stage.

V. CONCLUSION

This study focuses on extracting features from time series data of crude oil prices using LSTM and GRU models. In this research, we conduct a regression analysis using various machine-learning techniques. We apply extreme gradient boosting, random forest, and support vector regression as the regression algorithms. We evaluate the created model to determine which one is the best.

The results of this study show that the model built using window size 1 has better performance than the model built using window size 5. The research shows that the LSTM-

SVR 2 model performs best in this study. The MAE value produced by this model at the training stage is 1.050, while the MAE value at the testing stage is 1.558. The slight difference in MAE values at the training and testing stages shows that this model has stability. In addition, the error value generated by this model is relatively small, indicating that this model provides prediction results that are close to the actual price.

We evaluated the model using Mean Absolute Error (MAE) and assessed its performance with the R^2 score. The R^2 score at the training stage was 99.8%, and at the testing stage, it was 97.3%. The high R^2 Score value produced by the model shows that it can explain almost all the variations in the dataset very well. The execution time required by the LSTM-SVR 2 model, whether CPU or GPU, is notably efficient, with execution times consistently remaining below 0.5 s. Based on the two types of evaluations conducted and their respective execution times, we conclude that the model demonstrates strong performance. It generates predicted prices that closely align with actual prices and exhibits robust generalization ability.

The hyperparameters used in the LSTM-SVR 2 model include LSTM and SVR hyperparameters. The LSTM model's hyperparameter details include 10 LSTM units with a dropout value of 0.05, 10 epochs, and a batch size of 32. Meanwhile, the SVR model is built using an RBF kernel with a hyperparameter of C worth 100, epsilon 0.01, and gamma, which is worth auto.

Predicting crude oil prices is a complex task that involves various factors. Global supply and demand dynamics, geopolitical events, regulation changes, and economic conditions can influence prices. Natural disasters and advancements in extraction technology can also play significant roles. As a result, accurately forecasting future prices requires careful analysis and consideration of current market trends and potential future developments. This is due to various factors that can affect them and cannot be controlled. This research uses feature extraction to capture patterns and essential information in the data to predict crude oil prices as close as possible to the actual prices. Therefore, to create a more accurate and applicable model, it is suggested that future researchers consider several factors that may affect the movement of crude oil prices.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Rifdah Amelia conducted the research and authored the paper, while Lili Ayu Wulandhari provided supervision throughout the research process and the writing of the manuscript. All authors reviewed and approved the final version of the paper.

FUNDING

This research project received financial support from Bina Nusantara University in the amount of USD 415. This funding was instrumental in facilitating the various phases of the research, allowing the team to conduct comprehensive research and analyses.

ACKNOWLEDGMENT

The researchers sincerely appreciates Bina Nusantara University for their invaluable support in this research endeavor. The program and data used in this research can be accessed at the following site https://github.com/Rifdaha/Crude-Oil-Price-Forecasting.

REFERENCES

[1] G. A. Olah, A. Monlár, and G. K. S. Parkash, *Hydrocarbon Chemistry*, 3rd ed., Hoboken, NJ: John Willey & Sons, Inc, 2018, ch. 1, p. 16.

[2] M. Hasan *et al.* (January 2024). A blending ensemble learning model for crude oil price forecasting. *Annals of Operation Research*, [Online]. pp. 1–31. Available: https://link.springer.com/article/10.1007/s10479-023-05810-8.

[3] P. G. Saculsan and T. Kanamura, "Examining risk and return profiles of renewable energy investment in developing countries: The case of the Philippines," *Green Finance*, vol. 2, no. 2, pp. 135–150, Apr. 2020.

[4] L. Chen, Z. Zhang, F. Chen, and N. Zhou, "A study on the relationship between economic growth and energy consumption under the new normal," *National Accounting Review*, vol. 1, no. 1, pp. 28–41, June 2019.

[5] T. Li and G. Liao, "The heterogeneous impact of financial development on green total factor productivity," *Frontiers in Energy Research*, vol. 8, 29, Mar. 2020.

[6] K. Zhang and M. Hong, "Forecasting crude oil price using LSTM neural networks," *Data Science in Finance and Economics*, vol. 2, no. 3, pp. 163–180, July 2022.

[7] L. Guo, X. Huang, Y. Li, and H. Li, "Forecasting crude oil futures price using machine learning methods: Evidence from China," *Energy Economics*, vol. 127, 107089, Nov. 2023.

[8] M. Gyagri, E. M. Amarfio, and S. A. Marfo, "Determinants of global pricing of crude oil—A theoretical review," *International Journal of Petroleum and Petrochemical Engineering*, vol. 3, no. 3, pp. 7–15, July 2017.

[9] C. Wu *et al.*, "Influencing factors analysis of crude oil futures price volatility based on mixed-frequency data," *Applied Sciences*, vol. 10, no. 23, 8393, Nov. 2020.

[10] Y. Lin *et al.*, "Forecasting crude oil futures prices using BiLSTM-Attention-CNN model with wavelet transform," *Applied Soft Computing*, vol. 130, 109723, Nov. 2022.

[11] S. Akil, S. Sekkate, and A. Adib, "Exploring machine learning techniques for oil price forecasting: A comparative study of SVM, SMO, and SGD-base models," *Procedia Computer Science*, vol. 232, pp. 924–933, March 2024.

[12] P. Foroutan and S. Lahmiri, "Deep learning systems for forecasting the prices of crude oil and precious metals," *Financial Innovation*, vol. 10, no. 1, 111, July 2024.

[13] K. Gulati *et al.*, "Crude oil prices predictions in India using machine learning based hybrid model," in *Proc. 2022 10th International Conf. on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2022, pp. 1–6.

[14] Q. Yang *et al.*, "Forecasting crude oil futures prices using extreme gradient boosting," *Procedia Computer Science*, vol. 221, pp. 920–926, Jan. 2023.

[15] G. A. Busari and D. H. Lim, "Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance," *Computers and Chemical Engineering*, vol. 155, 107513, Dec. 2021.

[16] D. Malhotra and G. K. Sodhi, "Hybrid deep learning model for COVID-19 prediction using Convolutional Neural Network (CNN) and bidirectional Long Short-Term Memory (LSTM) network," *International Journal of Computer Theory and Engineering*, vol. 15, no. 3, pp. 125–129, Aug. 2023.

[17] K. E. Rajakumari, M. S. Kalyan, and M. V. Bhaskar, "Forward forecast of stock price using LSTM machine learning algorithm," *Int. J. Comput. Technol. Eng.*, vol. 12, no. 3, pp. 74–79, Jan. 2020.
[18] F. M. M. Alsheebah and B. Al-Fuhaidi, "Emerging stock market prediction using GRU algorithm: Incorporating endogenous and exogenous variables," *IEEE Access*, vol. 12, pp. 132964–132971, 2024. https://doi.org/10.1109/ACCESS.2024.3444699
[19] K. Li *et al.*, "Livestock product price forecasting method based on heterogeneous GRU neural network and energy decomposition," *IEEE Access*, vol. 9, pp. 158322–158330, Nov. 2021.
[20] H. Wang, X. Ye, Y. Li, and G. Zhu, "Remaining useful life prediction for lithium-ion batteries based on improved mode decomposition and time series," *Sustainability*, vol. 15, no. 12, 9176, June 2023.
[21] M. F. Maulana, S. Sa'adah, and P. E. Yunanto, "Crude oil price forecasting using long short-term memory," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 2, pp. 286–295, Sep. 2021.
[22] X. Ding, L. Fu, Y. Ding, and Y. Wang, "A novel hybrid method for oil price forecasting with ensemble thought," *Energy Reports*, vol. 8, pp. 15365–15376, Nov. 2022.
[23] K. He, Q. Yang, and Y. Zou, "Crude oil price prediction using embedding convolutional neural network model," *Procedia Computer Science*, vol. 214, pp. 959–964. Dec 2022.
[24] A. Jahandoost, M. Houshmand, and S. A. Hosseini, "Prediction of West Texas intermediate crude-oil price using hybrid attention-based deep neural networks: A comparative study," in *Proc. 2023 13th International Conf. on Computer and Knowledge Engineering (ICCKE)*, 2023, pp. 240–245.
[25] A. Sen and K. D. Choudhury, "Forecasting the crude oil prices for last four decades using deep learning approach," *Resources Policy*, vol. 88, 104438, Jan. 2024.
[26] S. Kostadinov. (December 2017). Understanding GRU networks. *Medium*. [Online]. Available: https://medium.com/towards-data-science/understanding-gru-networks-2ef37df6c9be.
[27] H. Aldabagh, X. Zheng, and R. Mukkamala, "A hybrid deep learning approach for crude oil price prediction," *Journal of Risk and Financial Management*, vol. 16, no. 12, 503, Dec. 2023.
[28] C. Olah. (August 2015). Understanding LSTM networks. *Colah's Blog*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[29] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv Preprint, arXiv: 1406.1078, 2014.
[30] J. Chung, C. Gulcehre, K. Cho, and B. Yoshua, "Gated feedback recurrent neural networks," in *Proc. 32nd International Conf. on Machine Learning*, 2015, pp. 2067–2075.
[31] R. Rodríguez-Pérez and J. Bajorath, "Evolution of support vector machine and regression modeling in chemoinformatics and drug discovery," *Journal of Computer-Aided Molecular Design*, vol. 36, no. 5, pp. 355–362, Mar. 2022.
[32] U.S. Energy Information Administration. What drives crude oil prices? *Eia*. [Online]. Available: https://www.eia.gov/finance/markets/crudeoil/spot_prices.php.
[33] A. Jadon, A. Patil, and S. Jadon, "A comprehensive survey of regression based loss functions for time series forecasting," in *Proc. International Conf. on Data Management, Analytics & Innovation*, 2024, pp. 117–147.