

FORECASTING GLOBAL BRENT CRUDE OIL PRICE BY STACKED  
DEEP LEARNING MODEL



AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN  
DATA SCIENCE AND ANALYTICS  
KMITL-DIGITAL ANALYTICS AND INTELLIGENCE CENTER SCHOOL OF SCIENCE  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG  
2023

KMITL-2023-SC-M-017-006

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.



**COPYRIGHT 2023**

**SCHOOL OF SCIENCE**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

**Independent Study Title :** Forecasting Global Brent Crude Oil Price by Stacked Deep Learning Model

**Student Name :** Marit Asavamahakul

**Student ID :** 64605091

**Degree :** Master of Science (Data Science and Analytics)  
KMITL-Digital Analytics and Intelligence Center

**Year :** 2023

**Independent Study Advisor :** Dr. Jiraphat Yokrattanasak



### ABSTRACT

This study's goal is to forecast crude oil prices using a multi-stacking model methodology. To improve the model's performance, the input characteristics, number of layers, and inner models (a combination of three models, including Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Artificial Neural Network (ANN)) will be changed. Data will be taken from TradingView, and the study period for the crude oil data will be from January 2016 to December 2021. The importance of this research rests in its ability to offer a crude oil price forecast model that is more accurate, which can help relevant parties create better situational plans. Future scholars may make modifications to the model's architecture and build upon it.

**Keywords:** Brent Crude Oil Price, ANN, GRU, LSTM, Multi-layer Stacked Model

## ACKNOWLEDGEMENTS

I cannot express enough thanks to my advisor for the continued support and encouragement: Dr. Jiraphat Yokrattanasak offer my sincere appreciation for the learning opportunities provided by my advisor.

My completion of this project could not have been accomplished without the support of my classmates, Pongpeera Jaruvijitrattan, Paweena Rouyprasert and Piyanuch Viwatborvornwong.

Finally, to my caring, loving, and supportive family for providing me anything I needed to let me focus on this research without any distractions.



Marit Asavamahakul

## TABLE OF CONTENTS

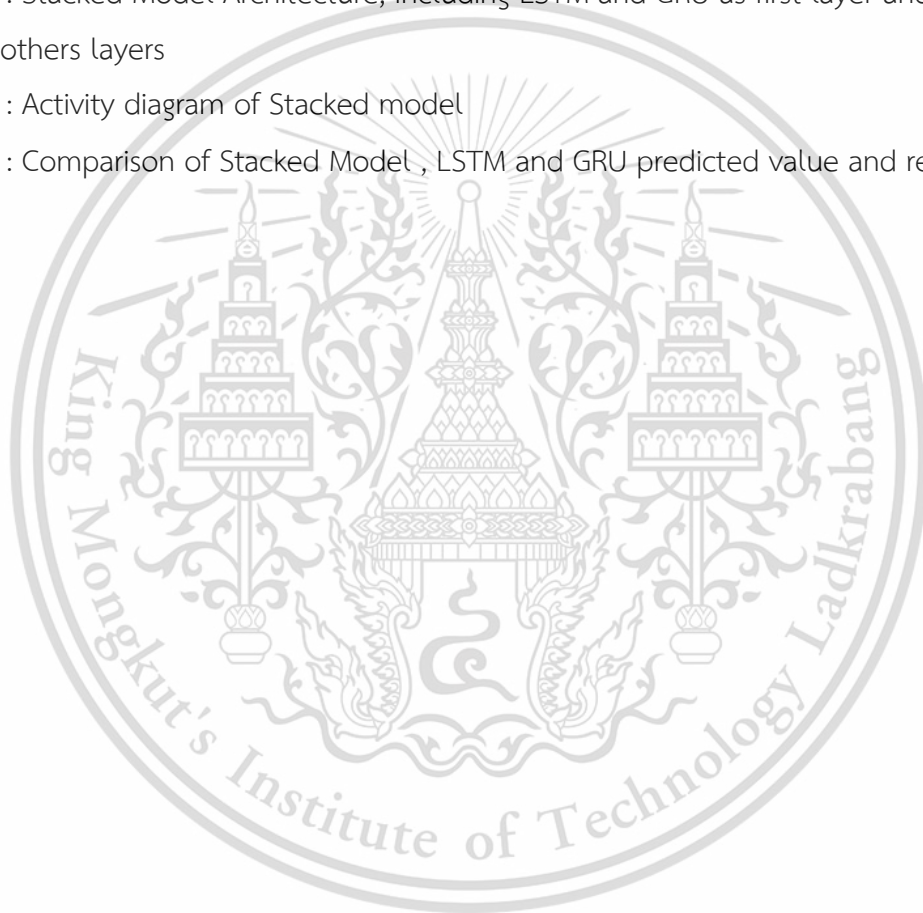
<i>ABSTRACT</i>	<i>A</i>
<i>ACKNOWLEDGEMENTS</i>	<i>B</i>
<i>TABLE OF CONTENTS : Table</i>	<i>D</i>
<i>TABLE OF CONTENTS : Figure</i>	<i>E</i>
<i>CHAPTER 1 INTRODUCTION</i>	<i>1</i>
1.1 Statement of Problems	1
1.2 Objectives	2
1.3 Research Scope	2
1.4 Significant of the study	2
<i>CHAPTER 2 LITERATURE REVIEW</i>	<i>3</i>
2.1 Data	3
2.2 Data Preparation	4
2.3 Modelling	6
2.4 Evaluation	14
2.5 Related Paper	15
<i>CHAPTER 3 RESEARCH METHODOLOGY</i>	<i>19</i>
3.1 Dataset Preparation	19
3.2 Stacking Model Architecture	21
3.3 Experiments	27
3.4 Evaluation	30
<i>CHAPTER 4 RESULT AND DISCUSSION</i>	<i>31</i>
<i>CHAPTER 5 CONCLUSION</i>	<i>35</i>
<i>REFERENCE</i>	<i>37</i>
<i>AUTHOR BIOGRAPHY</i>	<i>39</i>

## TABLE OF CONTENTS : Table

Table	Page
<b>No table of figures entries found.</b>	
3.1 : Partial sample data	20
3.2 : Setting of Parameters in LSTM - 1	23
3.3 : Setting of Parameters in LSTM - 2	23
3.4 : Setting of Parameters in LSTM - 3	23
3.5 : Setting of Parameters in LSTM - 4	23
3.6 : Setting of Parameters in LSTM - 5	24
3.7 : Setting of Parameters in GRU - 1	24
3.8 : Setting of Parameters in GRU - 2	24
3.9 : Setting of Parameters in GRU - 3	24
3.10 : Setting of Parameters in GRU - 4	25
3.11 : Setting of Parameters in GRU - 5	25
3.12 : Setting of Parameters in ANN - 1	25
3.13 : Setting of Parameters in ANN - 2	25
3.14 : Setting of Parameters in ANN - 3	26
3.15 : Setting of Parameters in ANN - 4	26
3.16 : Setting of Parameters in ANN - 5	26
3.17 : Setting of Parameters in ANN - 6	26
3.18 : Setting of Parameters in ANN - 7	27
3.19 : Setting of Parameters in ANN - 8	27
4.1 : Prediction performance of individual model in stacked model for Brent Crude Oil price prediction	32
4.2 : Prediction performance of based LSTM model, based GRU model and Stacked model	33

**TABLE OF CONTENTS : Figure**

Figure	Page
2.1 : Illustration of Candle stick	4
2.2 : LSTM's node architecture	6
2.3 : GRU's node architecture	10
2.4 : ANN's Architecture	12
3.1 : Experiment Workflow	19
3.2 : Training, Validation, Testing sets for Brent Crude Oil Price	20
3.3 : Stacked Model Architecture, including LSTM and GRU as first layer and ANN for all others layers	21
3.4 : Activity diagram of Stacked model	28
4.1 : Comparison of Stacked Model , LSTM and GRU predicted value and real value	34



# CHAPTER 1

## INTRODUCTION

### 1.1 Statement of Problems

Crude oil is a key input factor that drives both economic and commercial activities around the world. Therefore, the volatility of oil prices directly affects economic stability in every sector, including academia, the oil industry, and governments. For example, if oil prices rise significantly, the cost of production will increase and the profits of manufacturers will decrease as a result. On the other hand, if oil prices decline, the enthusiasm of oil production companies will be diminished due to the decrease in profits, which could result in a shortage of oil supply in the long run. To emphasize, in late 2019, COVID-19 was discovered and spread around the world, leading to lockdowns in many countries and a drop in oil demand. Oil prices were above \$60 per barrel before falling to \$30 per barrel. Moreover, after the recovery period of two years, there was a war between Russia and Ukraine that began in early 2022, greatly impacting the crude oil price. The highest price of crude oil increased 100% within three months, from around \$62 per barrel in December 2021 to \$123 per barrel in March 2022. Therefore, to prevent such catastrophes from occurring, it is essential to know the future trend of crude oil prices. As mentioned, predicting oil prices requires a combination of many factors to establish an accurate prediction. However, there are many tools available today that make the prediction process easier and more accurate than in the past [1, 4].

A large number of models have been applied to predict crude oil prices [4]. There are roughly two approaches to price prediction: traditional econometric approaches and machine learning methods. First, traditional econometric approaches such as AR (Auto Regressive), MA (Moving Average), ARIMA (Auto-Regressive Integrated Moving Average), and GARCH (Generalized Auto-Regressive Conditional Heteroskedasticity) models were used to forecast the price of many assets. Second, machine learning methods; as knowledge of machine learning has expanded, many scholars are using more deep learning to predict the movement of asset prices, including crude oil prices, due to better performance compared to traditional

econometric approaches. The popular models used by many journals are LSTM (Long Short-Term Memory) and ANN (Artificial Neural Network), and many studies also show that the more models combined, the better the performance of the model [1, 6, 9].

## 1.2 Objectives

This study aims to predict the crude oil price with a multi-stacking model as an approach. Also, this study will attempt to enhance the performance of the model by adjusting the model's architecture by using different input features, a different number of layers, and also different inner models to stack in the overall model.

## 1.3 Research Scope

In order to include recent COVID-19 pandemics into account of this experiment, the study period of the crude oil data will be from January 2016 to December 2021. The data source is from TradingView, where there are Open, High, Low, Close, and Volume contained in the retrieved dataset.

## 1.4 Significant of the study

This research will provide a better result of a price prediction model for crude oil prices. The more accurate the model, the better situation plan the related parties can establish. Furthermore, an enhancement of the model can also be better if the later researchers apply this architecture to theirs with some adjustment.

## CHAPTER 2

# LITERATURE REVIEW

The content in this literature review section covers associated theories and related papers. For this particular paper, we will include data sources, data preparation, modeling, and evaluation, totaling five topics to be presented as follows:

### 2.1 Data

The research will make use of structured data from the TradingView platform via a python library called "tvdatafeed." The retrieved data includes open, close, low, high, volume, and mid, which are manipulated by the researcher using the formula:  $(\text{high} - \text{low}) / 2$  to represent the true range of the price in its particular timeframe. The experiment in this research will be conducted in a daily timeframe between 2016-01-01 and 2021-12-31.

#### 2.1.1 TradingView

TradingView is a charting platform and social network used by 30 million traders and investors worldwide to spot opportunities across global markets. This platform is capable of mostly everything associated with trading and investing, for example, it is possible to visualize charts of crude oil prices along with various indicators such as Moving Average (MA) and Relative Strength Index (RSI). Furthermore, within this platform, there are communities that share their perspectives on product trends or even trading signals, which can be helpful for beginner traders.

### 2.1.2 Candle stick information

A candlestick is a type of price chart used in technical analysis that displays the high, low, open, and closing prices of a security for a specific period. It originated from Japanese rice merchants and traders to track market prices and daily momentum hundreds of years before becoming popularized in the United States. The wide part of the candlestick is called the "real body" and tells investors whether the closing price was higher or lower than the opening price (black/red if the stock closed lower, white/green if the stock closed higher). Candlesticks reflect the impact of investor sentiment on security prices and are used by technical analysts to determine when to enter and exit trades. The visual of the candle is shown in Figure 2.1.

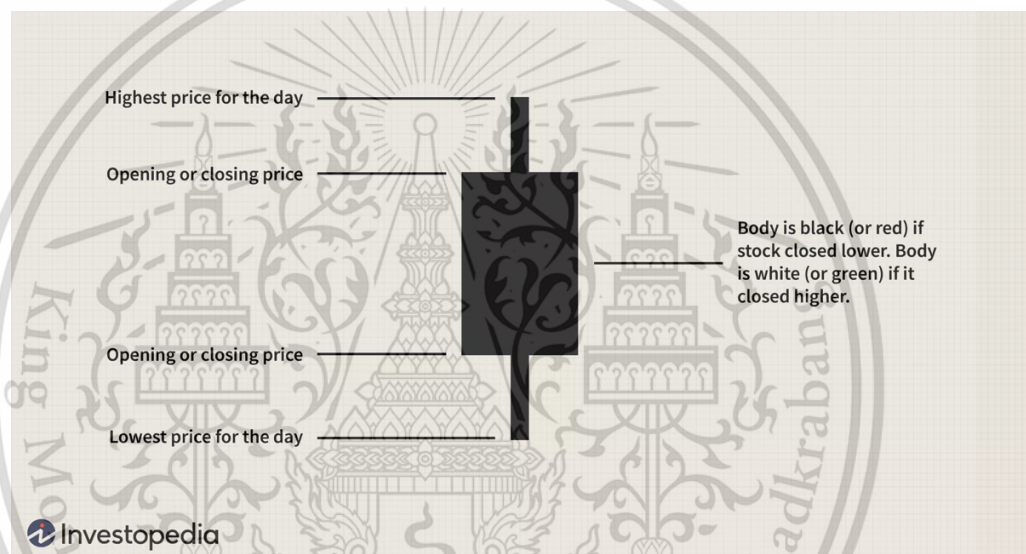


Figure 2.1 : Illustration of Candle stick

Reference : <https://www.investopedia.com/terms/c/candlestick.asp>

### 2.2 Data Preparation

When data is created, it frequently includes missing numbers, inaccuracies, or other problems, and when disparate data sets are merged, they frequently have different formats that need to be reconciled. Large portions of data preparation tasks involve correcting data problems, confirming data quality, and consolidating data sets.

One of the primary purposes of data preparation is to ensure that the raw data being readied for processing and analysis is accurate and consistent so that the results of the prediction will be valid.

### 2.2.1 Z – score (Standard Scaler)

The Z-score is a numerical measurement that describes a value's relationship to the mean of a group of values. Z-score is measured in terms of standard deviations from the mean. If a Z-score is 0, it indicates that the data point's score is identical to the mean score. A Z-score of 1.0 would indicate a value that is one standard deviation from the mean. Z-scores may be positive or negative, with a positive value indicating the score is above the mean and a negative score indicating it is below the mean. The Z score equation is as (2.1):

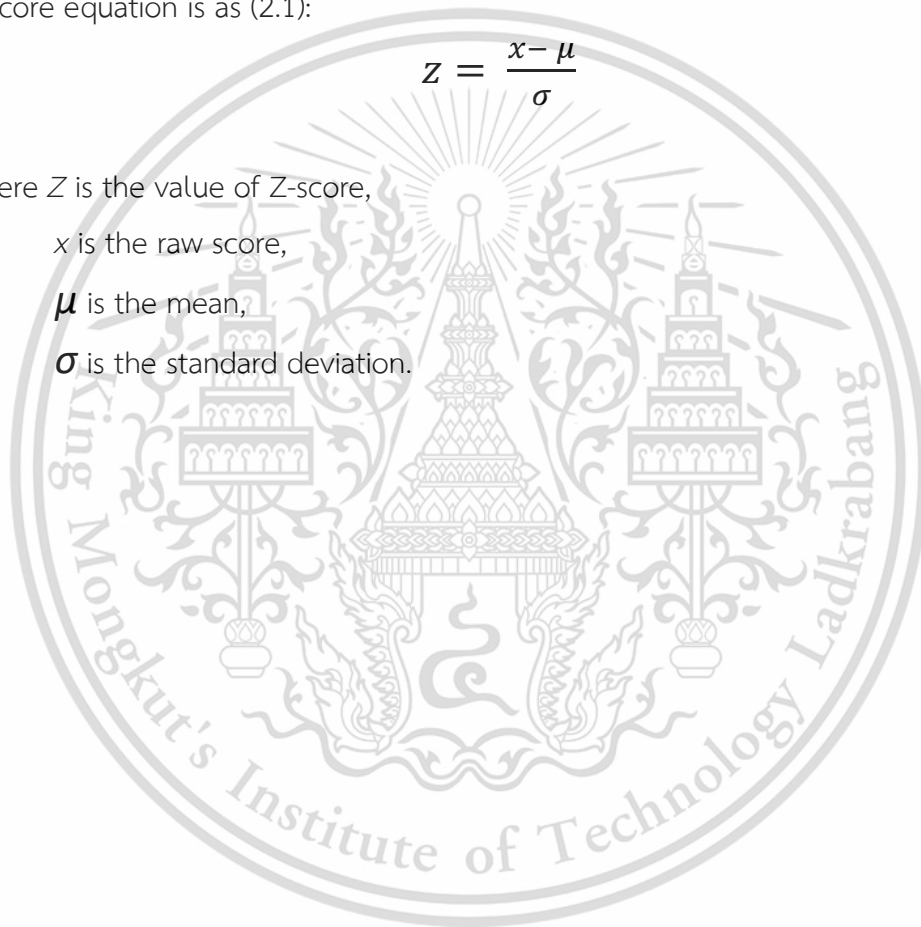
$$Z = \frac{x - \mu}{\sigma} \quad (2.1)$$

where Z is the value of Z-score,

x is the raw score,

$\mu$  is the mean,

$\sigma$  is the standard deviation.



## 2.3 Modelling

### 2.3.1 LSTM (Long short-term memory)

LSTM is the enhance version of RNN proposed by Schmidhuber et al. in 1997 [5]. LSTM is a neural network model that solve RNN's gradient diminishing and gradient exploding problems the model perform back propagation [7]. LSTM's main components are 3 gates that illustrated in Figure 2.2 (Forget gate, Input gate, Output gate). Forget gate helps the LSTM to decide what information to discard from the previous time step's cell state. It takes in the previous cell state and the current input, and outputs a number between 0 and 1 for each element of the cell state. A value of 0 means "completely forget this information", while a value of 1 means "completely remember this information". Input gates helps the LSTM to decide what new information to store in the cell state. It takes in the previous cell state and the current input, and uses a sigmoid activation function to determine which parts of the input to update. It also uses a tanh activation function to create a vector of new candidate values that could be added to the cell state. Output gate helps the LSTM to decide what information to output from the current time step's cell state. It takes in the current input and the updated cell state, and uses a sigmoid activation function to determine which parts of the cell state to output. It then applies a tanh activation function to the selected parts of the cell state, which produces the output for the current time step.

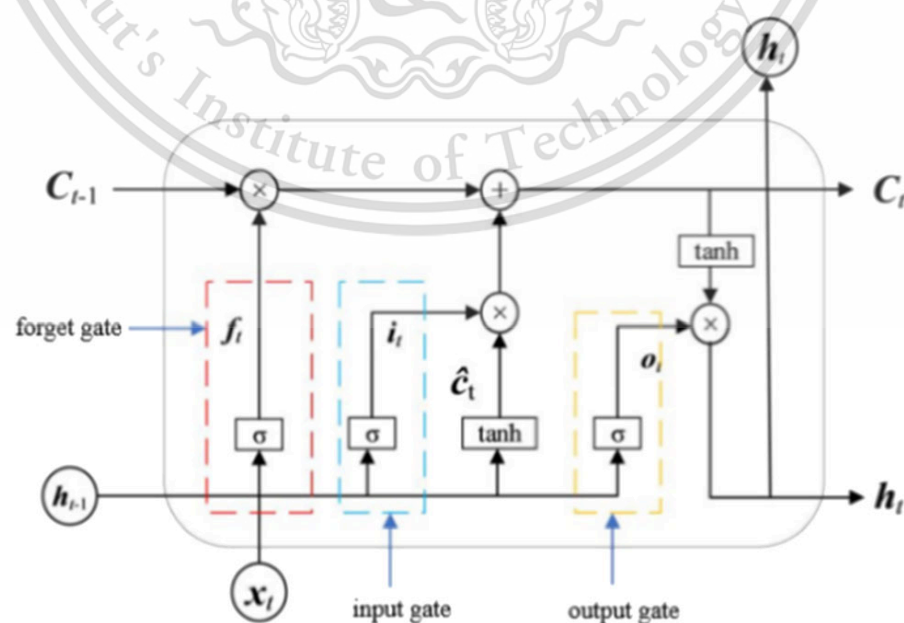


Figure 2.2 : LSTM's node architecture

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

where  $C_{t-1}$  is the cell state from the previous timestep,

$h_{t-1}$  is the output from the previous timestep,

$X_t$  is the input for current timestep,

$\sigma$  is sigmoid function,

$f_t$  is the output of forget gate,

$i_t$  is the output of input gate,

$\hat{C}_t$  is the candidate cell state,

$\tanh$  is an activation function,

$o_t$  is the output of output gate,

$C_t$  is the output cell state of current timestep,

$h_t$  is the output of the current timestep.

The flow of how the inputs is pass into each node are

- (1) The output from the previous timestep and the input of the current timestep are combined and parsed into forget gate. The output of the forget gate is as depicted in equation (2.2):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.2)$$

where  $f_t$  is the result of sigmoid function which range between 0 and 1,

$W_f$  is the weight of the forget gate,

$b_f$  is the bias of the forget gates,

$x_t$  is the input value of current timestep,

$h_{t-1}$  is the output from the previous timestep.

- (2) The output from the previous timestep and the input of the current timestep are combined and parsed into input gate. The output of input gate is the multiplication of  $i_t$  and  $\hat{C}_t$  as equation (2.3), (2.4):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

$$\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.4)$$

where  $i_t$  is the result of sigmoid function which range between 0 and 1,

$W_i$  is the weight of the input gate,

$b_i$  is the bias of the input gates,

$W_c$  is the weight of the candidate input gate,

$b_c$  is the bias of the candidate input gates.

- (3) The current cell state is updated as depicted in equation (2.5):

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (2.5)$$

where  $C_t$  is the result which range between 0 and 1.

- (4) The output from the previous timestep and the input of the current timestep are combined and parsed into output gate. The output of output gate is as equation (2.6):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.6)$$

where  $o_t$  is the result of sigmoid function which range between 0 and 1,

$W_o$  is the weight of the output gate,

$b_o$  is the bias of the output gates.

- (5) The output of LSTM is from the multiplication of output from output gate and the tanh of Cell state as depicted equation (2.7):

$$h_t = o_t * \tanh (C_t) \quad (2.7)$$

where  $h_t$  is the output of the current timestep.

$o_t$  is the result of sigmoid function which range between 0 and 1,

$\tanh$  is an activation function,

$C_t$  is the output cell state of current timestep



### 2.3.2 GRU (Gated Recurrent Unit network)

Gated recurrent unit is a modified LSTM model with two gates. A reset gate and an update gate are used to form the hidden state. The reset gate determines the how to combine the current input in the current state with the memory from previous state. Update gate is responsible for deciding how the memory from previous state should be maintained in the node. [8]

GRU's main components, Reset gate and Update gate, is shown in Figure 2.3

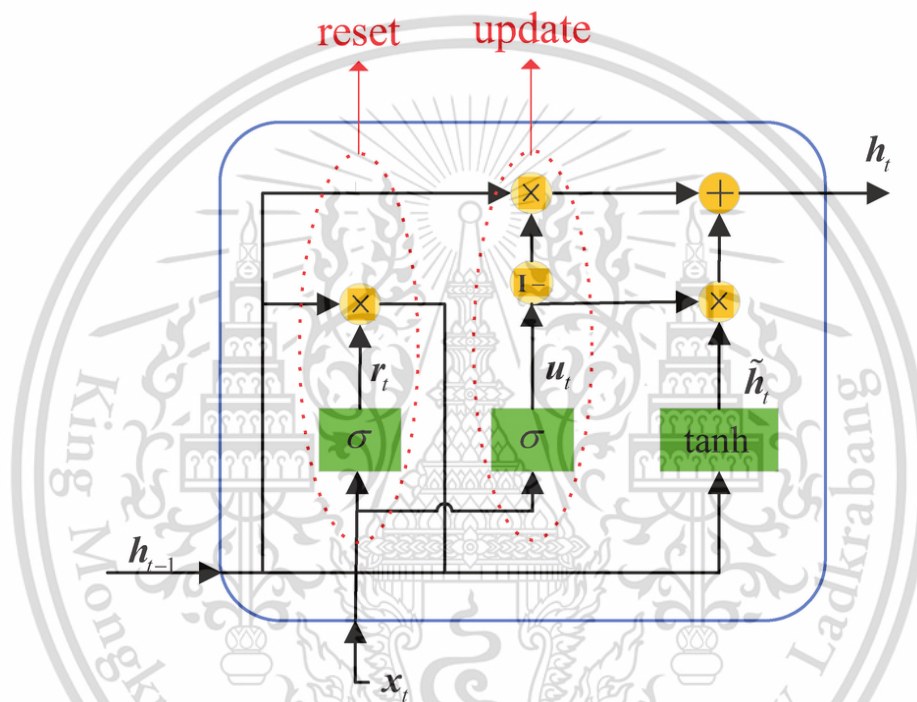


Figure 2.3 : GRU's node architecture

where  $h_{t-1}$  is the output from the previous timestep,

$x_t$  is the input for current timestep,

$\sigma$  is sigmoid function,

$r_t$  is the output from reset gate,

$u_t$  is the output from update gate,

$\tilde{h}_t$  is the candidate cell state,

$\tanh$  is an activation function,

$\hat{h}$  is a candidate activation function,

$h_t$  is the output of the current timestep.

The flow of how the inputs is pass into each node are

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

- (1) The output from the previous timestep and the input of the current timestep are combined and parsed into reset gate. The output of the reset gate is as depicted in equation (2.8):

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (2.8)$$

where  $r_t$  is the output of reset gate,

$W_r$  is the weight of reset gate,

$b_r$  is the bias of reset gate,

$x_t$  is the input value of current timestep.

- (2) The output from the previous timestep and the input of the current timestep are combined and parsed into update gate. The output of the update gate is as depicted in equation (2.9):

$$u_t = \sigma(W_u \cdot [h_{t-1}, x_t] + b_u) \quad (2.9)$$

where  $u_t$  is the output of update gate,

$W_u$  is the weight of update gate,

$b_u$  is the bias of update gate.

- (3) The multiplication of reset gate output and the output from previous timestep combine with the input of the current timestep, are parsed into tanh activation function. The output, candidate activation function, will be as equation (2.10):

$$\hat{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h) \quad (2.10)$$

where  $W_h$  is the weight of candidate hidden state,

$b_h$  is the bias of candidate hidden state.

- (4) The output of GRU is as depicted equation (2.11):

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \quad (2.11)$$

### 2.3.3 ANN (Artificial Neural Network)

ANN is nonlinear model (Figure 2.4) by the help of various activation functions such as sigmoid, Relu (Rectified linear unit). The model methodology is imitated from human brain which consist plenty connected nodes all together. As the model works the same as human brain, it can distinguish some patterns normally hard to be found out by normal data mining method. The precedence of ANN is that the model is data-driven which means the more data fed into the model the better output it likely to be.

In an ANN model, it consists of 3 identified layers called input layer, hidden layer and output layer. The Input layer gets information from the outside in order to process in the next layer (hidden layer). The hidden layer, the layer between input layer and output layer, contain node(s) which each node consists of activation function or complex function inside. Moreover, the hidden layer can have more than 1 layer depending on the model architecture of the implementer. The output layer is the result after all the process from hidden layer. [10]

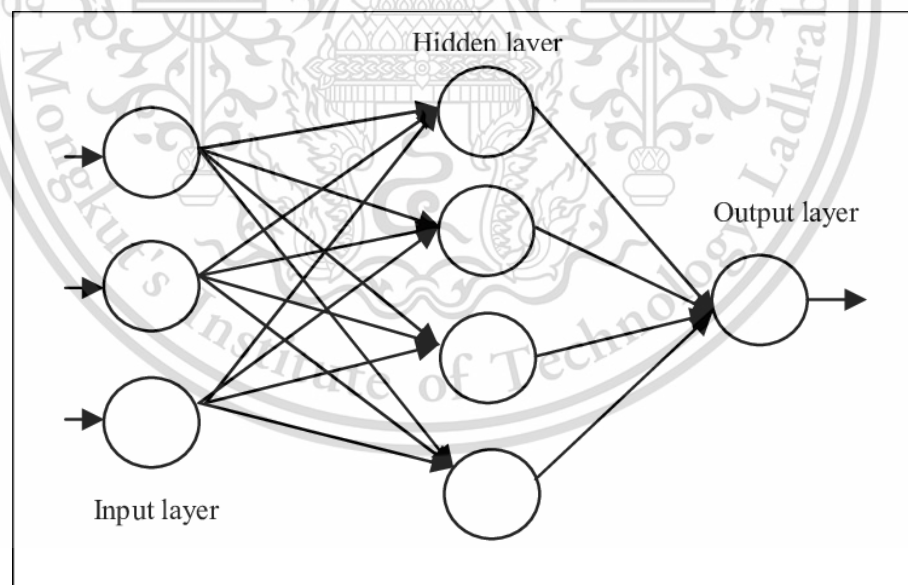


Figure 2.4 : ANN's Architecture

### 2.3.3.1 Sigmoid function

Sigmoid function is the mathematical function that takes any real number as an input and the output will be between 0 and 1. This function is also known as logistic function. The equation is as equation (2.12) :

$$y = \frac{1}{1+e^{-z}} \quad (2.12)$$

where  $y$  is the output of sigmoid function which is range between 0 and 1,  
 $e$  is the base of natural logarithm ( $\approx 2.71$ ),  
 $z$  is the product of the input node and weight for the node plus the bias.

### 2.3.3.2 ReLU(The rectified linear activation unit) function

ReLU is the mathematical function that takes any real number as an input and the output will be between 0 and input. This method is popular in ANN's node due to its low requirement for computation resource which makes it faster than using any other activation function. This function solves vanishing gradient problem while suffer from exploding gradient sometimes when the model does back propagation to train the model. equation is as (2.13)

$$f(x) = \max(0, x) \quad (2.13)$$

where  $f(x)$  is the output of sigmoid function which is range between 0 and input value  
 $x$  is the input value.

## 2.4 Evaluation

Evaluation plays the most crucial role in this research since the result relies on how we evaluate at it at the end of the experiment. For this case, Crude oil price prediction, we need to compare several models whether it is good or not. Thus, to compare the forecasting accuracy and stability, in this study, we mainly chose the mean square error (MSE), Root mean square error (RMSE) and standard deviation of absolute percentage error (SDAPE) to effectively evaluate stacked model proposed.

For MSE and RMSE, the lower the MSE or RMSE means the better forecasting performance. For SDAPE, the smaller the SDAPE value, the higher the prediction stability. In addition, the value of SDAPE is between 0 and 1.

These mentioned indicators' formulas are as follow:

$$MSE = \frac{1}{n} (\sum_{i=1}^n (y_i(t) - \hat{y}_i(t))^2) \quad (2.14)$$

$$RMSE = \sqrt{\frac{1}{n} (\sum_{i=1}^n (y_i(t) - \hat{y}_i(t))^2)} \quad (2.15)$$

$$SDAPE = \sqrt{\frac{1}{n} \sum_{i=1}^n (|\frac{\hat{y}_i(t) - y_i(t)}{y_i(t)}| - MAPE)^2} \quad (2.16)$$

where  $n$  is the number of samples in the training set,

$\hat{y}$  is the predictive value,

$y$  is the true value,

MAPE (Mean Absolute Percent Errors) is a component used to calculate the SDAPE.

$$\text{Remark : } MAPE = \frac{1}{n} * \sum_{i=1}^n |\frac{\hat{y}_i(t) - y_i(t)}{y_i(t)}| * 100\%$$

## 2.5 Related Paper

From literature [1], the authors propose a hybrid ARIMA-NARNN model for predicting the global price of bananas. The model combines the linear equation provided by the ARIMA model with the nonlinear equation provided by the NARNN model. The ARIMA model is used to predict the initial price, while the NARNN model is used to predict the residual error from the ARIMA model. The authors assume that the residual error will have only a nonlinear relationship, and that using the NARNN model will result in residual errors that are white noise. The mean square errors of the three proposed models (ARIMA, NARNN, and the hybrid ARIMA-NARNN model) are 4,475.51, 3,058.11, and 2,982.67, respectively.

Overall, the idea behind the hybrid model is well-explained and makes sense. However, it can be seen that using the hybrid model only improves the model performance by less than 3% compared to using the NARNN model alone. This suggests that the NARNN model is able to capture more than 97% of the price information without the need for a linear model. In addition, the authors only use price as a feature, which means that the model could be improved by adding additional features.

From literature [10], a method is proposed to forecast the crude oil (WTI, Brent) price using the LSTM (Long Short Term Memory) model. The experiment was conducted with 3 timeframes: short term using daily timeframe data, medium term using monthly timeframe data, and long term using yearly timeframe data. In each timeframe, ARIMA, ANN, and LSTM were implemented. The results showed that ARIMA performed much worse than the other two models, and ANN performed better than LSTM in the medium term. However, LSTM performed better in the short term and long term. The MSE of WTI price in the long term for ARIMA, ANN, and LSTM were 4,623.76, 0.24, and 0.12, respectively. This result shows that deep learning models can outperform statistical models in time series data.

One limitation of this study is that it only used price as a feature, which means that the model could be improved by including additional factors. Additionally, the author noted that the longer the timeframe, the less accurate the model becomes, so

incorporating more factors into the model could improve its accuracy. Overall, while the LSTM model performed well in the short and long term, it is still subjective to conclude that it is the better model in this particular experiment.

From literature [6], the proposed method for predicting the Shanghai Composite Index price uses a CNN-BiLSTM-AM model. This model consists of convolutional neural networks (CNN), bi-directional long short-term memory (BiLSTM), and an attention mechanism (AM). The features used in this experiment are the opening price, highest price, lowest price, closing price, volume, turnover ups and downs, and change. According to the results, the more complex the model, the better its performance. The CNN-BiLSTM-AM model performs the best with a mean absolute error (MAE) of 21.95, compared to other models such as MLP, CNN, RNN, LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM, and BiLSTM-AM. The MLP model performed the worst with an MAE of 31.49.

To enhance the model's accuracy for price prediction, the use of newer models with complex architectures is crucial. The Attention Mechanism model, discovered by Google in 2017, is key in distinguishing this model from others.

From literature [9], a method of multi-level stacking of LSTM recurrent models was proposed for predicting stock-market indices. The main idea is to use three consecutive layers. The first consists of  $M$  LSTM units, each receiving successive values related to SMI closing prices of  $N$  previous days, including the current day, to predict tomorrow's SMI closing price. To be precise, each LSTM unit contains only two features: the closing price and another feature, which helps to reduce variance since the features are spread evenly across each LSTM unit instead of being packed into one unit. The second layer contains  $S$  MLP meta-models, each of which collects the outputs of the LSTM base models. The third layer contains a single MLP meta-model that combines the outputs of the meta-models. According to the results, the deeper the layer in the architecture, the better the performance (lower MSE, MAE). Moreover, compared to other models such as Moving Average, Exponential Moving Average, Support Vector Machine, and LSTM, the multi-level stacking LSTM performs the best.

To discuss, this model architecture is easy to understand and can be easily enhanced by adding more features or layers. However, the obvious trade-off of this kind of model is the consumption of computation power because the number of sub-models to be trained scales up depending on the number of features and layers added.

From the literature [2 - 4], various methods have been proposed to predict the price of crude oil. Using ANN with different lag periods (2, 3, 4, 5 timesteps) has been found to be effective, with a lag of 3 performing better than the others. However, the author notes that financial markets are generally unpredictable, and while ANN is an effective tool, it is not a guaranteed solution [4]. Another method is the stream learning model, which uses previous timesteps' prices to predict the next price and updates every timestep as the window slides. According to the author, the stream learning method outperforms heuristic, econometric, and ANN methods [3]. Lastly, the ensemble model combines the output of three other models and averages them to produce the ensemble method's output [2].

Although these proposed ideas are effective, they only use price as a feature, which may limit performance. Therefore, there is room for improvement by constructing models that incorporate additional features. The research's last finding supports the notion that using modern models or combining models can significantly enhance performance.

After researching the mentioned literature, it appears that the proposed model architecture that improves model performance lies in the complexity of the model architecture. Furthermore, having more relevant features also helps to improve the accuracy of the results, according to literature [6] and literature [9].

To conclude, I am excited by the review I have done, which has led me to the idea of using multiple features, not just the closing price, as input for a model. My model architecture will be a stacked model that includes LSTM and GRU as the first layer. The output of the first layer will be used as features in the second layer of multiple ANNs. This layer will provide insight into the optimizers that will be varied in this layer. The third layer will be similar to the previous layer but instead of varying

This material is reserved for educational use only, not allowed for commercial use.

the optimizer, we will differ the number of nodes of each model in this layer. The fourth layer will be the final layer to retrieve the output from the model, which will illustrate whether the stacked model improves the overall performance.



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

## CHAPTER 3

### RESEARCH METHODOLOGY

This section explains the experimental methods used in this study, from data preparation to model evaluation, providing the necessary information for other researchers to replicate, modify, and enhance the study. The workflow is shown as Figure 3.1 .

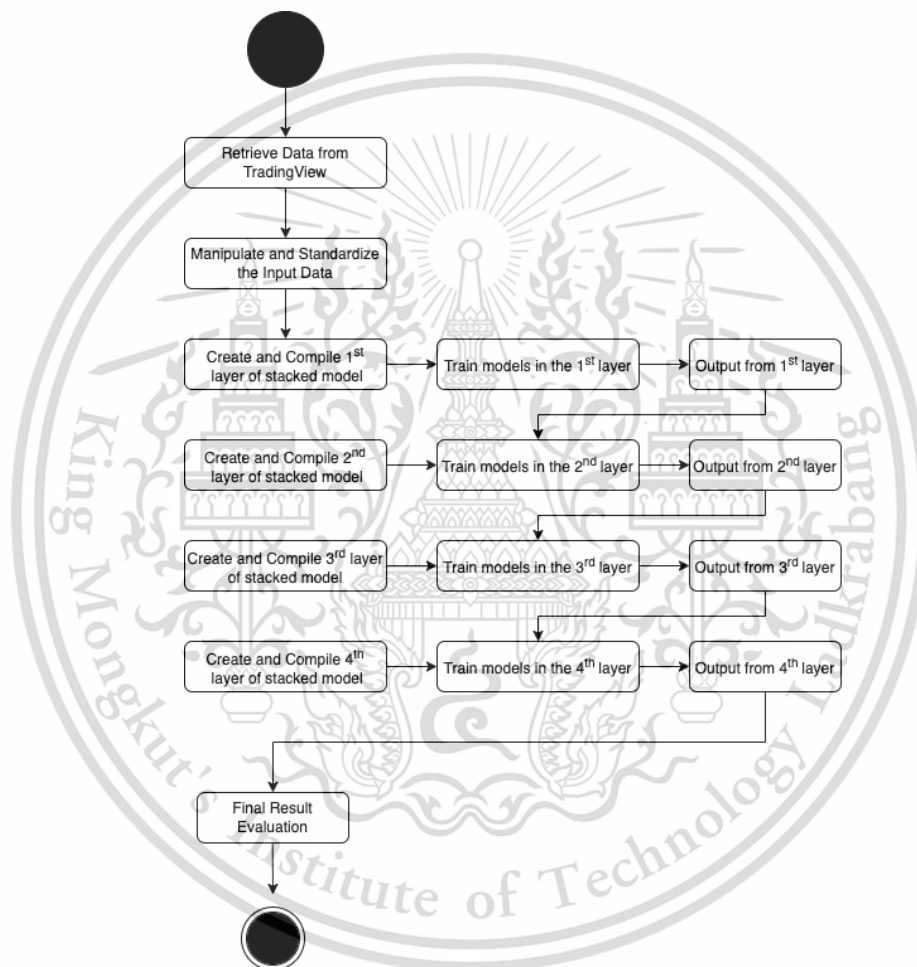


Figure 3.1 : Experiment Workflow

#### 3.1 Dataset Preparation

Datasets are critical in data analytics, especially for prediction and forecasting, and it is important to retrieve data from reliable sources. This study uses data from TradingView, a top website for investing. The data is fetched using the open-source API "tvDatafeed," which allows us to obtain the open, high, low, close (OHLC) values and traded volume (Volume) as initial features. In addition, the mid-price is calculated as

This material is reserved for educational use only, not allowed for commercial use.

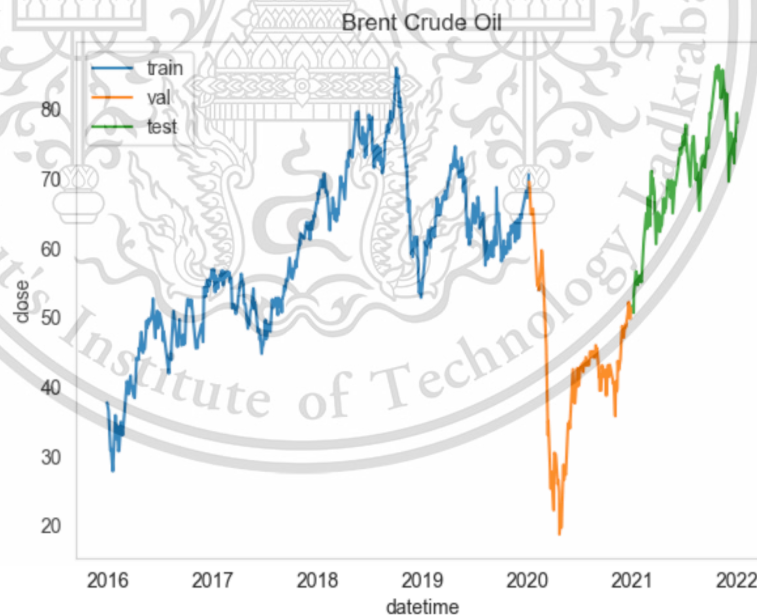
Forbidden to modify the content, and cite the document when use.

the average of the high and low prices as an extra feature to represent the true range of the price in its particular timeframe. Some of the data are shown in Table 3.1.

**Table 3.1** : Partial sample data

<i>date</i>	<i>open</i>	<i>high</i>	<i>low</i>	<i>close</i>	<i>volume</i>	<i>mid</i>
2021-12-21	72.114	74.482	71.452	74.311	65713.0	72.967
2021-12-22	74.247	75.857	73.826	75.767	61683.0	74.841
2021-12-23	75.652	75.968	74.965	75.296	11895.0	75.466
2021-12-29	78.604	80.216	78.291	79.436	35547.0	79.253
2021-12-30	79.215	80.200	78.594	79.200	40840.0	79.397

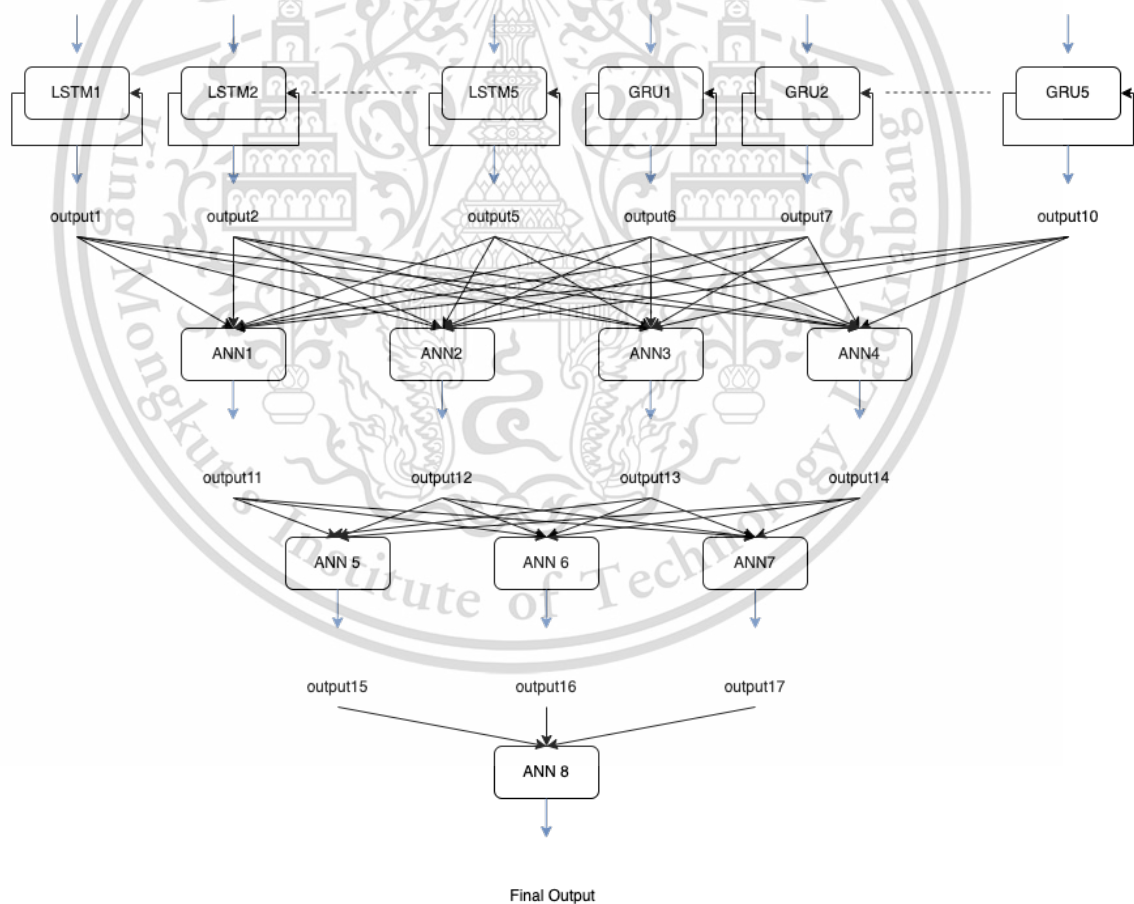
This study will be conducted using daily Brent Crude Oil prices from January 4, 2016, to December 31, 2021. The data is divided into a training set (60%), a validation set (20%), and a testing set (20%) chronologically, as shown in Figure 3.2.



**Figure 3.2** : Training, Validation, Testing sets for Brent Crude Oil Price

### 3.2 Stacking Model Architecture

The key components of a stacking model are the ensemble model and the stacking model. The model used in this study has four layers: an LSTM and GRU layer, a first ANN layer, a second ANN layer, and a third ANN layer. The first layer includes ten models, five LSTM and five GRU, each of which uses two features (close and one other feature, such as open, high, low, volume, mid) and has a lookback step of 5. The first ANN layer, which retrieves 10 inputs from the LSTM-GRU models, contains four different configurations of ANN models. The second ANN layer, which retrieves 4 inputs from the first ANN layer, contains three different configurations of ANN. The third ANN layer, which retrieves 3 inputs from the second ANN layer, outputs the final result. The entire model architecture is shown in Figure 3.3. The parameter settings for each model in the stacked model used in this experiment are shown in Table 3.2 – 3.19.



**Figure 3.3 :** Stacked Model Architecture, including LSTM and GRU as first layer and ANN for all others layers

There is a reason for each layer in this experiment. The first layer of the stacked model, which consists of LSTM and GRU models, experiments whether LSTM or GRU is better and also which features input into the model perform the best. Both GRU and LSTM are types of recurrent neural networks (RNNs) that are well-suited for processing sequential data. However, they possess unique strengths and limitations. Longer sequences are better handled by LSTM, whereas smaller datasets are better served by GRU, which is more effective and less prone to overfitting. By combining their individual advantages, we might be able to stack them and increase performance. Each model consists of the combination of closing price and another feature as follows [close, open], [close, high], [close, low], [close, volume] and [close, mid]. The rationale behind why the close price is often considered as the most important feature in financial time series analysis, as it represents the price of the asset at the end of the trading day, which is what most traders and investors base their decisions on. Therefore, it is crucial to use the close price as a main feature in model prediction. Combining the close price with other features such as open, high, low, volume, or mid can provide additional information that can help improve the accuracy of the prediction.

The second layer of the stacked model shows how effectively the initial output from the first layer is used and focuses on the configuration of the models, particularly the optimizers Adam, Adadelta, and Rmsprop because these 3 are commonly used optimizers in deep learning. Therefore, they were chosen as the optimizers for the experiment to evaluate their performance in predicting stock prices using deep learning models.

The third layer of the stacked model examines whether the number of nodes in the hidden layer affects the model's performance, assuming that a greater number of nodes leads to better performance. Thus, the 3 models in this layer are distinguished by the number of nodes in the hidden layers, which are 6, 16, and 32 nodes in ANN5, ANN6, and ANN7 model respectively.

The final layer of the stacked model examines whether a deeper layer leads to better performance. This layer provides the conclusion on whether a deeper level

This material is reserved for educational use only, not allowed for commercial use.

leads to better performance in terms of RMSE, MSE, and SDAPE when comparing with the output of each sub-model contained in this stacked model.

**Table 3.2 :** Setting of Parameters in LSTM - 1

Parameters	Value
Features	Close , Open
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.3 :** Setting of Parameters in LSTM - 2

Parameters	Value
Features	Close , High
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.4 :** Setting of Parameters in LSTM - 3

Parameters	Value
Features	Close , Low
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.5 :** Setting of Parameters in LSTM - 4

Parameters	Value
Features	Close , Volume
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

**Table 3.6** : Setting of Parameters in LSTM - 5

Parameters	Value
Features	Close , Mid
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.7** : Setting of Parameters in GRU - 1

Parameters	Value
Features	Close , Open
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.8** : Setting of Parameters in GRU - 2

Parameters	Value
Features	Close , High
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.9** : Setting of Parameters in GRU -3

Parameters	Value
Features	Close , Low
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.10** : Setting of Parameters in GRU - 4

Parameters	Value
Features	Close , Volume
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.11** : Setting of Parameters in GRU - 5

Parameters	Value
Features	Close , Mid
Hidden Layer	1
Number of nodes in hidden layer	128
Optimizer	adam

**Table 3.12** : Setting of Parameters in ANN - 1

Parameters	Value
Features	10 outputs from LSTM and GRU
Hidden Layer	1
Number of nodes in hidden layer	9
Optimizer	adam

**Table 3.13** : Setting of Parameters in ANN - 2

Parameters	Value
Features	10 outputs from LSTM and GRU
Hidden Layer	1
Number of nodes in hidden layer	10
Optimizer	adam

**Table 3.14** : Setting of Parameters in ANN - 3

Parameters	Value
Features	10 outputs from LSTM and GRU
Hidden Layer	1
Number of nodes in hidden layer	10
Optimizer	adadelata

**Table 3.15** : Setting of Parameters in ANN - 4

Parameters	Value
Features	10 outputs from LSTM and GRU
Hidden Layer	1
Number of nodes in hidden layer	10
Optimizer	RMSprop

**Table 3.16** : Setting of Parameters in ANN - 5

Parameters	Value
Features	4 outputs from Layer 2's ANN
Hidden Layer	1
Number of nodes in hidden layer	6
Optimizer	adam

**Table 3.17** : Setting of Parameters in ANN - 6

Parameters	Value
Features	4 outputs from Layer 2's ANN
Hidden Layer	1
Number of nodes in hidden layer	16
Optimizer	adam

**Table 3.18** : Setting of Parameters in ANN - 7

Parameters	Value
Features	4 outputs from Layer 2's ANN
Hidden Layer	1
Number of nodes in hidden layer	32
Optimizer	adam

**Table 3.19** : Setting of Parameters in ANN - 8

Parameters	Value
Features	3 outputs from Layer 3's ANN
Hidden Layer	1
Number of nodes in hidden layer	32
Optimizer	adam

### 3.3 Experiments

In order to substantiate the effectiveness of stacked model, this experiment will compare base LSTM, base GRU model, and its own sub model within the stacked model using the same training, validating, testing data set to depict the holistic performance. In our study, the process is shown in Figure 3.4.

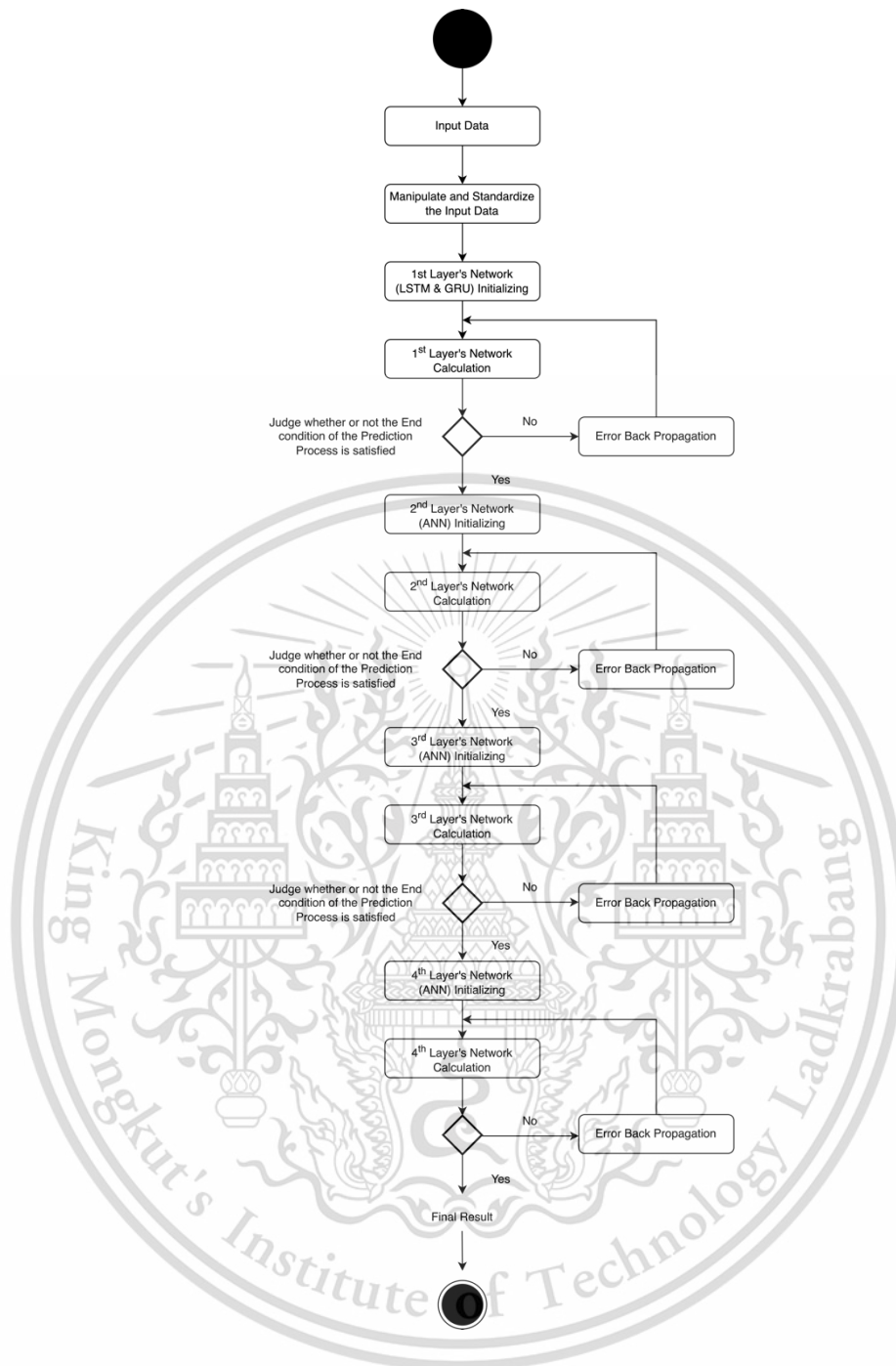


Figure 3.4 : Activity diagram of Stacked model

The main steps are as follows :

- (1) Input Data : Retrieve Brent Crude Oil Data from TvDatafeed.
- (2) Manipulate and standardize the Input Data : Generating the mid-price by average the low price and high price then standardize all the features with Z – score.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

- (3) 1<sup>st</sup> Layer's Network Initializing : Generate LSTM and GRU models, each containing close price and another features.
- (4) 1<sup>st</sup> Layer's Network Calculation : Feed the Input data into the initialized model and let the model tuning through back propagation method until the error is satisfied the condition then the output is obtained in each model.
- (5) 2<sup>nd</sup> Layer's Network Initializing : Generate ANN models, each containing features from the outputs from the 1<sup>st</sup> layer, 10 features in total
- (6) 2<sup>nd</sup> Layer's Network Calculation : Feed the Input data into the initialized model and let the model tuning through back propagation method until the error is satisfied the condition then the output is obtained in each model.
- (7) 3<sup>rd</sup> Layer's Network Initializing : Generate ANN models, each containing features from the outputs from the 2<sup>nd</sup> layer, 4 features in total
- (8) 3<sup>rd</sup> Layer's Network Calculation : Feed the Input data into the initialized model and let the model tuning through back propagation method until the error is satisfied the condition then the output is obtained in each model.
- (9) 4<sup>th</sup> Layer's Network Initializing : Generate ANN models, each containing features from the outputs from the 3<sup>rd</sup> layer, 3 features in total
- (10) 4<sup>th</sup> Layer's Network Calculation : Feed the Input data into the initialized model and let the model tuning through back propagation method until the error is satisfied the condition then the final output is obtained.

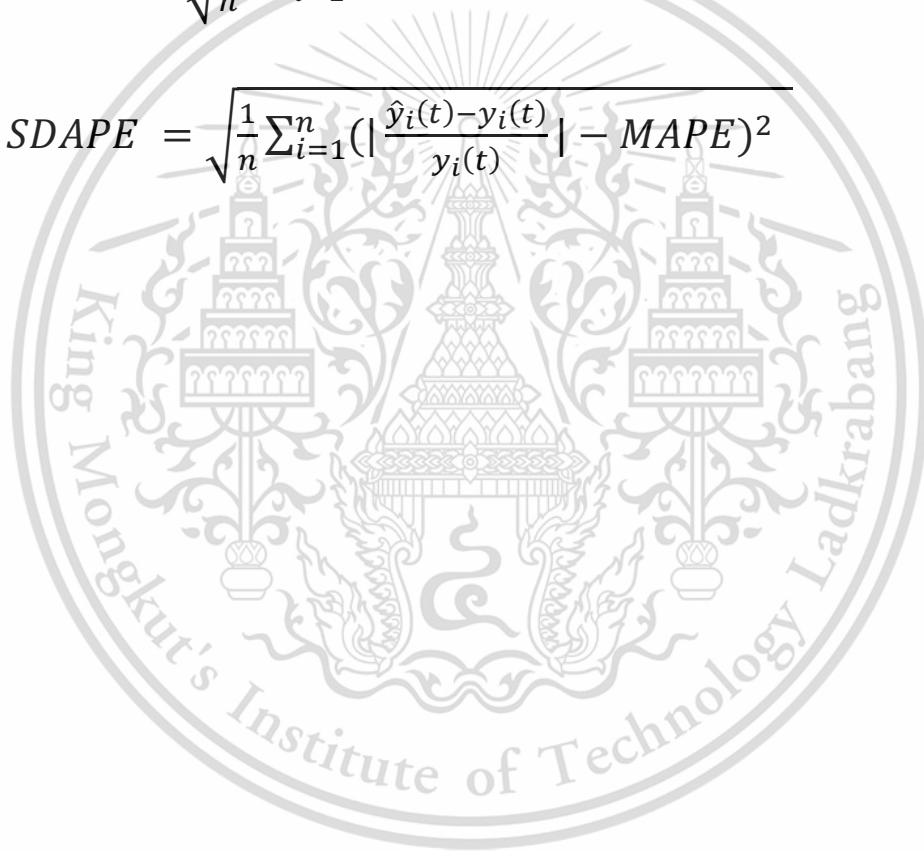
### 3.4 Evaluation

The performance of a model for forecasting can be evaluated by two metrics: MSE (3.1) and RMSE (3.2). A lower MSE or RMSE indicates better performance. Another metric, SDAPE (3.3), measures the stability of predictions. A smaller SDAPE value indicates higher prediction stability. The value of SDAPE is between 0 and 1.

$$MSE = \frac{1}{n} (\sum_{i=1}^n (y_i(t) - \hat{y}_i(t))^2) \quad (3.1)$$

$$RMSE = \sqrt{\frac{1}{n} (\sum_{i=1}^n (y_i(t) - \hat{y}_i(t))^2)} \quad (3.2)$$

$$SDAPE = \sqrt{\frac{1}{n} \sum_{i=1}^n (|\frac{\hat{y}_i(t) - y_i(t)}{y_i(t)}| - MAPE)^2} \quad (3.3)$$



## CHAPTER 4

# RESULT AND DISCUSSION

In this chapter, we will illustrate the results of our model evaluation using three evaluation metrics: mean squared error (MSE), root mean squared error (RMSE), and standard deviation mean absolute percentage error (SDAPE). We also discuss the performance of each layer of the model to gain insights into its strengths and limitations.

Table 4.1 illustrates the prediction results for all the models within the stacked model. Column 1 in Table 4.1 presents the models that were used to predict the next day's closing price. Columns 2, 3, and 4 represent the MSE, RMSE, and SDAPE for the prediction results, respectively.

According to Table 4.1, in the first layer that consists of LSTM and GRU models, LSTM-1 and GRU-1, which contain the close price and open price as features, perform better than others in the same layer in every metric. However, although LSTM-1 performs the best in the first layer of the stacked model, the other combinations of features besides open and volume, GRU performs better than LSTM in every metric. Additionally, the GRU-4 model with close and volume as features seems to predict worse than the others because the combination of features might not be good or it might be because we did not adjust the learning rate to make this combination converge to better performance with smaller error.

The second layer of the stacked model depicts how well the initial output from the first layer is used, and the configuration of the models focuses on the optimizer, Adam, Adadelata, and Rmsprop. According to Table 4.1, ANN-1, using the Adam optimizer, performs the best among other optimizers.

The third layer of the stacked model examines whether the number of nodes in the hidden layer affects the model's performance. The results show that ANN-6 performs the best among others in the same layer, given that ANN-5 contains 6 nodes, ANN-6 contains 16 nodes, and ANN-7 contains 32 nodes. The outcome can clearly

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

indicate that a greater number of nodes does not necessarily lead to better performance, since ANN-7 did not perform the best compared to the model with a lower number of nodes.

**Table 4.1** : Prediction performance of individual model in stacked model for Brent Crude Oil price prediction

MODEL	MSE (MEAN SQUARED ERROR)	RMSE (ROOT MEAN SQUARED ERROR)	SDAPE (STANDARD DEVIATION MEAN ABSOLUTE PERCENTAGE ERROR)
L1: LSTM-1	4.2507	2.0617	0.0427
L1: LSTM-2	5.3826	2.32	0.0498
L1: LSTM-3	5.9008	2.4292	0.0549
L1: LSTM-4	5.0847	2.2549	0.0479
L1: LSTM-5	5.5078	2.3469	0.0521
L1: GRU-1	4.2518	2.062	0.0421
L1: GRU-2	4.6741	2.162	0.0437
L1: GRU-3	4.6439	2.155	0.045
L1: GRU-4	19.5877	4.4258	0.0978
L1: GRU-5	4.8738	2.2077	0.0452
L2 – ANN-1	5.0562	2.2486	0.0498
L2 – ANN-2	7.9449	2.8187	0.0648
L2 – ANN-3	8.9916	2.9986	0.0708
L2 – ANN-4	8.718	2.9526	0.0696
L3 – ANN-5	5.3051	2.3033	0.0495
L3 – ANN-6	3.663	1.9139	0.0326
L3 – ANN-7	3.7217	1.9292	0.0347
L4 – ANN-8	4.3366	2.0825	0.0294

The final layer of the stacked model examines whether a deeper layer leads to better performance. The results show that MSE and RMSE are worse than ANN-6. This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

and ANN-7 from the previous layer. However, SDAPE is lower by 9.2% (0.0326 to 0.0294) compared to the previous best SDAPE in the stacked model. This can be inferred that, to minimize the error, building more stack layers is not the solution, while to make the model more stable with a lower standard deviation, building more stack layers is a helpful method to implement.

From Table 4.2, the comparison of prediction performance between LSTM, GRU, and the stacked model shows that the stacked model performs the best in every metric, followed by LSTM and GRU, respectively. The MSE of the LSTM and GRU is 7.95% and 30.16% higher than the MSE of stacked model, respectively. The RMSE of the LSTM and GRU is 3.89% and 14.08% higher than the RMSE of stacked model, respectively. The SDAPE of the LSTM and GRU is 38.43% and 73.81% higher than the SDAPE of stacked model, respectively. Therefore, out of the three models, the stacked model method proposed by this paper predicts the closing price of Brent crude oil's daily closing price of the next day the most accurately, which will provide a good reference for anyone who wants to make investment decisions or make any deals related to oil prices.

**Table 4.2 :** Prediction performance of based LSTM model, based GRU model and Stacked model

MODEL	MSE	RMSE	SDAPE
LSTM	4.6815 (+7.95%)	2.1636 (+3.89%)	0.0407 (+38.43%)
GRU	5.6448 (+30.16%)	2.3759 (+14.08%)	0.0511 (+73.81%)
STACKED MODEL	<b>4.3366 (&lt;0.00%)</b>	<b>2.0825 (&lt;0.00%)</b>	<b>0.0294 (&lt;0.00%)</b>

As illustrated in figure 4.1, among the models used in this experiment perform good with bare eye. However, on November 2021, when the close price spiked up to around 85, stacked model predicted the closest accurate compared to other 2 models significantly. This occurrence can infer that stacked model perform better when the price volatile.

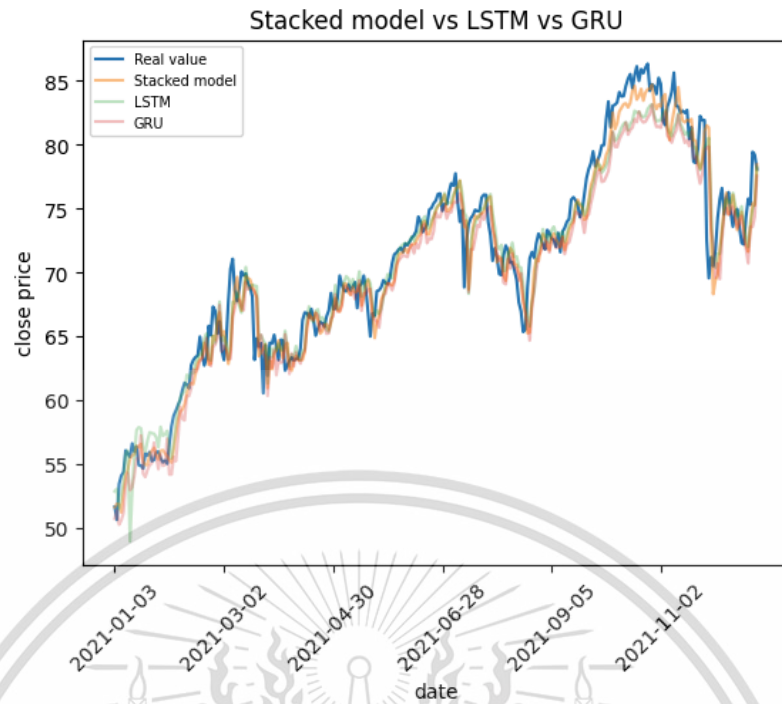


Figure 4.1 : Comparison of Stacked Model , LSTM and GRU predicted value and real value

## CHAPTER 5

# CONCLUSION

Brent crude oil price forecasting is important for understanding and predicting the direction of the global economy. As a widely-used benchmark for oil prices, the price of Brent crude oil is often seen as a good indicator of the overall health of the global economy. By forecasting the price of Brent crude oil, we can gain valuable insights into the direction of the economy and make more informed investment decisions. Additionally, forecasting the price of Brent crude oil can help businesses and organizations plan for the future and avoid potential financial losses. Overall, Brent crude oil price forecasting is an important tool for understanding and predicting the direction of the global economy and for making informed business decisions.

In this research, we have modified and enhanced the stacked model by various approaches including adding more features, more models to be ensembled in each layer, a greater number of nodes in the ANN's hidden layer, more layers in the stacked model, and exploring different optimizers for error backpropagation. The study focuses on enhancing the stacked model to its best performance while also taking computational constraint trade-offs into account.

In conclusion, our research demonstrates that the stacked model is an effective approach for predicting Brent crude oil prices, outperforming both the LSTM and GRU models. It is found that the Adam optimizer produced the best results in this context. Additionally, we discovered that simply increasing the number of nodes in the ANN's hidden layer does not necessarily lead to improved performance. Interestingly, we also noticed that the deeper the layer, the lower the SDAPE, but this was not necessarily true for the MSE and RMSE metrics. Ultimately, our findings suggest that a well-designed stacked model with carefully selected features and hyperparameters can be an effective tool for predicting future oil prices, though careful consideration must be given to computational constraints and the specific performance metrics of interest.

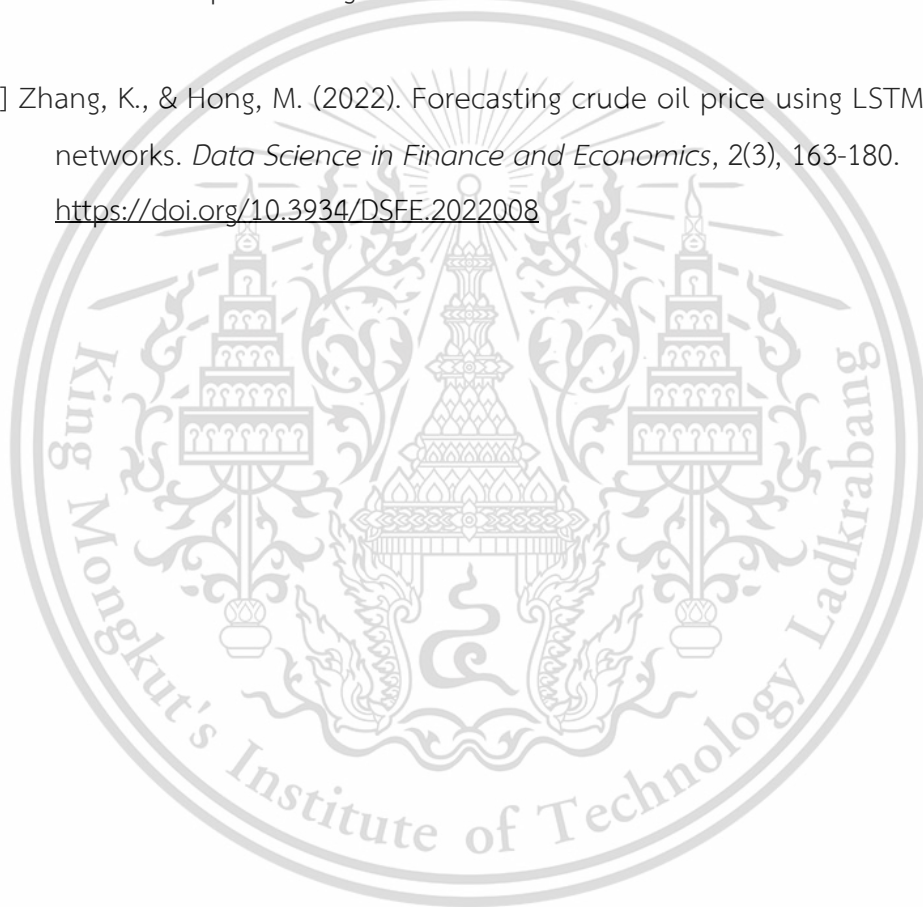
For further research, it is recommended to explore different combinations of features, such as technical indicators, to enhance the stacked model's performance. In addition, using different data sources could potentially improve the model's performance. It is also worth investigating the effects of other optimizers, testing the model on different time periods or asset classes. Additionally, alternative machine learning algorithms, such as gradient boosting or transformer, could be put into the stacked model to further improve the model's accuracy in forecasting asset prices.



## REFERENCE

- [1] Chi, Y. N., & Chi, O. (2022). Time Series forecasting global price of bananas using Hybrid ARIMA-NARNN model. *Data Science in Finance and Economics*, 2(3), 254-274. <https://doi.org/10.3934/DSFE.2022013>
- [2] Gabralla, L. A., Jammazi, R., & Abraham, A. (2013). Oil price prediction using ensemble machine learning. 2013 International Conference On Computing, Electrical and Electronic Engineering (ICCEEE), DOI: 10.1109/ICCEEE.2013.6634021
- [3] Gao, S., & Lei, Y. (2017). A new approach for crude oil price prediction based on stream learning. *Geoscience Frontiers*, 8(1), 183-187. <https://doi.org/https://doi.org/10.1016/j.gsf.2016.08.002>
- [4] Gupta, N., & Nigam, S. (2020). Crude Oil Price Prediction using Artificial Neural Network. *Procedia Computer Science*, 170, 642-647. <https://doi.org/https://doi.org/10.1016/j.procs.2020.03.136>
- [5] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [6] Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, 33(10), 4741-4753. <https://doi.org/10.1007/s00521-020-05532-z>
- [7] Manaswi, N. K. (2018). RNN and LSTM. In *Deep Learning with Applications Using Python : Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras* (pp. 115-126). Apress. [https://doi.org/10.1007/978-1-4842-3516-4\\_9](https://doi.org/10.1007/978-1-4842-3516-4_9)

- [8] Sethia, A., & Raut, P. (2019). Application of LSTM, GRU and ICA for Stock Price Prediction. In S. C. Satapathy & A. Joshi, *Information and Communication Technology for Intelligent Systems, Smart Innovation, Systems and Technologies*, vol 107. Springer, Singapore, 479-487. [https://doi.org/10.1007/978-981-13-1747-7\\_46](https://doi.org/10.1007/978-981-13-1747-7_46)
- [9] Tfaily, F., & Fouad, M. M. (2022). Multi-level stacking of LSTM recurrent models for predicting stock-market indices. *Data Science in Finance and Economics*, 2(2), 147-162. <https://doi.org/10.3934/DSFE.2022007>
- [10] Zhang, K., & Hong, M. (2022). Forecasting crude oil price using LSTM neural networks. *Data Science in Finance and Economics*, 2(3), 163-180. <https://doi.org/10.3934/DSFE.2022008>



## AUTHOR BIOGRAPHY

Name : Mr. Marit Asavamahakul

Date of Birth : 14 August 2539

Education : Bachelor of Arts (Communication Arts) in Chulalongkorn University

GPA : 3.27



This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.