TIME SERIES MODEL FOR FORECASTING BAHT/USD EXCHANGE RATES USING TREE-BASED ALGORITHM AND ENSEMBLE DEEP LEARNING

PAWEENA ROUYPRASERT

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-017-051

Independent Study Title	Time Series Model for Forecasting Baht/USD Exchange Rates Using Tree-Based Algorithm and Ensemble Deep Learning
Student Name	Paweena Rouyprasert
Student ID	64605069
Degree	Master of Science (Data Science and Analytics) KMITL Digital Analytics and Intelligence Center
Year	2023
Independent Study Advisor	Dr.Jiraphat Yokrattanasak

Abstract

The goal of this study is to develop a model for predicting THB/USD using a Tree-Based Model and Ensemble Deep Learning. The experiment applies inputs such as foreign currency, stock index, crude oil price, precious metal prices, and holidays to enhance the model's performance. Data is extracted from Yahoo Finance and related libraries in Python, covering the period from April 1, 2018, to December 31, 2022. The significance of the study lies in the model's efficiency and reduced errors. The study demonstrates that employing CHANGE, a data manipulation technique for the target variable, improves prediction outcomes for the THB/USD trend by reducing fluctuations over time. The top 5 models in the study, namely lstm_lstm_2, gru_gru_2, lgbm_2, xgb_2, and lstm_gru_2, indicate that Ensemble Deep Learning outperforms Plain Deep Learning in terms of prediction. Additionally, the Tree-based model and Ensemble Deep Learning approach exhibit similar RMSE values, suggesting comparable performance. These findings may provide support and guidance to relevant stakeholders for improving their work, planning, or policies.

Keywords: Currency Exchange Rate, XGBoost, LightGBM, Deep Learning, Stacked Model

Acknowledgment

I express profound gratitude to the individuals who have contributed to developing and completing the model for Forecasting Baht/USD Exchange Rates Using Tree-Based Algorithm and Ensemble Deep Learning. Their contributions have been vital to the successful realization of this research project.

First and foremost, I would like to acknowledge and extend my appreciation to our advisors and mentors. Their guidance, support, and insightful perspectives have been instrumental throughout the entire research process. Their expertise and constructive feedback have greatly influenced the direction and quality of this Individual Study.

Additionally, I recognize the significant contributions of the academic community and researchers in the fields of data science, time series analysis, and deep learning. Their pioneering work and published findings have provided a solid framework upon which my research has been built. I am indebted to their valuable contributions, which have expanded the body of knowledge in these areas.

Furthermore, I would like to express my gratitude to my classmates: Pongpeera Jaruvijitrattan, Piyanuch Viwatborvornwong, and Marit Asavamahakul and loved ones who have provided unwavering support, encouragement, and understanding throughout the preparation of this document. Their presence and encouragement have been a source of inspiration, and we are grateful for their constant belief in our abilities.

In conclusion, I emphasize that the achievement of this research endeavor would not have been possible without the collective efforts, knowledge, and contributions of all those mentioned above. Their involvement and support have been invaluable, and I extend my heartfelt appreciation for their unwavering dedication to this Individual Study.

Miss Paweena Rouyprasert

TABLE OF CONTENTS

Page

TABLE OF CONTENTS : Table

TABLE OF CONTENTS : Figure

Figure Page

CHAPTER 1

INTRODUCTION

This section explores the statement of the problems, objectives, scope of study, and contribution of knowledge of THB/USD prediction in the study.

1.1 Statement of the problems

The exchange rate of foreign currencies is a significant factor in the economy, as stated by the Bank of Thailand. The exchange rate is a crucial aspect that businesses engaged in exports or imports must face. Changes in foreign currency exchange rates can lead to uncertainty in business revenue or expenses in Thai baht. This uncertainty can result in increased profits or losses. Similarly, the public is also affected by the fluctuations in foreign currency exchange rates because everyone needs to consume goods or raw materials imported from other countries, such as fuel and smartphones.

Currently, the Thai economy experiences significant volatility, primarily due to the global impact of the COVID-19 pandemic that emerged in 2020. The pandemic has led to continuous global economic downturns. Furthermore, the escalation of the Russia-Ukraine conflict on February 24, 2023, has increased the intensity of the situation. This has significantly affected the general price level of goods, as Russia's suspension of energy exports, one of the fundamental commodities, has implications for the daily lives of people, leading to increased production costs, adjustments in general price levels, and global inflation (Rungnapa Pimmatsri, 2023). Subsequently, the Federal Reserve System (FED) announced a 0.25% interest rate hike in March 2023, which was the first increase since keeping the interest rate at zero in 2018. This move aimed to curb purchasing power and control inflation (Money Buffalo, 2023). As a result, on September 16, 2023, the Thai baht reached its weakest point in 16 years, with an exchange rate of only 37.08 baht per US dollar. Following this, central banks in several countries, including the Bank of Thailand, adjusted their interest rates in line with the FED, leading to fluctuations in foreign exchange rates, including digital currencies (cryptocurrency), as interest rates impact both investment and borrowing (Kingploy, 2023). These events indicate that the US dollar is highly popular in the foreign exchange market and exerts influence on other currencies as well. Moreover, it is evident that the Thai economy is highly vulnerable and susceptible to global economic fluctuations or the economies of major powers. According to statistics from The Global Economy, Thailand ranked 33rd out of 174 countries in terms of openness in 2018, with a score of 120.84%. Additionally, data on Thailand's imports and exports between January and April 2023 revealed that China, Japan, and the United States were the top three trading partners (Department of Foreign Trade, 2023).

Based on Chalermrat Nontapa (2022), it was found that the value of the Thai baht is related to foreign currencies such as the US dollar, yuan, and yen. The Thai baht is also influenced by the level of electricity consumption in the country, diesel oil, the average temperature in each month, and the average price level of the Stock Exchange of Thailand's main index (SET Index) for each month. This study demonstrates that, besides external factors, the Thai baht is also affected by internal factors within Thailand.

Therefore, this study aims to create a model to predict the exchange rate of the Thai baht against the US dollar using deep learning techniques and incorporating factors from both external and domestic sources, such as the interest rate of the United States or the value of the Stock Exchange of Thailand's main index. Additionally, alternative data, such as cryptocurrencies, will be utilized to mitigate investment risks and foreign currency exchange transactions between the Thai baht and the US dollar.

1.2 Objectives

1. To investigate and compare the effectiveness of tree-based algorithms and deep learning models in predicting the exchange rate between the Thai Baht and the US Dollar. The study focuses on examining the performance of these models in accurately forecasting currency exchange rates.
2. To explore the relationships between the Thai Baht exchange rate against the US Dollar and other selected variables incorporated in the various predictive models. The objective is to identify any significant associations or dependencies between these variables and the exchange rate.
3. To assess the effectiveness of various deep learning methods and structures, such as LSTM, GRU, and Ensemble Model.

4. To compare and explain the prediction between using actual target and using manipulated target.

1.3 Scope of Study

1. The dataset encompassing **daily records** of the exchange rates between the Thai Baht and the US Dollar, covering the period from April 2018 to December 2022, will be subjected to a thorough study.
2. Pertinent data such as Thai stock price, exchange currency rates, as well as the annual public holidays observed in Thailand and other countries, will be considered in the research investigation.

1.4 Contribution of Knowledge

1. To mitigate investment and foreign exchange risks associated with the Thai Baht to US Dollar currency pair, for both investors and the public.
2. To provide benefits for policy formulation or measures related to foreign exchange rates that fall under the supervision of the government, the Bank of Thailand, and relevant stakeholders.
3. To utilize it for the development of more efficient prediction models for the Thai Baht to US Dollar exchange rate.

CHAPTER 2

LITERATURE REVIEW

The chapter contains the definitions, theories, and literature used in the study for THB/USD prediction.

2.1 Machine Learning

Machine learning is a subfield of artificial intelligence involving the use of algorithms and statistical models to enable computers to learn from data and improve their performance over time to make decisions or predictions accurately. Supervised learning and unsupervised learning are the two primary categories within the field of machine learning. In supervised learning, the data set including input data and the corresponding correct output is used to train and test the algorithm respectively. An example of supervised learning might be a machine learning algorithm trained on a data set of images of different types of animals, along with labels indicating which type of animal is in each image. The algorithm is then tested on new images, and the accuracy of its predictions is evaluated based on how well it can classify the animals in the new images. Unlikely, in unsupervised learning, only input data is used to train the algorithm to discover patterns or relationships in the data on its own. One common application of unsupervised learning is clustering, where the algorithm groups similar data points together.

Due to predicting the THB/USD prices, supervised learning is selected to use in the study. One is the Tree-Based Algorithm described in section 2.1.1, and the other is Deep Learning described in section 2.2.

2.1.1 Tree-Based Algorithm

Tree-based algorithms, such as decision trees or ensembles of decision trees, are machine-learning algorithms used to address diverse predictive and classification tasks. These algorithms employ flowchart-like structures known as decision trees, which sequentially make decisions based on feature values, ultimately resulting in a final prediction or classification outcome.

These algorithms have gained popularity due to their interpretability, simplicity, and capacity to handle both numerical and categorical features. Some widely utilized tree-based algorithms are as follows:

1. **Decision Tree:** It involves a single decision tree that divides the data based on feature values, constructing a tree-shaped structure to generate predictions.
2. **Random Forest:** This ensemble method combines multiple decision trees and aggregates the results of individual trees to make predictions.
3. **Gradient Boosting:** This boosting algorithm builds decision trees sequentially, with each subsequent tree correcting the mistakes of the preceding ones. The predictions of all the trees are combined to produce a final prediction.
4. **XGBoost:** Extreme Gradient Boosting refers to an optimized implementation of gradient boosting that offers improved performance and speed.
5. **LightGBM:** Light Gradient Boosting Machine is another efficient gradient boosting framework that employs a leaf-wise growth strategy and various optimizations to achieve fast training speed.
6. **CatBoost:** This gradient-boosting algorithm effectively handles categorical features by utilizing diverse encoding techniques.

Tree-based algorithms find extensive applications across multiple domains, including finance, healthcare, marketing, and more. They are employed for various tasks such as classification, regression, feature selection, and anomaly detection.

In this study, XGBoost and LightGBM are applied to the experiment as several studies, such as (Jabeur, Mefteh-Wali, & Viviani, 2021) and (Jabeur, Mefteh-Wali, &

Viviani, 2021), suggested that these algorithms can perform efficiently through time series prediction.

The utilization of bagging or boosting techniques (shown in more detail in section 2.3) in time series prediction offers several advantages:

1. It effectively reduces variance by combining predictions from a diverse collection of models trained on distinct subsets of the data, resulting in more stable and accurate forecasts.
2. Bagging addresses the complexity inherent in time series data by constructing an ensemble of models that capture different facets of the data, enhancing the accuracy of predictions for nonlinear and intricate relationships.
3. Bagging mitigates overfitting by employing multiple models trained on different subsets, reducing the risk of excessively fitting noise or idiosyncrasies in the training data, and promoting better generalization to unseen data. Additionally, bagging improves the estimation of prediction intervals by considering the variability observed in predictions generated by the ensemble, leading to more reliable and precise uncertainty estimation.
4. Bagging demonstrates adaptability to changing patterns in time series data by training models on diverse subsets, enabling robust predictions even in the presence of evolving patterns.

Overall, the bagging technique proves advantageous in time series prediction by reducing variance, accommodating nonlinear relationships, mitigating overfitting, improving prediction intervals, and adapting to changing patterns, thereby enhancing the accuracy and reliability of time series forecasts.

2.1.1.1 XGBoost

XGBoost, known as Extreme Gradient Boosting, is a renowned machine learning algorithm celebrated for its remarkable performance across diverse domains. It has garnered significant attention owing to its capacity to handle heterogeneous data types, effectively handle missing values, and efficiently process voluminous datasets through parallel computing techniques. Belonging to the gradient-boosting algorithm family, XGBoost leverages an iterative approach to combine weak learners and construct a robust predictive model.

Within the XGBoost framework, decision trees are typically employed as weak learners. The algorithm incrementally builds an ensemble of decision trees, wherein each successive tree aims to rectify the errors made by its predecessors. This iterative process culminates in an ensemble model that furnishes highly accurate predictions.

XGBoost encompasses a range of hyperparameters that empower users to govern the behavior and performance of the algorithm as below:

General parameters encompass specifications such as the number of trees in the ensemble (n_estimators), the learning rate controlling each tree's contribution (learning_rate), and the fraction of training instances used for tree construction (subsample). Manipulating these parameters facilitates users in controlling overfitting and enhancing generalization.

Tree parameters pertain to aspects of individual decision trees within the ensemble. Parameters like max_depth dictate the maximum depth of each tree, balancing the complexity of captured patterns with the risk of overfitting. min_child_weight stipulates the minimum sum of instance weights required in a child node, aiding in managing complexity and curtailing overfitting. Additionally, gamma governs the minimum reduction in the objective function's loss needed to trigger further partitioning of a leaf node, thus encouraging pruning.

Regularization parameters integrate regularization techniques into the model to curb overfitting and manage complexity. lambda regulates the L2 regularization term on leaf node weights, serving as a penalty against large weights. Simultaneously, alpha governs the L1 regularization term, potentially inducing sparsity in the feature space.

The hyperparameters represent merely a subset of the comprehensive range provided by XGBoost. Selecting suitable hyperparameters is of paramount importance to achieve optimal model performance. This often necessitates the adoption of hyperparameter tuning methodologies, such as cross-validation, to assess diverse hyperparameter combinations and identify the most favorable settings based on performance metrics.

In conclusion, XGBoost is an exceedingly proficient gradient-boost algorithm employing decision trees as weak learners. Its extensive repertoire of hyperparameters empowers researchers and practitioners to tailor the model's behavior and performance to specific requirements. By judiciously configuring these hyperparameters, users can unlock the full potential of XGBoost, delivering accurate predictions across diverse machine-learning tasks.

2.1.1.2 LightGBM

LightGBM, also known as Light Gradient Boosting Machine, is a gradient boosting framework that has gained popularity in the field of machine learning due to its exceptional computational speed and accuracy. It shares the gradient boosting paradigm with other algorithms like XGBoost but distinguishes itself through optimizations that enhance its computational efficiency. LightGBM leverages decision trees as base learners and constructs an ensemble of these trees to make predictions. Its architecture involves an objective function comprising a loss function and a regularization term, gradient boosting for iterative model training, and a leaf-wise tree growth strategy for efficient tree construction. Various regularization techniques are employed to control model complexity and prevent overfitting.

When it comes to hyperparameters, LightGBM offers a range of options that can be fine-tuned to optimize model performance. Key hyperparameters include the learning rate, which governs the influence of each tree on the ensemble; the number of trees in the ensemble; the maximum depth of each tree; the fraction of features and training data randomly selected for each tree (feature fraction and bagging fraction, respectively); and regularization parameters such as lambda and gamma.

In summary, LightGBM is a highly efficient gradient-boosting framework that utilizes decision trees as base learners. It stands out for its computational speed and accuracy, making it suitable for large-scale datasets. LightGBM incorporates optimizations, such as leaf-wise tree growth and regularization techniques, to improve computational efficiency and control model complexity. By fine-tuning hyperparameters, it can be customized for various machine-learning tasks.

2.2 Deep Learning

Deep Learning, a subset of machine learning, focuses on learning successive layers, whereas other techniques of machine learning often focus on learning a few layers of representations. Additionally, deep learning is known as hierarchical learning. (Chollet, 2018)

The most notable distinction between the two types of neural networks is Deep Learning employs neurons to alter, extract, and create relationships between properties, meanwhile, neural networks use neurons to convey data in the form of input values and output values. Regardless, Neural networks or **Artificial Neural Networks** are a small particular in deep learning, in other words, it is recognized as vanilla deep learning.

Figure 2.1 the structure of the Neural Network
(Source: https://towardsdatascience.com/neural-networks-parameters-hyperparameters-and-optimization-strategies-3f0842fac0a5)

The architecture of **plain Deep Learning** is made up of three essential elements as the following.

Neurons: In Figure 1, the different circles are "neurons". What neurons do depend on where they are; for instance, if they are in input or layers, they contain the characteristics of an input or an output respectively. If they appear in Hidden Layers, they are a container for the formula, called the activation function.

Input Layer: The input layer acts as an outpost of the networks to receive the data and transmit it to the Hidden layer, the following layer.

Hidden Layer: The data is transformed by an activation function in each neuron. The additional hidden layers indicate how complex deep learning is, but there is no guarantee that the model will be more effective. (L, 2020)

Output Layer: The Output Layer is the final layer in deep learning. The values from the last hidden layer are transferred to this layer. Different neurons receive various Weights in each output layer, which heavily influences the output result. In technical terms, Weights are also called Parameters of a layer.

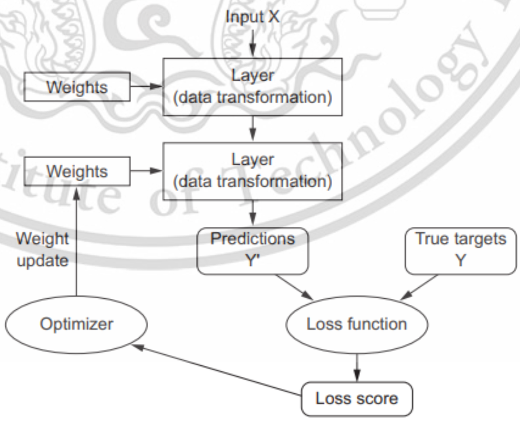

Figure 2.2 the structure of the modern deep learning
(Source: https://www.researchgate.net/publication/349023494)

Next, Figure 2 shows the anatomy of **Modern Deep Learning** in general. To establish the ideal forecasting. A loss function and Optimizer are added to the method to help the user control the expected output. Using the **Loss function** lets us know how far the output is supposed to be from its prediction. After that **Loss score** plays as a signal to update the value of the weight through the **Optimizer** using **Backpropagation** which is a mathematical tool for investigating how errors work reversely through output and input nodes. In the process, the loss function is required to minimize the errors, in other words, loss to find the best practice. (Chollet, 2018) Learning will stop when it encounters the minimum error.

2.2.1 Loss Function

The regression model typically returns a continuous target value for x, whereas the classification model returns a discrete target value for x. Their goal is to learn a function that achieves the lowest loss across the training data. The Loss function of the regressor is formulated as follows:

$$\min_{f(x)} \sum_{i=1}^{n} l(f(x_i) - y_i) + \mathcal{R}_\lambda(f)\,, \tag{2.1}$$

Overall, the Loss function is represented by $l(r)$ where $f(x_i) - y_i$ is the deviation between a function and an independent variable. $R_\lambda(f)$ is referred to as the regularization term to reduce the chance of overfitting.

While the Loss function of binary classification is formulated as:

$$\min_{f(x)} \sum_{i=1}^{n} l(y_i f(x_i)) + \mathcal{R}_\lambda(f)\,, \tag{2.2}$$

the Loss function is represented by those parameters as same as those of the regression where $y_i f(x_i)$ is the deviation between $f(x_i)$ and hyperplane. (Feiping Nie, 2018)

2.2.2 Optimization Learning Algorithm

As mentioned, the minimum loss is a desire for model learning improvement. Metrics for determining the efficacy of an optimizer include the speed of convergence where the learning reaches a global optimum, and Generalization where the model can perform efficiently on unseen data are metrics to determine the efficacy of an optimizer. Stochastic Gradient Descent (SGD) and **Adaptive Moment Estimation (Adam)** are well-known algorithms utilized by numerous models to outperform these indicators. (Chalermrat, 2022)

Adaptive Moment Estimation (Adam) Optimizer is used in the study. Many shreds of evidence point, such as (Kingma & Ba, 2015), (Tato & Nkambou, 2018), (Bushaev, 2018), (Gupta A. , 2021), and (Alom, 2021) show Adam is the recommended default optimizer for model development because it can overwhelm non-convex optimization problems.

There is a combination, including Gradient Descent with Momentum and Root Mean Square Propagation (RMSP), used in the Adam optimizer. Due to aiming to the minima with faster learning, Momentum is for speeding up the gradient descent algorithm by considering the "exponentially weighted average" of the gradients. However, Root Mean Square Propagation (RMSP) is for taking the "exponential moving average" instead of taking the accumulative sum of squared gradients. **Adam Optimizer Algorithm** can be explained as the following formulas:

$$\theta_i^{(r)} = \theta_{\mathrm{i}-\varepsilon}^{(r-1)} \frac{\partial L}{\partial \theta}\left(\theta_i^{(r-1)}\right) \tag{2.3}$$

Where $\mathcal{E}$ is the learning rate. (described in section 2.1.4) Adam improves the learning rate for each neuron weight by estimating the gradient's first and second moments. They offset these biases by calculating bias-corrected estimates of the first and second moments:

$$m_{1+t} = \beta_1 m_{1+t} + (1 - \beta_1) g_t \tag{2.4}$$

$$m_{2+t} = \beta_2 m_{2+t} + (1 - \beta_2){g_t}^2 \quad (2.5)$$

$$\widehat{m}_{1,t} = \frac{\beta_1 m_{1,t}}{1-\beta_1^t} \quad (2.6)$$

$$\widehat{m}_{2,t} = \frac{\beta_2 m_{2,t}}{1-\beta_2^t} \quad (2.7)$$

Table 2.1 Algorithm of Adam optimizer

Algorithm of Adam optimizer	
1:	Fix the parameters ε: learning rate, m: batch size, nb: number of epochs.
2:	For ℓ = 1 to nb epochs
3:	For ℓ = 1 to n/m
4:	select an unreplaced random batch of size m in the learning sample: $(x_i, y_i)_{i\epsilon B}$
5:	utilize the backpropagation algorithm to calculate the gradients. $\tilde{\nabla}_\theta = \frac{1}{m} \Sigma_{i\epsilon B} \nabla_\theta l\left(f(x_i, \theta), y_i\right)$
6"	6: update the parameters $\theta^{(r)} = \theta^{(r-1)} - \varepsilon\tilde{\nabla}_\theta$

Again, the same formula as Table 2.1 is used for the last updating process except for the gradient in the nominator is replaced by m1, t. Then it should be obvious that Adam joins the approaches of its predecessors and develops into a superior version. Note that ε is just a short-term preventing division by zero.

$$\theta_i^{(r)} = \theta_i^{(r-1)} - \varepsilon^1 \frac{m_{1,t}}{\sqrt{\widehat{m}_{2,t}+t}} \quad (2.8)$$

$$g_t = \frac{\widehat{m}_{1,t}}{\sqrt{\widehat{m}_{2,t}}+1} \quad (2.9)$$

2.2.3 Dropout

Dropout Method is a regularization method for minimizing overfitting in neural networks by eliminating compliance requirements on training data. The most common method for deep learning is the dropout presented by (Hinton, et al., 2012).

With a certain probability p and independently of the others, the probability p is another hyperparameter and each network unit is set to 0. Whereas 0.5 is for teams working on hidden levels and 0.2 for teams working on entry layers, however, it is still necessary to increase the number of epochs because training takes longer. To cut down on the computational overhead, setting a few weights with probability p to 0 is required. At present, dropout is the most popular regularization approach to enhance the generalization ability of deep neural networks.

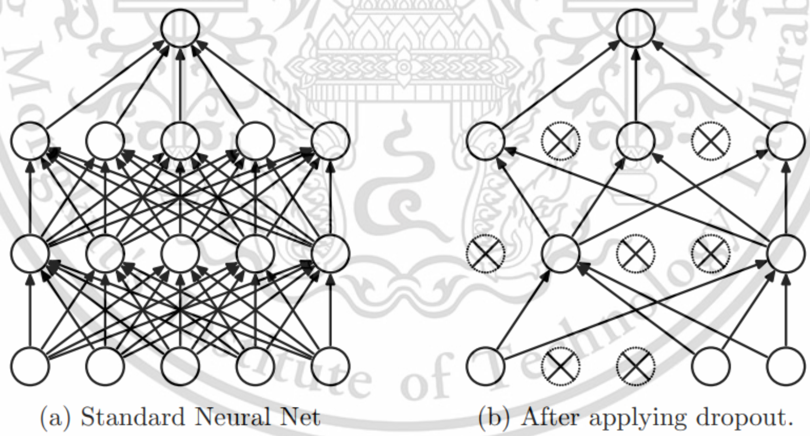

Figure 2.3 the structure of the Neural Network Before Dropout (Left) and After Dropout (Right)

(Source: https://medium.com/analytics-vidhya)

2.2.4 Hyperparameter

Nevertheless, to establish the best practice from an experiment, the network structure is also needed to set up suitable variables, called hyperparameters consisting of the following factors before the training mode.

Learning Rate: This factor is used to regulate how quickly a network picks up new data. The higher the rate, the faster learning in the process. In contrast, the lower the rate, the slower down learning. If the optimization is too tiny, it can be limited to a local minimum. If the learning rate is too high, the network will bubble about an optimal point without stabilizing or converging.

Momentum: the variable dominates the upcoming direction of the knowledge from the previous steps. The range of common values of momentum is between 0.5 and 0.9 (Radhakrishnan, 10).

Number of epochs: it controls the frequency of the data sent through the network for training. The model's accuracy rises as learning time increases, but the risk of overfitting also rises.

Batch size: the size is used to determine how larger the sub-sample given running to the network. Typically, the value is 32, 64, 128, or another number.

Learning rate: The learning rate determines how fast the model adjusts the weights and biases during training. A smaller learning rate means that the model will make smaller adjustments, which can lead to slower but more accurate training. A larger learning rate means that the model will make larger adjustments, which can lead to faster training but may result in less accurate results.

Number of layers: The number of layers in the network determines the depth of the model. A deeper model with more layers can learn more complex patterns in the data, but it may also be more prone to overfitting (performing poorly on new, unseen data).

Number of neurons: The number of neurons in each layer determines the model's capacity and how much it can learn. A larger number of neurons means that the model has more capacity to learn, but it may also be more prone to overfitting.

Activation function: The activation function is used to introduce nonlinearity into the model. Different activation functions can affect the model's ability to learn certain patterns in the data.

Weight initialization: The initial values of the weights in the model can affect the model's ability to learn. Different weight initialization techniques can be used to improve the model's performance.

Batch size: The batch size determines the number of training examples used in each iteration of training. Larger batch size can lead to faster training, but it may also result in less accurate results.

Dropout rate: Dropout is a regularization technique that involves randomly dropping out (setting to zero) a certain percentage of neurons during training. This helps to prevent overfitting by reducing the complexity of the model. The dropout rate determines the percentage of neurons that drop out.

Early stopping: Early stopping is a regularization technique that involves interrupting training when the model's performance on the validation set begins to degrade. This can help to prevent overfitting by limiting the amount of training the model receives.

In summary, hyperparameters are parameters of a deep learning model that are set before training and are not learned during the training process. They are used to control the behavior of the model and can significantly affect the model's performance. Tuning hyperparameters is an important step in the deep learning process and can be done using techniques such as hyperparameter optimization.

2.2.5 Deep Learning Framework

This study is designed to use approaches best suited for time series forecasting consisting of **Long Short-term Memory (LSTM),** and **Gated Recurrent (GRUs)** (Panda, Panda, & Pattnaik, 2020), (Jimeng Shi, 2020), (Mohammed, 2021). To develop the forecasting, using **Ensemble Learning** is applied in the study and explained in section 2.3.

2.2.5.1 Recurrent Neural Networks (RNN)

Generally, RNNs have been suited for modeling time series data (Hansika Hewamalage, 2020). The architecture of RNN consists of the functional relationship between input variables in the previous process to a target variable in the upcoming step.

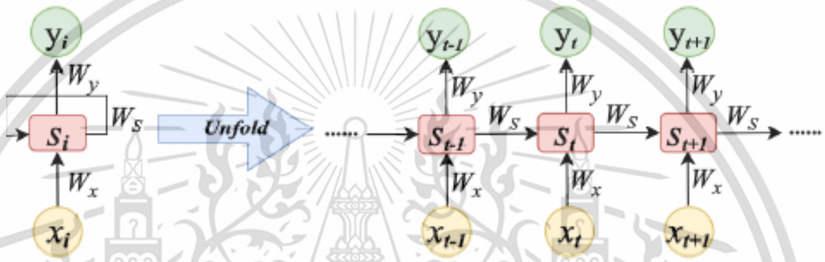

Figure 2.4 the structure of Recurrent Neural Networks

(Source: https://medium.com/analytics-vidhya/how-long-dependencies-can-lstm-t-cnn-really-remember-df3720dd4e4a)

With the emphasis on the transitions of a hidden state from time t – 1 to time t, an RNN is taught from a training set of historical data shown in Figure 4. To improve the model, the result of the approach is addressed by three matrices including w_x, w_y, and w_s, along with two bias vectors including b_s and b_y. The output y_t relies on the internal state s_t. While on the internal state s_t is determined by current input x_t and the late state s_{t-1}.

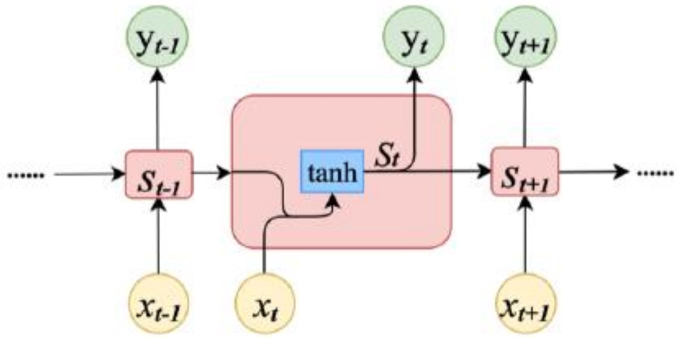

Figure 2.5 Computation process in the RNN

(Source: https://medium.com/analytics-vidhya/how-long-dependencies-can-lstm-t-cnn-really-remember-df3720dd4e4a)

Figure 2.5 describes the computational process of each hidden state with the given formula as follows:

$$s_t = tanh(w_{xs} \cdot (x_t \oplus S_{t-1}) + b_s \quad (2.10)$$

$$y_t = \sigma(x_y \cdot s_t + b_y) \quad (2.11)$$

where $x_t \in \mathbb{R}^m$ is the input vector of m input features with time t, $w_{xs} \in \mathbb{R}^{n\times(m+n)}$ and $w_y \in \mathbb{R}^{n\times n}$ are parameter matrices, n is the number of neurons, b_s and $b_y \in \mathbb{R}^N$ are bias vectors for the internal state and output respectively, σ is referred to the sigmoid activation function, S_t is the internal (hidden) state, and $x_t \bigoplus S_{t-1}$ is the concatenation of vectors, x_t and S_{t-1}.

Furthermore, long-term dependencies are learned using recurrent neural networks as an initiative technique. Because the gradients are essential to transmit the information in updating parameter processes, which become fewer as the gradient diminishes, RNNs suffer from the vanishing and expanding gradient problem, which makes learning from greater data difficult. (Arbel, 2018)

2.2.5.2 Long Short-term Model (LSTM)

Long Short-term Memory (LSTM) is an adaptation of RNNs that can relieve the vanishing gradient problem of RNNs and learn longer-term dependencies.

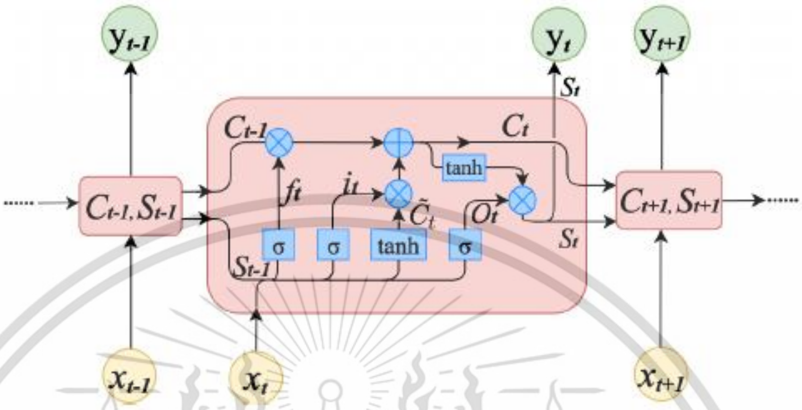

Figure 2.6 the structure of Long Short-term Model

(Source: https://medium.com/analytics-vidhya/how-long-dependencies-can-lstm-t-cnn-really-remember-df3720dd4e4a)

From figure 2.6, LSTM is explained at time t in terms of a hidden state, S_t, and a cell state, C_t. There are three dependencies in terms of C_t composing of C_{t-1} which is referred to the previous hidden state, S_{t-1} which referred to the previous internal state, and x_t which is referred to input at the current time. For filtering, multiplication, and addition of information in controlling longer-term dependencies, there are forget gate, input gate, addition gate, and output gate using the mathematic functions as f_t, i_t, $\tilde{C}_t$, and O_t respectively.

$$f_t = \sigma\left(w_f \cdot (x_t \oplus s_{t-1}) + b_f\right) \tag{2.12}$$

$$i_t = \sigma(w_i \cdot (x_t \oplus S_{t-1}) + b_i) \tag{2.13}$$

$$\tilde{c}_t = tanh(w_c \cdot (x_t \oplus S_{t-1}) + b_c) \tag{2.14}$$

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

(2.15)

$$O_t = \sigma(w_0 \cdot (x_t \oplus S_{t-1}) + b_0) \tag{2.16}$$

$$S_t = tanh(C_t) \cdot O_t \tag{2.17}$$

$$y_t = \sigma\left(w_y \cdot s_t + b_y\right) \tag{2.18}$$

where $x_t \in \mathbb{R}^m$ is the input vector of m input features at time t; w_f, w_i, w_c, $w_o \in \mathbb{R}^{n\times(m+n)}$ and $w_y \in \mathbb{R}^{n\times n}$ are parameter matrices, n is the number of neurons, b_f, b_i, b_c, b_0, $b_y \in \mathbb{R}^n$ are bias vectors, σ is referred to the sigmoid activation function, S_t is the internal (hidden) state.

After researchers proposed various variants of LSTMs to verify the performance, LSTMs became one of the famous algorithms. Next, Researchers have added more gates to the original LSTM architecture with a purpose of a more effective LSTM. The additional multiple gates in developing LSTM are known as Gated Recurrent Unit (GRU). Long-term memory (LSTM) and Gated Recurrent Units (GRU) perform better than Recurrent Neural Networks for Time Series Forecasting. (Amal Mahmoud, 2021)

2.2.5.3 Gated Recurrent (GRUs)

Gated Recurrent Units (GRU) is a variant of LSTMs to further address the vanishing gradient problem, but with fewer parameters than those of LSTMs.

In the algorithm, the update gate, reset gate, and the third gate, implementing as z_t, r_t and $\tilde{s}_t$ respectively, are applied in the process. Each gate performs a different task. One controls how to filter, use, or combine information. The first component in the expression over the next state, given by $(1 - z_t) \cdot s_{t-1,}$ is for determining what to retain from the past with $z_t \cdot \widetilde{S}_t$ that control what to collect from the current memory.

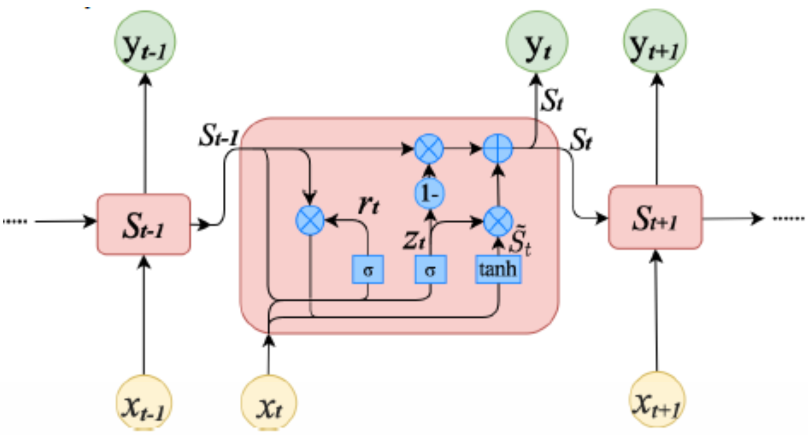

Figure 2.7 the structure of Gated Recurrent

(Source: https://medium.com/analytics-vidhya/how-long-dependencies-can-lstm-t-cnn-really-remember-df3720dd4e4a)

$$r_t = \sigma(w_r \cdot (x_t \oplus s_{t-1}) + b_r) \quad (2.19)$$

$$z_t = \sigma(x \cdot (x_t \oplus s_{t-1}) + b_z) \quad (2.20)$$

$$\tilde{s}_t = tanh(w_s \cdot (x_t \oplus s_{t-i} r_t) + b_s) \quad (2.21)$$

$$s_t = (1 - z_t) \cdot s_{t-1} + z_t \cdot \tilde{s_t} \quad (2.22)$$

$$y_t = \sigma\left(w_y \cdot S_t + b_y\right) \quad (2.23)$$

where $x_t \in \mathbb{R}^m$ is the input vector of m input features at time t ; $w_r, w_z, w_s \in \mathbb{R}^{n\times(m+n)}$ and $w_y \in \mathbb{R}^{n\times n}$; n is the number of neurons in the GRU layer; b_r, $b_z, b_s, b_y \in \mathbb{R}^n$ are bias vectors; σ is the sigmoid activation function; and S_t is the internal (hidden) state.

2.3 Ensemble Learning

An ensemble is a combination of predictions from two or more models. The models that influence the ensemble, also known as ensemble members, may be the same or different algorithms before training data. Perhaps, the ensemble members' predictions are merged using statistics, such as the mode or the mean, or by more complicated approaches. (Singh, 2022)

Over time, the various ensemble methods have evolved, resulting in better generalization of the learning approaches. Many studies in an improvement of currency exchange rate or price forecasting such as (Yunze Li, 2022), (Suna, Wang, & Wei, 2020), (Filippou, Rapach, Taylor, & Zhou, 2021), (Gang Wang, 2021), (Isaac Kofi Nti, 2020) and (Trilok Nath Pandey, 2020) show the ensemble learning leads the prediction model by machine learning or deep learning perform better.

Generally, the ensemble strategies (M.A. Ganaiea, 2022) can be classified as follows:

2.3.1 Bagging Ensemble Learning

Bagging, also known as "bootstrap aggregating," is one of the main techniques for making ensemble-based algorithms. To enhance the performance of ensemble learning, bagging is employed. The principal goal of bagging is to develop a set of data points with the same size and distribution as the original data. Given the pattern of observations, construct an ensemble predictor that is stronger than the learning algorithm derived from the initial data. Two more steps have been taken. The generation of bagged samples and the passing of each bag of data to the underlying models then provide a mechanism for integrating the predictions of different predictors with or without replacement. When combining the results of the base classifier, the ensemble result may be different because classification predictors tend to use majority voting while regression tends to use averaging. As shown in Figure 2.8 where Di represents the bagged datasets, Ci represents the algorithms and Fens calculates the outcome.

2.3.2 Boosting Ensemble Learning

In ensemble models, the boosting technique is used to improve a weak learning model into one with enhanced generalization. Figure 2.9 illustrates a diagram of the boosting method. Methodologies, such as majority voting in classification or weak learners in regression problems, result in more precise predictions than a single weak learner independently. Now, boosting techniques such as AdaBoost and Gradient Boosting have been adopted.

2.3.3 Stacking Ensemble Learning

Stacking ensemble learning, described in Figure 2.10, is another process that combines various predictive models into a single model by stacking. This technique presents the idea of meta-learning, defines an optimal learning system, and aims to minimize generalization errors by diminishing the bias of its generalizers. A stacked system is constructed using the predictions of weak learners as features. As a result, those features allow the final model to combine the initial models, and the model disregards results that underperformed. Often, the model consists of diverse predictors that are trained in parallel. The model is divided into training sets; next, the baseline model is examined, and training is performed; and last, predictions are generated based on the training set. The predictions from the training set are leveraged as features to construct the final ensemble learning model. (St´efano Frizzo Stefenon, 2022)

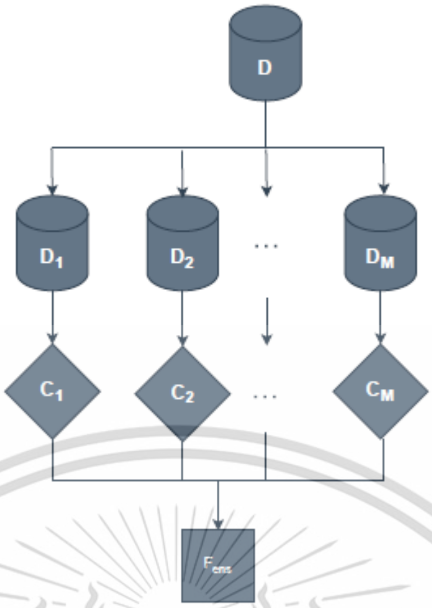

Figure 2.8 Bagging

(Source: https://www.researchgate.net/publication/339851307)

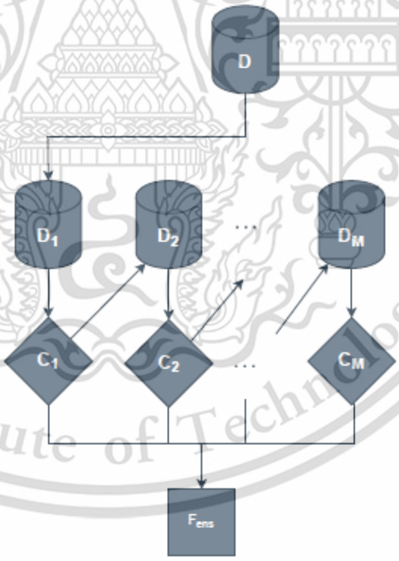

Figure 2.9 Boosting

(Source: https://www.researchgate.net/publication/339851307)

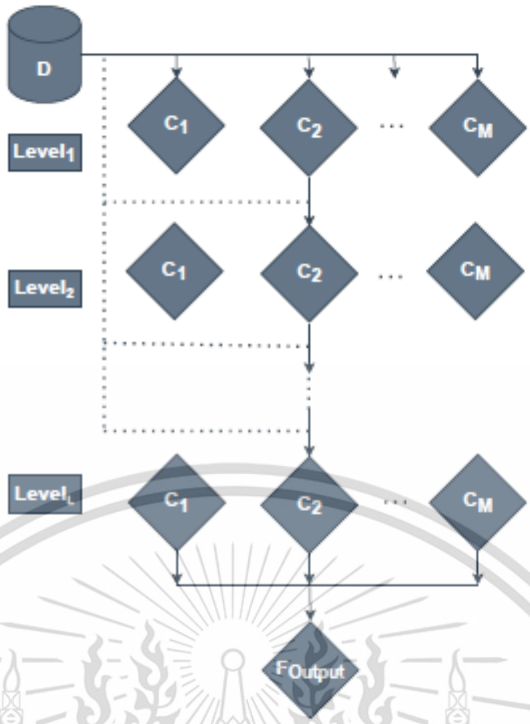

Figure 2.10 Stacking

(Source: https://www.researchgate.net/publication/339851307)

2.4 Evaluation Technique

Numerous criteria have been used to compare the performance of different models' predictions of price trends and price movement direction. In the experiments, there are three commonly used indicators consisting of Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) which are widely used evaluation metrics in regression analysis. They assess the performance of predictive models by quantifying the differences between predicted and true values. MSE calculates the average squared difference, RMSE provides a more interpretable measure by taking the square root of MSE, and MAE measures the average absolute difference. These metrics offer valuable insights into the accuracy and magnitude of errors in regression models. Where y_i is the actual value and $\hat{y}_i$ is the predicted value of the model.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - y\hat{\imath}| \qquad (2.24)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\frac{y_i - \hat{y}_i}{y_i})^2 \quad (2.25)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\frac{y_i - \hat{y}_i}{y_i})^2} \quad (2.26)$$

2.5 Exchange Rate

The ability to predict currency exchange rates holds significant implications across various domains. Forecasting these rates is valuable for financial planning, enabling individuals, businesses, and governments to anticipate fluctuations and make informed decisions. In international trade and commerce, accurate predictions assist with pricing, contract negotiation, and trade optimization. Participants in foreign exchange markets rely on exchange rate forecasts for trading strategies and risk management. Macroeconomic analysis benefits from these predictions, aiding policymakers in formulating effective monetary and fiscal policies. Furthermore, the tourism industry utilizes currency forecasts to assist travelers in planning trips and help businesses adjust pricing strategies. Overall, accurate predictions of currency exchange rates contribute to decision-making, risk mitigation, and economic optimization in multiple sectors.

However, many currency exchanges are dominated by other factors. Since the Thai Baht is an inferior currency compared to the USD. Its value is often dominated by other currencies such as USD, CAD, JPY, and SGD. (Investing in Thai Baht, n.d.), (Thailand's currency weakens to 38 baht per US dollar, 2022). As (Chalermrat, 2022) explained, the Thai Baht also is dominated by other factors such as SET which is the Stock Exchange Value in the Thai stock market, and Diesel price which is a major fuel of transportation. Additionally, there is evidence indicating that USD is related to other crypto/currencies such as CNY, GBP, CHF, and Bitcoin (Dash, 2018), (Shaolong Sun, 2020), (Panda, Panda, & Pattnaik, 2020)

The pieces of evidence mentioned suggest international currencies, stock prices, and fuel price impact currency prices. These factors would be applied to the study to improve the accuracy of the prediction.

2.6 Literature Review

Shaolong Sun (2020) focused on forecasting the exchange rates between the US dollars (USD) against the other four major currencies, such as GBP, JPY, EUR, and CNY using ensemble deep learning. The study found a new ensemble deep learning approach called LSTM-B by integrating a long-short-term memory (LSTM) neural network with bagging ensemble learning. Nevertheless, the study focuses on the exchange rate only, while other affecting the exchange rate factors are not taken into consideration. It is essential to select other component predictors which can enhance the accuracy of the model.

Ling Qi (2020) had the main objective of this study in financial trading focused on predicting a retracement point providing a perfect entry point to gain maximum profit under Forex Trading. RNN, LSTM, GRU, and Bi-LSTM are utilized in the experiment. The best model on 15-minute interval data for the EUR/GBP currency achieved RMSE 0.0015 with MAPE 0.12%e outperforming previous studies.

M.S. Islam (2021) presented a new model using an ensemble model between Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), for predicting the future prices of FOREX currencies among four major currency pairs including EUR/USD, GBP/USD, USD/CAD, and USD/CHF. The experimental results point out that the GRU-LSTM ensemble model predicted prices on the FOREX currencies more accurately than a single LSTM and GRU when testing in 10- and 30-minutes datasets. Though, the suggestion for further study is evaluating the accuracy of the proposed model for 5 and 15 minutes before the actual time.

Jabeur, Mefteh-Wali, & Viviani (2021) compared different machine learning models for predicting gold prices, with XGBoost outperforming other techniques and benchmark models. Significant correlations are found between gold prices and various predictor variables, demonstrating their ability to forecast future gold price volatility. The study introduces the SHAP method for interpretable machine learning, improving the understanding of complex models like XGBoost and capturing nonlinear relationships. The findings have practical implications for investors, policymakers, and traders, enabling better decision-making, risk management, and identification of factors influencing gold prices.

Paliari, Karanikola, & Kotsiantis (2021) examined the feasibility of using modern machine learning techniques, specifically LSTM and XGBoost, for forecasting financial time series data. Australian stock market data was analyzed, and the advantages and disadvantages of both methods were considered. The experimental results showed that LSTM and XGBoost outperformed the traditional ARIMA model in various metrics, except for cases with extremely low closing values. This suggests that modern ML techniques are suitable for stock market prediction and tend to outperform traditional statistical approaches. The study also emphasized the importance of defining prediction criteria and proposed incorporating exploratory data analysis to enhance decision-making. Future work includes comparing different markets and developing a hybrid model that combines LSTM and XGBoost for improved prediction accuracy.

Chalermrat (2022) studied in improving time series forecasting using deep learning along with an experiment in a hybrid model. The hybrid model is a kind of stacking ensemble learning between traditional regression and deep learning. SARIMAX model and LSTM with application to six simple monthly datasets, including electricity consumption in a Thai province, diesel consumption in Thailand, average temperature in Bangkok, Thailand Stock Market Index, USD/THB exchange rate, and JPY/THB exchange rate is the best practice compared to other models in the study. While Gated Recurrent Units is a suggested algorithm to study in future research and the hybrid model extension will be a challenge with a large dataset.

Yunze Li (2022) used XGBoost, LightGBM, LSTM, and GRU to establish stacking models and compare their results to a single model in the currency exchange prediction of AUD/USD and EUR/USD. With 31 stacking combinations, the model stacked by XGBoost, LightGBM, and GRU has the minimum RMSE when using the AUD/USD dataset, however, Random Forest, LSTM, and GRU have the minimum RMSE when using EUR/USD. To conclude, the result of the study shows that the performance is improved with only certain combinations, and overall, the combinations with both LSTM and GRU outperform other models.

CHAPTER 3
METHODOLOGY

This section delineates the approach utilized in this research to predict the Thai Baht by employing time series data and deep learning models, with a particular emphasis on the influence of USD exchange rates. In this section, we elaborate on the comprehensive procedure adopted for implementing the methodology.

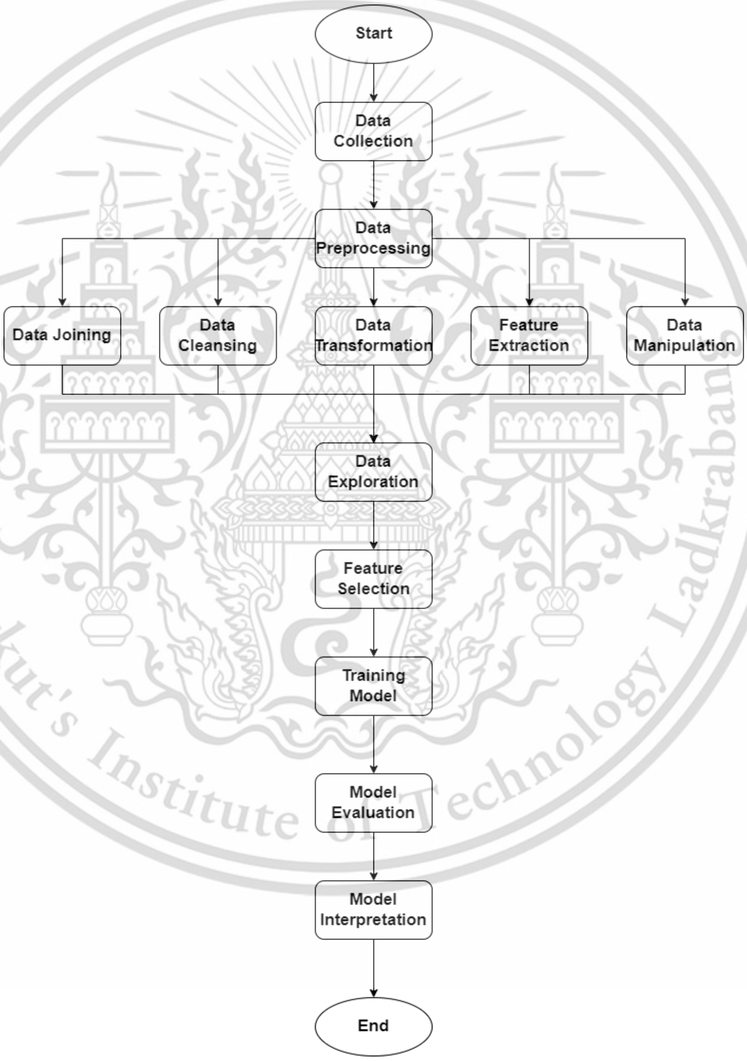

Figure 3.1 Flowchart of the main process in the study

3.1 Data Collection

In this study, there are 3 sources to retrieve historical data for THB/USD prediction.

3.1.1 Data Sources

1. **Yahoo Finance:** The Yahoo Finance API was a widely used service that allowed developers to retrieve financial data from Yahoo Finance, such as stock quotes, historical prices, and company information. In the study, this API is mainly for currency exchange rates, and stock prices as follow:
    a. Currency Exchange Rates including THB/USD, CNY/USD, JPY/USD, EUR/USD, CAD/USD, AUD/USD, SGD/USD, KRW/USD, GBP/USD, CHF/USD, ZAR/USD, and NZD/USD
    b. Stock Prices including SET Index:
        - SET (Stock Exchange of Thailand) Index
        - NASDAQ: NASDAQ (National Association of Securities Dealers Automated Quotations)
        - SSE Composite Index: SSE (Shanghai Stock Exchange) Composite Index
        - HSI: HSI (Hang Seng Index)
        - London Stock: London Stock Exchange (LSE)
        - Euronext: a pan-European stock exchange that operates multiple exchanges across several European countries, TSX stock: TSX (Toronto Stock Exchange)
        - DB1.DE: Deutsche Börse AG
        - Nikkei 225: the main stock market index for the Tokyo Stock Exchange in Japan
        - Tadawul: the stock exchange of Saudi Arabia
        - DAX: The DAX (Deutscher Aktienindex)
        - KOSPI: KOSPI (Korea Composite Stock Price Index)
        - ASX 200: the main stock market index of the Australian Securities Exchange (ASX)

- TAIEX: TAIEX (Taiwan Stock Exchange Capitalization Weighted Stock Index)

c. Other Features: Crude Oil, Gold, and Silver in USD

The features above are adaptive by the studies of Chalermrat (2022), Dash (2018), Shaolong Sun (2020), and Panda, Panda, & Pattnaik (2020) as mentioned in section 2.5.

2. **"holidays" package in Python:** This package offers a range of functionalities aimed at facilitating the handling of public holidays in various countries consisting of holidays in the U.S., the U.K., Australia, and Canada
3. **Other sources:** The acquisition and extraction of historical data through reliable web scraping methods are employed to incorporate additional features, such as Japanese, Chinese, and Thai holidays, which cannot be retrieved from the "holiday" package.

As a study from (Greenwood, 2006) shows that there is a relationship between the currency exchange rate and holidays. So, holidays and dates are used in the study.

All these historical data are transformed into Parquet files and act as input data preprocessing.

3.2 Data Preprocessing

Data preprocessing is a critical step in data analysis, involving techniques like cleaning, standardization, and transformation. Its purpose is to address quality issues and enhance accuracy before further analysis. By treating missing values, outliers, and inconsistencies, data preprocessing ensures data integrity and reliability for subsequent analytical tasks.

1. **Data Joining:** All datasets are loaded and joined with 'Date' which is a primary key of every single data frame.
2. **Data Cleansing:** To eliminate potentially misleading features, it is essential to rename the columns during the data preprocessing stage.

Additionally, as per the study's scope, the dataset is truncated to a specific period. Furthermore, in the context of the prediction task, any missing values in the THB/USD variable, which serves as the dependent variable, are removed from the dataset.

3. **Data Transformation:** The categorical data are transformed to binary values using one-hot encoding. In data transformation of the dataset from deep learning modeling, Min-Max Scaler is used for rescaling the value to the range between 0 and 1.

4. **Feature Extraction:** The process of manipulating data to derive features encompassing weekdays, months, week numbers, and weekend indicators from the "Date" variable holds significant importance in the context of model performance. This approach exhibits potency as certain derived features are likely to contribute positively to the efficacy of the model.

5. **Data Manipulation:** Due to deep learning being applied to the study, **timesteps** play a crucial role in deep learning for effectively handling sequential data by capturing temporal patterns and dependencies. They are vital in tasks such as sequence modeling, where the order of data points is significant. Timesteps facilitate the understanding of long-term dependencies in sequences. Additionally, timesteps allow for the incorporation of time-dependent features, enabling the model to explicitly consider the temporal aspect of the data. They provide a structured framework for training and prediction, ensuring sequential learning and generation of outputs. The study assigns timesteps at 30.

3.3 Data Exploration

Data exploration is a crucial step in the analysis, involving descriptive statistics, visualization, and profiling. It uncovers patterns, trends, and outliers, informing subsequent decisions. Thorough exploration enhances understanding, identifies relationships, and guides modeling choices, leading to valid and data-driven analysis.

1. **Statistical Indicators:** Statistical indicators, including standard deviation, minimum and maximum values, and median, serve as crucial tools for comprehending the underlying characteristics of data and its features. Standard deviation quantifies the degree of variability, illuminating the dispersion of data points around the mean. Minimum and maximum values demarcate the range within which data values are contained, enabling the identification of potential outliers or extreme observations. The median, representing the central tendency, provides a robust measure less susceptible to the influence of extreme values. These statistical indicators facilitate the examination of data distribution, anomaly detection, and data quality assessment. Consequently, analysts can make informed decisions and acquire a more profound understanding of the data's profile and attributes.
2. **Visualization:** a valuable technique in data analysis that involves creating graphs to visually understand data trends, relationships, and behaviors. Graphs such as lines, heatmaps, and histograms are used to explore data distribution, identify correlations, detect anomalies, and observe temporal patterns. Visualization aids in the effective communication of findings and supports decision-making by providing clear representations of data characteristics. It enables analysts to uncover insights that may be hidden in raw data and facilitates intuitive exploration and interpretation of data.

3.4 Feature Selection

A critical step in machine learning and data analysis involves identifying relevant features from a dataset. The goal is to reduce dimensionality by selecting a subset of features that significantly impact the target variable or contribute to model performance. Various techniques, including filter, wrapper, and embedded methods, are used for feature selection. It offers benefits such as improved model performance, reduced overfitting, enhanced interpretability, and faster training. Ultimately, feature selection enables efficient and accurate analysis, leading to improved machine learning models, and deep learning.

In the study, 2 statistical indicators are playing a main role to exclude less-important features as follows:

1. **Pearson Correlation:** The Pearson correlation coefficient is a statistical measure used to assess the strength and direction of the linear relationship between two continuous variables. It provides a numerical value between -1 and +1, where positive values indicate a direct relationship, negative values indicate an inverse relationship and a value of zero indicates no linear relationship. The benchmark for the Pearson correlation coefficient provides a framework for assessing the strength of the linear relationship between variables. It establishes guidelines based on specific coefficient ranges to determine the degree of correlation. Correlation coefficients within the range of 0 to 0.3 suggest a weak or negligible correlation, while coefficients ranging from 0.3 to 0.5 indicate a moderate correlation. Coefficients falling between 0.5 and 0.7 suggest a moderately strong correlation, whereas coefficients between 0.7 and 1 signify a strong correlation. It is imperative to consider the contextual nuances and statistical significance alongside these benchmarks to ensure an accurate interpretation of the association between variables. Most features in the range of 0.3 – 0.7 are selected while most others are excluded. However, it must be noted that this measure reflects only the linearity relationship among features.
2. **Tree-based model:** Tree-based models, such as XGBoost, Random Forest, and Decision Trees, are widely employed in academic research for feature selection due to their ability to estimate feature importance, capture non-linear relationships and interactions, handle missing data, exhibit robustness to outliers, and offer interpretability. These advantages make tree-based models a popular choice for selecting relevant features and enhancing model performance in academic settings.

In the study, the choice between XGBoost and LightGBM for feature selection depends on factors such as maturity, popularity, flexibility in handling custom objectives, interpretability, handling missing values, and the existing ecosystem. XGBoost has the advantage of being a more established and widely adopted model with extensive community support. It offers greater flexibility in handling custom objectives, provides better interpretability through visualization and understanding of decision-making, has built-in capabilities to handle missing values, and benefits from a well-developed ecosystem. These factors make XGBoost a preferred choice for feature selection in academic research.

After using the initial metrics, the selected features are then applied in model training. However, in order to enhance runtime and reduce model complexity, XGBoost, a tree-based model, is utilized to rearrange the feature importance. Only the top 10 features with the highest importance are adopted.

3.5 Training Models

A training model in the field of machine learning and deep learning acquires patterns and insights from data through a dedicated training procedure. Through iterative refinement of its internal parameters using optimization algorithms and a designated loss function, the model endeavors to minimize the disparity between anticipated outputs and the true target values. The effectiveness and diversity of the training data play a pivotal role in shaping the model's behavior and performance. Following successful training, the model gains proficiency in making predictions or executing tasks on previously unseen data instances.

1. **Train-Test Split:** In the process of training models, the input data is typically divided into two distinct sets: the training dataset and the test dataset. The training dataset encompasses data recorded between April 1, 2018, and June 30, 2022 (approximately 90% of total records), while the test dataset comprises data captured between July 1, 2022, and December 31, 2022 (approximately 10% of total records). However, in

this study, the test dataset serves as an out-of-time validation, implying that the prediction necessitates a sequence of historical data within a specific period. The reason why the study uses 90-10 in train-test split is that is a fluctuation in currency exchange market. The fewer training records, the more unpredictable information.

Table 3.1 Train-Test split period

Dataset	From	To	Records
Train	April 1, 2018	June 30, 2022	1,108
Test (Out-of-time)	July 1, 2022	December 31, 2022	131

2. **Optimization:** This research employs Grid Search as a method to identify the most optimal hyperparameter combination for the models under consideration. Grid Search demonstrates superior accuracy compared to Random Search when working with small-scale datasets. Notably, a previous study conducted by (Paliari, Karanikola, & Kotsiantis, 2021) investigates the effectiveness of Grid Search, Random Search, and Genetic Algorithm in neural architecture search (NAS), concluding that **Grid Search** achieves the best results by exhaustively exploring a specific subset of hyperparameter space. Conversely, in scenarios involving large datasets, Random Search is a more efficient approach, although it does not guarantee the attainment of the best outcome. In this study, Grid Search is employed to determine the optimal hyperparameters in each model for the best outcome.

3.6 Model Evaluation

Model evaluation in the context of time series analysis encompasses the comprehensive assessment of the predictive performance of models specifically designed to handle time-dependent data. This evaluation process entails dividing the time series dataset into distinct training and validation periods. The out-of-sample testing is performed to evaluate the model's ability to generalize to unseen data beyond the validation period. The model is then trained using historical data up to a particular point, after which its predictive capability is evaluated using various performance metrics such as MSE, MAE, and RMSE. Additionally, the model's performance is compared against benchmark models or established baseline methods to gauge its relative effectiveness. Visual analysis of the model's predictions and residual patterns is also conducted to provide qualitative insights. Statistical test as the Population Stability Index (PSI) is utilized to evaluate changes in the distribution of a categorical variable between two datasets or over time. It provides a quantitative assessment of the variability within the variable. This comprehensive evaluation process aids researchers and practitioners in assessing the accuracy, robustness, and potential areas for enhancement of time series models in forecasting and prediction tasks.

3.7 Model interpretation

The model explanation involves interpreting and understanding the factors and relationships utilized by a machine learning model to make predictions. It aims to enhance transparency and trust by providing insights into the model's decision-making process. Common techniques include:

1. **Feature Importance:** To assess the impact of individual input features on the model's output. It quantifies the contribution of each feature in influencing predictions. This technique often involves calculating permutation importance, SHAP values, or using feature importance scores derived from tree-based models. These approaches are widely utilized in determining the significance of input features in the model's overall performance.
   a. **Feature importance from Gini Importance:** The feature importance can be obtained from the feature_importances_

attribute of a fitted model. This importance is calculated by considering the average accumulation of impurity decrease within each tree, along with their standard deviation. On the other hand, Gini Importance or Mean Decrease in Impurity (MDI) measures feature importance by summing the number of splits involving a feature across all trees, considering the proportion of samples affected by those splits. (Gupta M. , 2022) This index can be obtained from scikit-learn library in python to assess the tree-based models in the study.

b. **Permutation feature importance:** a methodology employed to assess the significance of individual features within a predictive model. It gauges the extent to which the model's prediction error increases when the values of a particular feature are randomly permuted, thus disrupting its relationship with the true outcome. The algorithm encompasses several steps, including the estimation of the original model error, the generation of permuted feature matrices, and the calculation of the resulting error. Subsequently, the features are ranked in descending order based on their permutation feature importance. (Molnar, 2023)

In the study, this approach is applied for deep learning models because time series prediction with time steps does not have finished function from famous libraries, such as scikit-learn and Keras, support.

2. **Data Drift:** a methodology used to detect changes in data distributions over time. It involves comparing new data with a reference dataset to identify significant deviations. By detecting data drift, this approach helps uncover issues such as concept drift or dataset bias. Various techniques, including statistical tests and machine learning models, are utilized to quantify and detect data drift. The goal is to maintain the

accuracy and reliability of machine learning models by adapting to changes in the data.

a. **Population Stability Index (PSI):** a metric used to assess the consistency and stability of a variable's distribution across different populations or time periods. It is commonly applied in fields such as credit risk modeling and fraud detection to evaluate the reliability of predictive models and monitor changes in population characteristics. By comparing expected and observed frequencies of the variable, PSI provides insights into model performance and data quality. (Agarwal, 2022) If the PSI is below 0.1, no change is observed, and the existing model can be used. For PSI values between 0.1 and 0.2, slight adjustments are recommended. However, if the PSI exceeds 0.2, significant changes are necessary, and it is advisable to recalibrate or develop a new model. This ensures that the model remains accurate and aligned with the changing data patterns. (Ramzai, 2020)
b. **Characteristic Stability Index (CSI):** a quantitative measure employed to assess the shifts in population distribution and identify the variables accountable for these changes. By comparing the distribution of an independent variable in a scoring dataset to that in a development dataset, CSI enables the detection of variations in the distributions of input variables over a specific period. The calculation procedure of CSI closely aligns with that of PSI, wherein the development sample values are employed to establish cutoffs for binning and subsequently compute frequency values in validation or out-of-time samples. (Ramzai, 2020)

CHAPTER 4
RESULTS AND ANALYSIS

This chapter shows the models' performance, compares the results, and analyzes the outcomes following the objectives of the study. Additionally, data drift is used to explain the profile of the train and test dataset.

4.1 Category of Models in the study

Table 4.1 Category of models in the study separated by Target

No.	Target	Model	Model Name
1	THB_CLOSE	XGBoost	xgb_1
2	THB_CLOSE	LightGBM	lght_gbm_1
3	THB_CLOSE	LSTM	lstm_1
4	THB_CLOSE	GRU	gru_1
5	THB_CLOSE	LSTM-LSTM (Stacked Model)	lstm_lstm_1
6	THB_CLOSE	LTSM-GRU (Stacked Model)	ltsm_gru_1
7	THB_CLOSE	GRU-GRU (Stacked Model)	gru_gru_1
8	CHANGE	XGBoost	xgb_2
9	CHANGE	LightGBM	lght_gbm_2
10	CHANGE	LSTM	lstm_2
11	CHANGE	GRU	gru_2
12	CHANGE	LSTM-LSTM (Stacked Model)	lstm_lstm_2
13	CHANGE	LTSM-GRU (Stacked Model)	ltsm_gru_2
14	CHANGE	GRU-GRU (Stacked Model)	gru_gru_2

This study encompasses four techniques of tree-based machine learning and deep learning, namely XGBoost, LightGBM, LSTM, and GRU. These techniques are employed to predict the Thai baht (THB) and focus on two specific targets: the precise value of THB/USD (THB_CLOSE) and the difference (CHANGE) between THB_CLOSE and the 30-day moving average of THB_CLOSE (THB_CLOSE_MA30). The summary of model training is shown in Table 4.1.

4.2 Feature Selection

After utilizing statistical indicators such as Pearson Correlation and T-test, the remaining features include: 'THB_OPEN', 'THB_HIGH', 'THB_LOW', 'AUD_CLOSE', 'CRUDE_OIL_CLOSE', 'DB1_DE_CLOS', 'EUR_CLOSE', 'GBP_CLOSE', 'HSI_CLOSE', 'JPY_CLOSE', 'KRW_CLOSE', 'NIKKEI_225_CLOSE', 'NZD_CLOSE', 'SET_CLOSE', 'SGD_CLOSE', 'TPE_TAIEX_CLOSE_1D', 'ZAR_CLOSE_1D', 'US_HOLIDAY', 'UK_HOLIDAY', 'AUS_HOLIDAY', 'CA_HOLIDAY', 'JP_HOLIDAY', 'TH_HOLIDAY', 'CN_HOLIDAY', and 'BIG_HOLIDAY'. Moreover, each feature has undergone the feature engineering process to determine its values in the previous 1, 4, 14, and 30 days. Additionally, the 'DATE' has been extracted into weekday, month, and week number. (See more information in Appendix A)

Next, to improve the runtime in Grid Search process, we use XGBoost as a baseline model to choose only 10 features which provide the highest impact descending. The results are as follow:

THB_CLOSE: 'WK_YR_53', 'THB_OPEN_30D', 'THB_HIGH_30D', 'HSI_CLOSE_30D','CRUDE_OIL_CLOSE_1D', 'NZD_CLOSE_14D', 'CRUDE_OIL_CLOSE_14D', 'JPY_CLOSE_1D',

CHANGE: 'SGD_CLOSE_30D', 'AUD_CLOSE_30D', 'THB_OPEN_30D', 'UK_HOLIDAY', 'THB_HIGH_1D', 'THB_OPEN_1D', 'DB1_DE_CLOSE_30D', 'SGD_CLOSE_1D', 'MONTH_3', 'NZD_CLOSE_1D'

However, there are 9 features remaining in the prediction using “CHANGE” after using XGBoost for the feature selection while the prediction using “THB_CLOSE” remains 10 features.

4.3 Result in Modeling Training

Table 4.2 Model Training Summary

NO.	Target	Model Name	Best Parameters	RMSE (10^{-5})
1	THB_CLOSE	xgb_1	reg_alpha=0, reg_lambda=1, learning_rate=0.1, max_depth=5, min_child_weight=1, subsample=0.5, n_estimators=500	26.7206
2	THB_CLOSE	lgbm_1	learning_rate = 0.07, reg_alpha = 0, reg_lambda =: 1, ' max_depth = 3, subsample = 0.5, n_estimators = 300, num_leaves = 21, feature_fraction = 0.5, colsample_bytree = 0.5, bagging_fraction = 0.5	17.4058
3	THB_CLOSE	lstm_1	'batch_size': 32, 'dropout_rate': 0.4, 'epochs': 100, 'units': 64	86.7595
4	THB_CLOSE	gru_1	'batch_size': 64, 'dropout_rate': 0.4, 'epochs': 100, 'units': 64	181.3838
5	THB_CLOSE	lstm_lstm_1	'batch_size': 32, 'dropout_rate': 0.3, 'epochs': 200, 'units_layer_1': 64, 'units_layer_2': 64	76.5078
6	THB_CLOSE	lstm_gru_1	'batch_size': 32, 'dropout_rate': 0.2, 'epochs': 200, 'units_layer_1': 64, 'units_layer_2': 64	63.1602

Table 4.2 Model Training Summary (Cont.)

No.	Target	Model Name	Best Parameters	RMSE (10^{-5})
7	THB_CLOSE	gru_gru_1	'batch_size': 32, 'dropout_rate': 0.3, 'epochs': 100, 'units_layer_1': 128, 'units_layer_2': 64	82.0067
8	CHANGE	xgb_2	reg_alpha=0, reg_lambda=0, learning_rate=0.1, max_depth=5, min_child_weight=1, subsample=0.5, n_estimators=500	18.9829
9	CHANGE	lgbm_2	reg_alpha = 0, reg_lambda = 1, learning_rate = 0.1, max_depth = 5, subsample = 0.5, n_estimators = 150, num_leaves = 31, feature_fraction = 0.5, colsample_bytree = 0.5, bagging_fraction = 0.5	7.843
10	CHANGE	lstm_2	'batch_size': 32, 'dropout_rate': 0.2, 'epochs': 200, 'units': 32	144.6408
11	CHANGE	gru_2	'batch_size': 32, 'dropout_rate': 0.3, 'epochs': 200, 'units': 32	166.3595
12	CHANGE	lstm_lstm_2	'batch_size': 32, 'dropout_rate': 0.4, 'epochs': 200, 'units_layer_1': 64, 'units_layer_2': 32	40.7018

Table 4.2 Model Training Summary (Cont.)

No.	Target	Model Name	Best Parameters	RMSE (10^{-5})
13	CHANGE	lstm_gru_2	'batch_size': 32, 'dropout_rate': 0.4, 'epochs': 200, 'units_layer_1': 64, 'units_layer_2': 32	55.7342
14	CHANGE	gru_gru_2	'batch_size': 64, 'dropout_rate': 0.3, 'epochs': 100, 'units_layer_1': 64, 'units_layer_2': 64	94.6257

Table 4.3 MSE, MAE and RMSE with train dataset

Model Name	MSE (10^{-5})	MAE (10^{-5})	RMSE (10^{-5})
xgb_1	0.0071	20.6661	26.7206
lgbm_1	0.003	13.3073	17.4058
lstm_1	0.0753	66.3783	86.7595
gru_1	0.329	126.3595	181.3838
lstm_lstm_1	0.0585	62.3202	76.5078
lstm_gru_1	0.0399	51.0237	63.1602
gru_gru_1	0.0673	66.1614	82.0067
xgb_2	0.0036	14.6135	18.9829
lgbm_2	0.0006	5.8859	7.843
lstm_2	0.2092	134.4386	144.6408
gru_2	0.2768	142.0833	166.3595
lstm_lstm_2	0.0166	32.9191	40.7018
lstm_gru_2	0.0311	46.0276	55.7342
gru_gru_2	0.0895	84.3833	94.6257

Table 4.2 shows how each model was fine-tuned using Grid Search including its best parameters with the lowest RMSE.

The results of training the models reveal that those utilizing tree-based algorithms and stacked deep learning can effectively capture the trend of THB/USD, as indicated by the black line in Figure 4.1. In contrast, models employing vanilla deep learning techniques exhibit highly fluctuating trend lines. These outcomes have a direct impact on the errors measured by the metrics presented in Table 4.3.

However, it is important to note that this step alone does not determine which model performs the best. The errors observed in the training dataset indicate the chance for overfitting in each model. A model that shows no errors may indicate an excessive fit to the training dataset, leading to challenges when presented with unseen data.

4.4 Model Evaluation

To ensure the performance of these models throughout different periods, a test dataset (out-of-time) is crucial for confirming their effectiveness. The result is shown in Table 4.3.

However, this study mainly focused on **RMSE** rather than **MAE** because (JJ, 2016) the choice between RMSE and MAE as evaluation metrics depends on the specific use case and dataset. However, in general, RMSE is typically preferred over MAE for measuring model performance. RMSE places more emphasis on larger errors, making it more effective in identifying and addressing outliers in predictions. The evaluation contains three parts including Target Manipulation, Model algorithms, and Overview of the top 5 models.

1. **Target Manipulation**

   According to Table 4.4 and Table 4.5, a comparison was made between two variables - THB_CLOSE, which represents the actual THB/USD value, and CHANGE, denoting the difference between THB_CLOSE and its 30-day moving average price. The analysis reveals that predictions based on CHANGE consistently demonstrate better performance in terms of RMSE (Root Mean Squared Error) compared to predictions based on THB_CLOSE. Moreover, when comparing the same algorithm using RMSE, CHANGE consistently outperforms THB_CLOSE across all models. It is also noteworthy that all five models with the lowest RMSE utilize CHANGE as the target variable. The trend lines of each model are shown in Figure 4.1 and 4.2.

In summary, utilizing the CHANGE technique for prediction yields more efficient outcomes. This is attributed to the fact that this technique effectively reduces the fluctuations in the THB/USD trend over time.

2. Model Algorithms

The evaluation criterion employed in this comparison is an average of RMSE (Root Mean Squared Error) in each technique. The result is shown in Table 4.6. The results of this study indicate that among the three techniques considered, namely Tree-based, Plain Deep Learning, and Ensemble Deep Learning, the Tree-based algorithm emerges as the most effective predictor for THB_CLOSE. Furthermore, both Ensemble Deep Learning and Tree-based Algorithms exhibit superior performance compared to Plain Deep Learning. It is confirmed that Ensemble Deep Learning is more powerful than the plain ones.

Next, the five models with the lowest RMSE values are selected for further explanation. These models include lstm_lstm_2, gru_gru_2, lgbm_2, xgb_2, and lstm_gru_2, as presented in Table 4.5. Observing the results, it is evident that both the Tree-based model and Ensemble Deep Learning approach exhibit relatively similar RMSE values, indicating comparable performance.

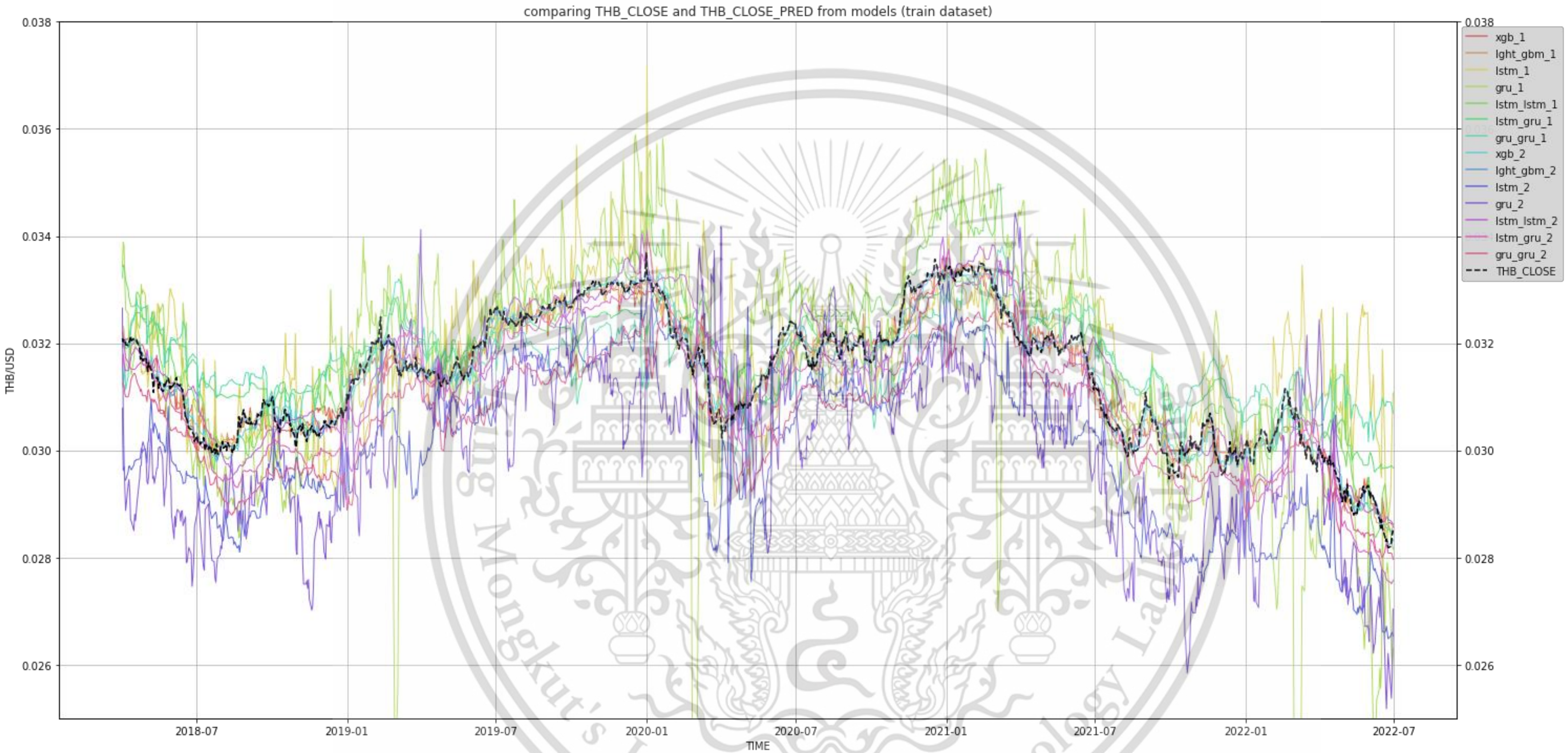

Figure 4.1 Predicted THB/USD from all model's and Actual THB/USD (train dataset)

Table 4.4 MSE, MAE and RMSE with test (out-of-time) dataset comparing between targets

Target	Model Name	MSE (10^{-5})	MAE (10^{-5})	RMSE (10^{-5})
THB_CLOSE	gru_1	4.8319	588.8810	695.1171
CHANGE	gru_2	1.0668	281.7904	326.6186
THB_CLOSE	gru_gru_1	0.9424	295.8747	306.9905
CHANGE	gru_gru_2	0.0481	56.2303	69.3406
THB_CLOSE	lgbm_1	0.3277	159.1001	181.0309
CHANGE	lgbm_2	0.0485	54.5091	69.6669
THB_CLOSE	lstm_1	2.6370	450.0564	513.5154
CHANGE	lstm_2	0.5580	169.1332	236.2101
THB_CLOSE	lstm_gru_1	1.3424	350.7087	366.3939
CHANGE	lstm_gru_2	0.0560	63.7562	74.8327
THB_CLOSE	lstm_lstm_1	2.3246	408.3642	482.1364
CHANGE	lstm_lstm_2	0.0465	57.7205	68.1661
THB_CLOSE	xgb_1	0.2178	125.4625	147.5808
CHANGE	xgb_2	0.0502	55.0936	70.8278

Table 4.5 MSE, MAE and RMSE with test (out-of-time) dataset

Target	Model Name	MSE (10^{-5})	MAE (10^{-5})	RMSE (10^{-5})	Ranking
CHANGE	lstm_lstm_2	0.0465	57.7205	68.1661	1
CHANGE	gru_gru_2	0.0481	56.2303	69.3406	2
CHANGE	lgbm_2	0.0485	54.5091	69.6669	3
CHANGE	xgb_2	0.0502	55.0936	70.8278	4
CHANGE	lstm_gru_2	0.0560	63.7562	74.8327	5
THB_CLOSE	xgb_1	0.2178	125.4625	147.5808	6

Table 4.5 MSE, MAE and RMSE with test (out-of-time) dataset (Cont.)

Target	Model Name	MSE (10^{-5})	MAE (10^{-5})	RMSE (10^{-5})	Ranking
THB_CLOSE	lgbm_1	0.3277	159.1001	181.0309	7
CHANGE	lstm_2	0.5580	169.1332	236.2101	8
THB_CLOSE	gru_gru_1	0.9424	295.8747	306.9905	9
CHANGE	gru_2	1.0668	281.7904	326.6186	10
THB_CLOSE	lstm_gru_1	1.3424	350.7087	366.3939	11
THB_CLOSE	lstm_lstm_1	2.3246	408.3642	482.1364	12
THB_CLOSE	lstm_1	2.6370	450.0564	513.5154	13
THB_CLOSE	gru_1	4.8319	588.8810	695.1171	14

Table 4.6 The comparison among different algorithms separated by targets

Target	Algorithm	Model Name	MSE(10^{-5})	MAE(10^{-5})	RMSE(10^{-5})	AVG. RMSE(10^{-5})
CHANGE	Deep Learning	gru_2	1.0668	281.7904	326.6186	326.6186
		lstm_2	0.5580	169.1332	236.2101	
	Ensemble Deep Learning	gru_gru_2	0.0481	56.2303	69.3406	70.7798
		lstm_gru_2	0.0560	63.7562	74.8327	
		lstm_lstm_2	0.0465	57.7205	68.1661	
	Tree-Based	lgbm_2	0.0485	54.5091	69.6669	70.2474
		xgb_2	0.0502	55.0936	70.8278	
THB_CLOSE	Deep Learning	gru_1	4.8319	588.8810	695.1171	695.1171
		lstm_1	2.6370	450.0564	513.5154	
	Ensemble Deep Learning	gru_gru_1	0.9424	295.8747	306.9905	385.1736
		lstm_gru_1	1.3424	350.7087	366.3939	
		lstm_lstm_1	2.3246	408.3642	482.1364	
	Tree-Based	lgbm_1	0.3277	159.1001	181.0309	164.3059
		xgb_1	0.2178	125.4625	147.5808	

Though, upon analyzing the trends of the top 5 models, as depicted in Figure 4.4, it becomes apparent that Ensemble Deep Learning models are more adept at capturing the upward and downward movements of THB/USD compared to the Tree-Based Algorithm. It is crucial to note that while RMSE and MAE provide information about the magnitude of errors in the predictions, they do not offer detailed insights into the trends of the predicted and actual values.

The lstm_lstm_2 model stands out among other models due to its architecture, which is the most complex compared to GRU and other Tree-based algorithms. LSTM incorporates a cell state that allows the network to retain and propagate information over time. The cell state is updated through input, forget, and output gates, determining which information to include, discard, or output from the cell state. This mechanism enables LSTMs to effectively capture long-term dependencies in sequential data. Additionally, stacked deep learning models outperform plain models for various reasons. Stacked models leverage the strengths of multiple individual models, resulting in increased model diversity, reduced error correlation, improved resistance to overfitting, enhanced decision-making through adaptive weighting, and greater flexibility and adaptability. These factors collectively contribute to the superior performance of stacked models in capturing complex patterns and achieving accurate predictions when compared to plain models.

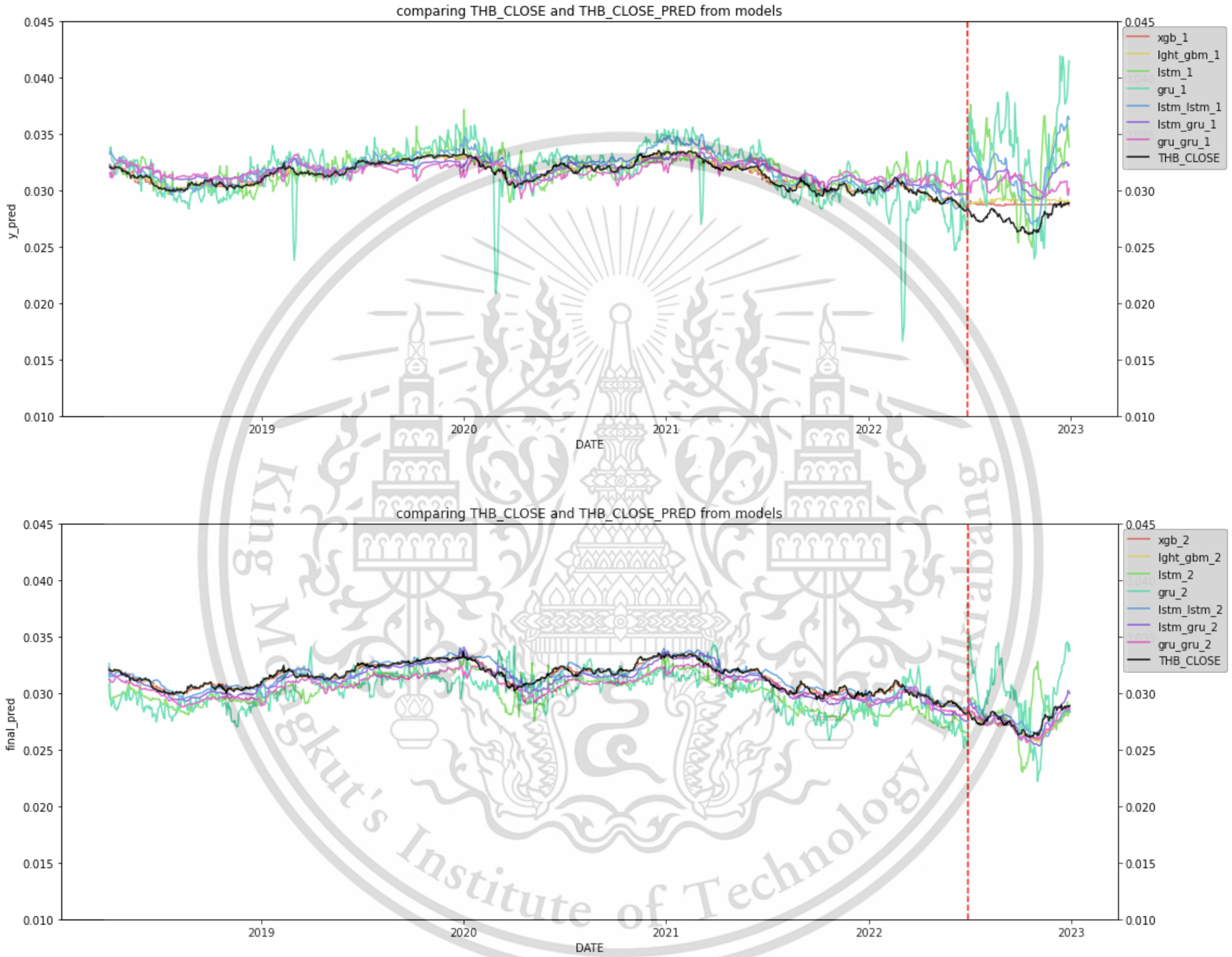

Figure 4.2 Predicted THB/USD from all model's and Actual THB/USD)

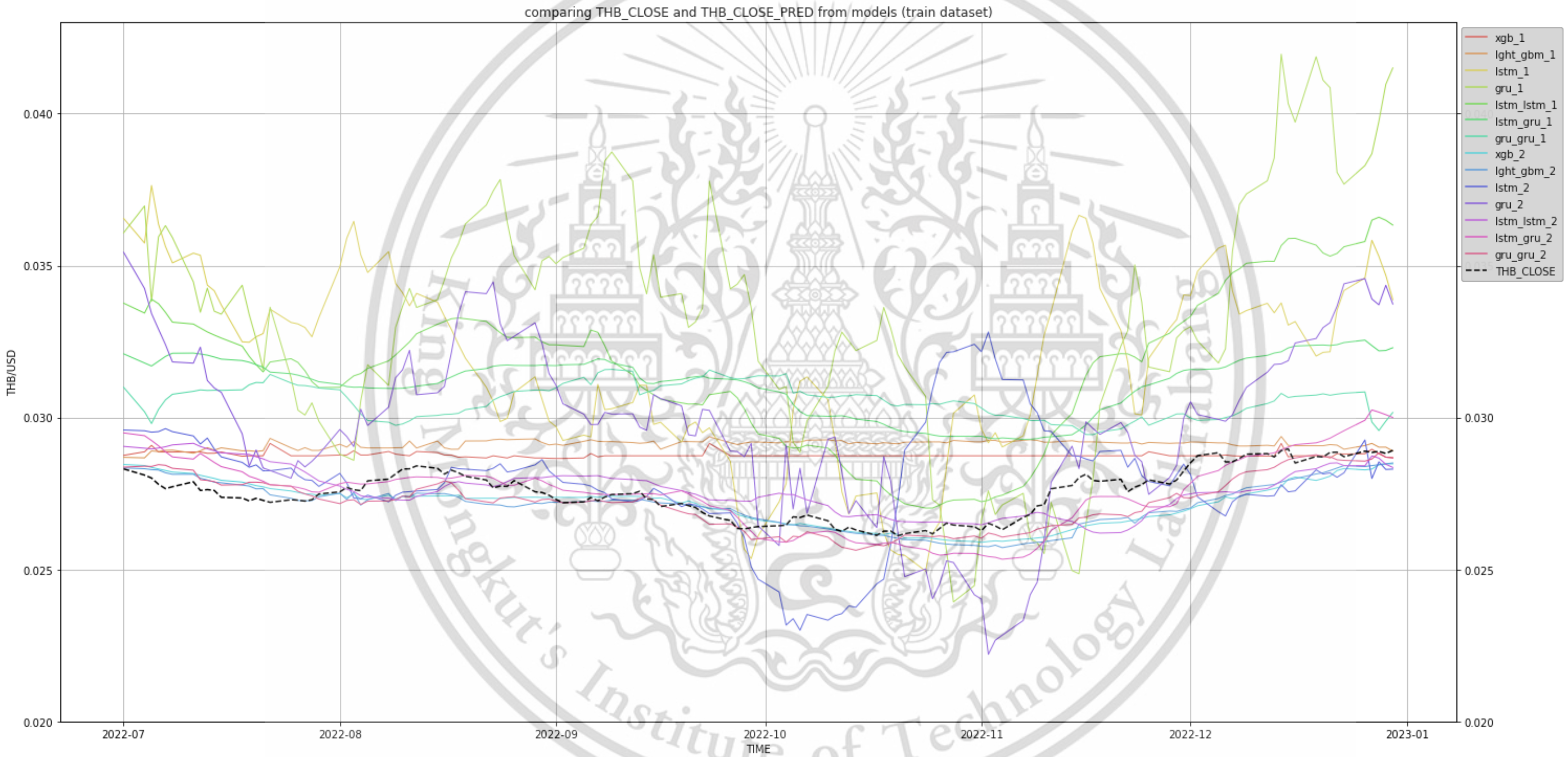

Figure 4.3 Predicted THB/USD from all models and Actual THB/USD (Out of Time dataset)

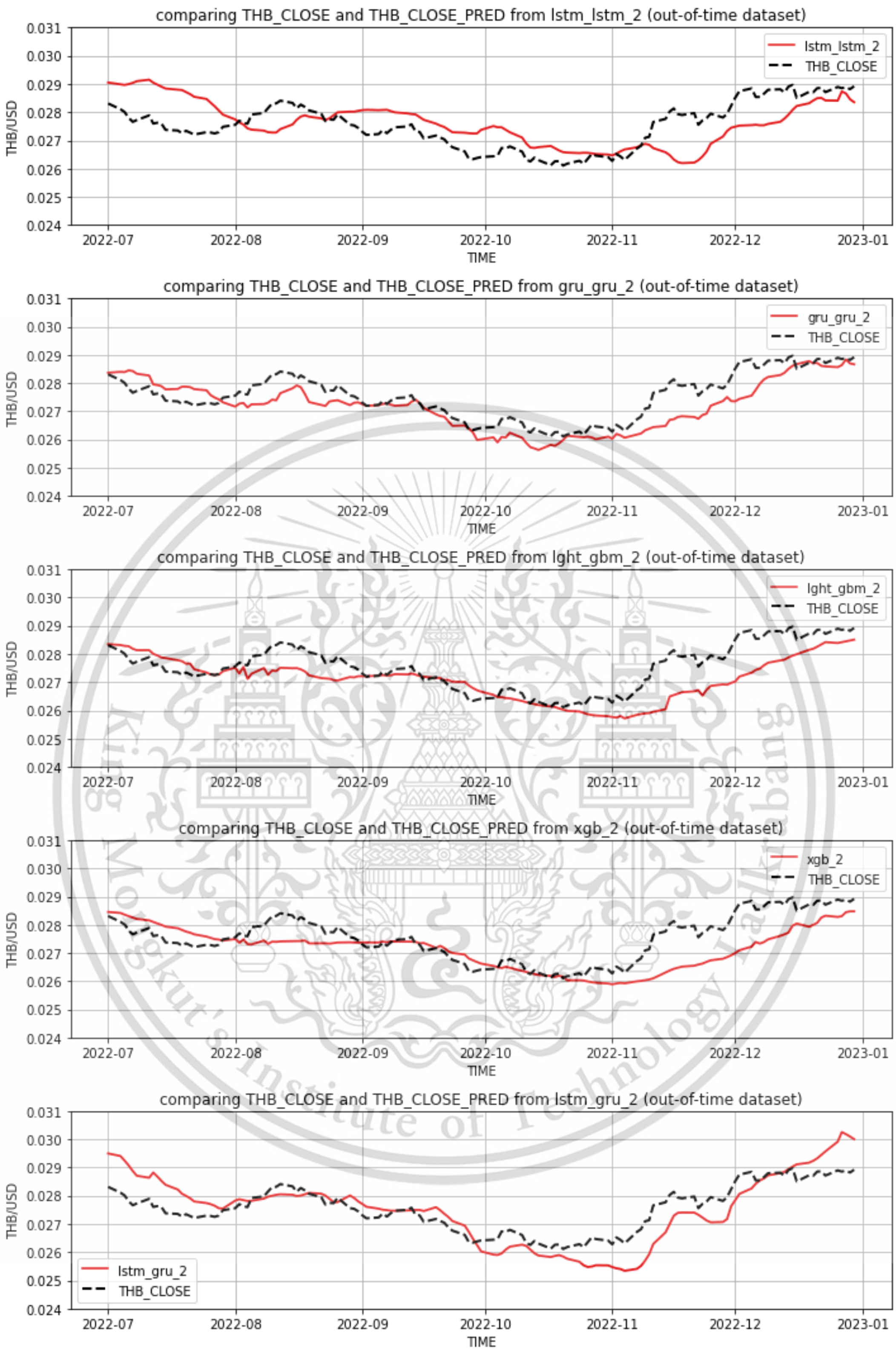

Figure 4.4 Predicted THB/USD from TOP 5 models' and Actual THB/USD (Out of Time dataset)

4.5 Feature Importance

In this section, the explanation focuses on the top 5 models with the top 10 features of each model presented in Table 4.5. For more detailed information on all models, refer to Appendix.

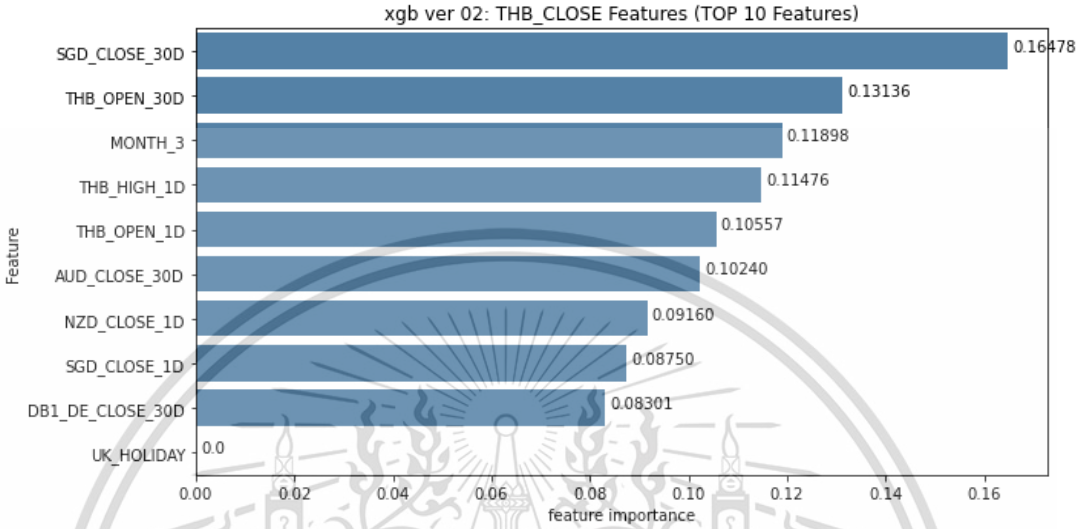

Figure 4.5 Feature Importance xgb_2

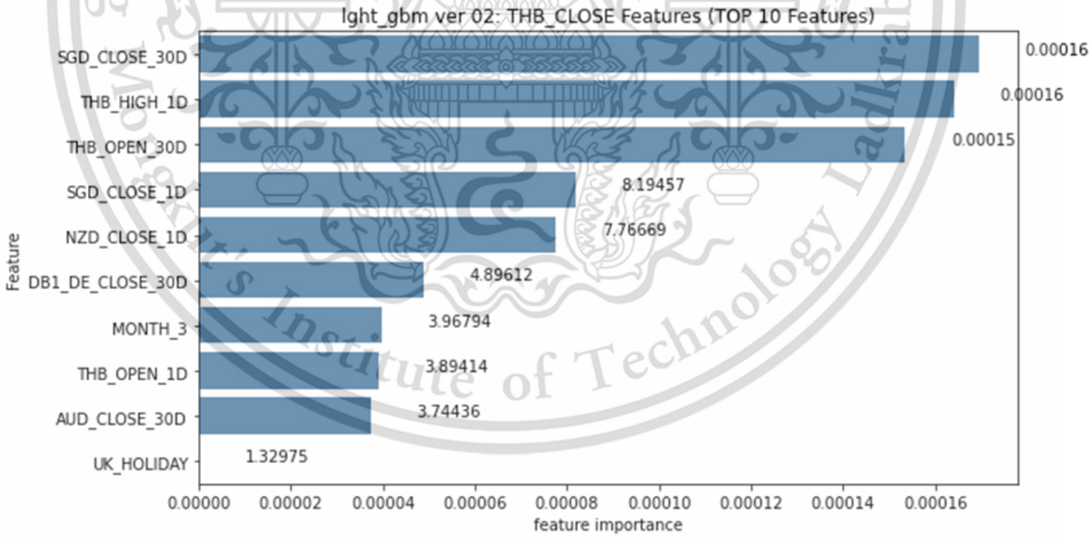

Figure 4.6 Feature Importance lght_gbm_2

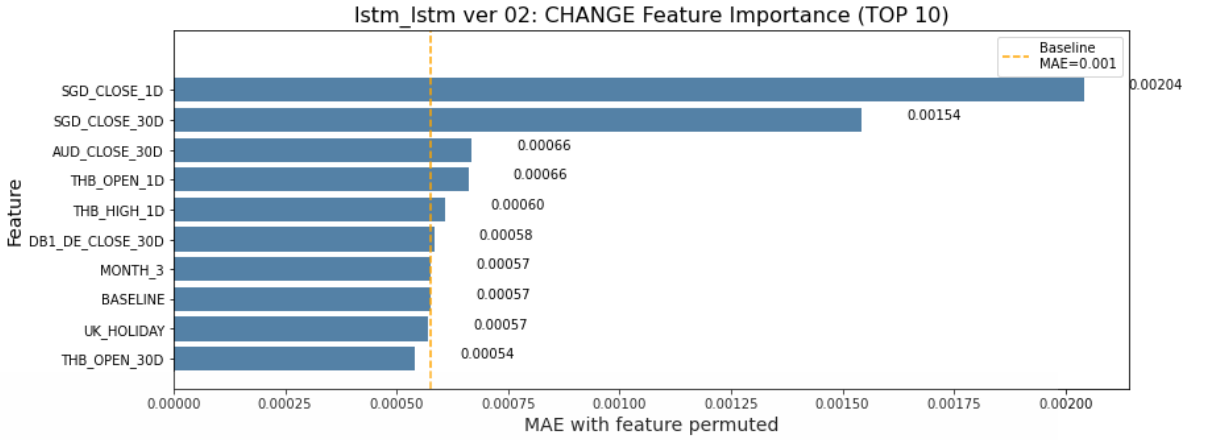

Figure 4.7 Feature Importance lstm_lstm_2

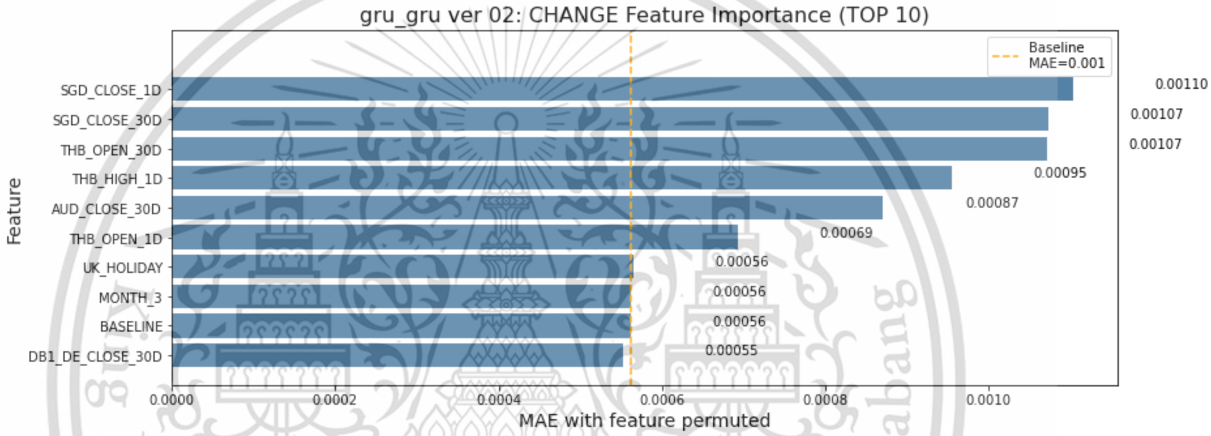

Figure 4.8 Feature Importance gru_gru_2

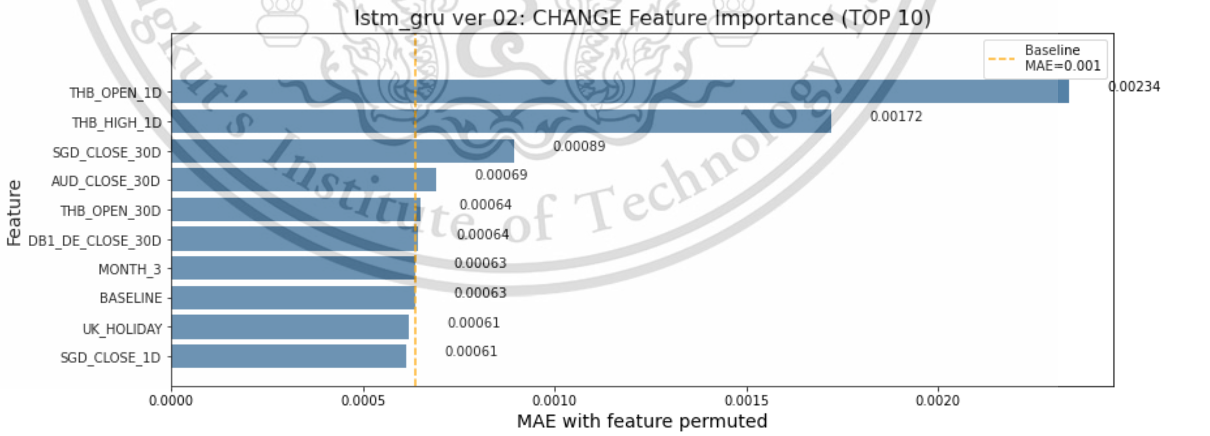

Figure 4.9 Feature Importance lstm_gru_2

In this assessment, the feature importance is obtained using Gini Importance for Tree-Based Models shown in Figure 4.5, and Figure 4.6. However, considering the complexity of deep learning models with timesteps, permutation feature importance emerges as a prominent method to evaluate the significance of each feature in deep learning shown in Figure 4.7, Figure 4.8, and Figure 4.9. The rankings of feature importance vary considerably among the top 5 models. However, in four of the five models, namely SGD_CLOSE1D and SGD_CLOSE30D, SGD/USD-related features, emerge as the most important features, except for lstm_gru_2. This finding suggests a strong relationship between THB/USD and SGD/USD. Interestingly, the UK_holiday feature exhibits significant influence on THB/USD, and it is also noteworthy that the MONTH_3 feature, representing March, dominates THB/USD.

4.6 Data Drift

To investigate the reasons behind the reduced predictability of THB/USD predictions from multiple models in the study beyond the training period, the Population Stability Index (PSI) and Characteristic Stability Index (CSI) are utilized. These indices are employed to evaluate the consistency and stability of the distribution of a variable across different populations or time periods.

4.6.1 Population Stability Index (PSI)

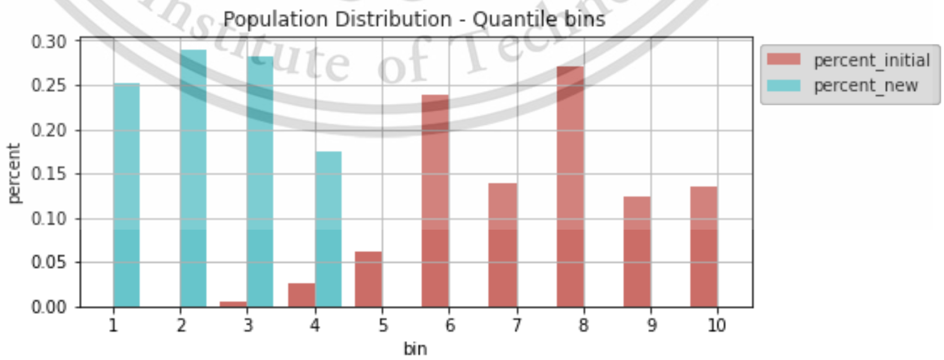

Figure 4.10 Population Distribution of THB_CLOSE (train and out-of-time datasets)

Table 4.7 PSI score of THB_CLOSE comparison between train and out-of-time dataset

bin	initial	percent_initial	new	percent_new	psi
1	0	0.0001	33.0000	0.2519	1.9721
2	0	0.0001	38.0000	0.2901	2.3119
3	6	0.0054	37.0000	0.2824	1.0954
4	29	0.0262	23.0000	0.1756	0.2844
5	69	0.0623	0.0000	0.0001	0.4000
6	264	0.2383	0.0000	0.0001	1.8520
7	154	0.1390	0.0000	0.0001	1.0051
8	299	0.2699	0.0000	0.0001	2.1312
9	138	0.1245	0.0000	0.0001	0.8870
10	149	0.1345	0.0000	0.0001	0.9680

To evaluate the stability of the target variable, THB_CLOSE, in this study, the Population Stability Index (PSI) is applied. The results are presented in Figure 4.10 and Table 4.7. A PSI value below 0.1 indicates the absence of substantial changes, suggesting that the existing model can be retained without modifications. However, if the PSI value falls within the range of 0.1 to 0.2, it suggests the necessity of slight adjustments to ensure the ongoing effectiveness of the model. There are many changes in bin numbers and the changes are shown in figure 4.3. However, an average PSI is 1.2907, It is advisable to thoroughly assess the consequences of these changes and their potential effects on the model's performance.

4.6.2 Characteristic Stability Index (CSI)

Table 4.8 CSI Score

feature	csi_score
THB_OPEN_30D	1.37508
THB_HIGH_1D	1.32684
THB_OPEN_1D	1.29151
NZD_CLOSE_1D	0.77398
DB1_DE_CLOSE_30D	0.73993
SGD_CLOSE_30D	0.57476
AUD_CLOSE_30D	0.35290
SGD_CLOSE_1D	0.21836

The errors in model prediction can be influenced by population drift and changing feature distribution when comparing the training dataset with out-of-time datasets. Table 4.8 reveals that all features undergo changes over time.

4.7 Discussion

The study emphasizes that training models using the "CHANGE" variable yield more powerful and accurate models compared to using the "THB_CLOSE" variable. Both Tree-based Algorithm and Ensemble Deep Learning outperform Plain Deep Learning in terms of lower RMSE. While tree-based models and ensemble deep learning demonstrate similar error rates, ensemble deep learning models excel in capturing the trends of the target variable.

The analysis of data drift reveals a significant shift in the population distribution of the target variable, THB_CLOSE, which has a considerable impact on prediction outcomes. Furthermore, the Population Stability Index (PSI) score and Characteristic Stability Index (CSI) indicate that the duration of the training dataset may be excessively long. Consequently, it is recommended to shorten the training period to improve the efficiency and accuracy of predictions on out-of-time datasets.

To compare this study and relevant research, as Chalermrat (2022) who studied in USD/THB prediction by LSTM and SARIMAX suggesting that using GRU is a challenging technique. The result from this study suggests that using GRU for both CHANGE and THB_CLOSE predictions provide a larger number of errors compared to the outcome of using LSTM. On the other hand, as Yunze Li (2022) who studied EUR/USD prediction shows the prediction using ensemble deep learning: LSTM-GRU provides the best performance. In the study, a combination of LSTM and GRU for CLOSE prediction, lstm_gru_2 is in the top 5 models which indicates that this ensemble deep learning also outperforms other models as it does in the study of Yunze Li (2022).

CHAPTER 5
CONCLUSION

This section aims to summarize the study's outcomes and provide insights on how future studies can enhance prediction performance and improve process efficiency.

5.1 Conclusion

The study provides evidence supporting the effectiveness of the CHANGE technique in improving prediction outcomes by reducing fluctuations in the THB/USD trend over time. Comparative analysis of the top 5 models, namely lstm_lstm_2, gru_gru_2, lgbm_2, xgb_2, and lstm_gru_2, reveals similar RMSE values for the Tree-based model and Ensemble Deep Learning approach, indicating comparable performance.

The architectural complexity of the lstm_lstm_2 model sets it apart from other models, as it incorporates a cell state that facilitates information retention and propagation over time. This characteristic enables Long Short-Term Memory (LSTM) models to capture long-term dependencies in sequential data. Stacked deep learning models, which combine multiple individual models, exhibit superior performance compared to plain models. This advantage stems from increased model diversity, reduced error correlation, improved resistance to overfitting, enhanced decision-making capabilities through adaptive weighting, and greater flexibility and adaptability.

Regarding feature importance, SGD_CLOSE1D and SGD_CLOSE30D consistently emerge as significant features in four of the five models, indicating a strong relationship between THB/USD and SGD/USD. The UK_holiday feature also demonstrates a notable influence on THB/USD, while the MONTH_3 feature (representing March) dominates THB/USD.

The stability of the target variable, THB_CLOSE, is evaluated using the Population Stability Index (PSI). PSI values below 0.1 indicate a lack of substantial changes, while values between 0.1 and 0.2 suggest the need for minor adjustments to maintain the effectiveness of the model. Notably, changes in bin numbers are observed, necessitating careful assessment of their potential impact on model performance.

The errors in model prediction can be affected by population drift and changes in feature distribution when comparing the training dataset with out-of-time datasets. Analysis of Table 4.8 reveals that all features undergo changes over time, emphasizing the need for consideration of temporal variations in predictive modeling.

5.2 Future Study

In future studies, it is suggested to avoid solely using Pearson's correlation as the sole criterion for selecting significant features. This measure primarily captures linear relationships among continuous features, potentially overlooking strong non-linear associations. It is suggested that Pearson's correlation is required to apply with other indicators such as Chi-square test, and ANOVA test. Additionally, incorporating multiple timesteps, such as 1, 15, 30, and 45, in the experimental design could help identify the optimal prediction model for THB/USD.

Furthermore, considering the differences in distribution between the training and test datasets, it is advisable to use shorter records that closely resemble the current distribution of the target variable and features. This would help ensure consistency and improve the accuracy of predictions.

REFERENCES

Agarwal, A. (2022, November 10). Population Stability Index (PSI). Retrieved from https://medium.com/model-monitoring-psi/population-stability-index-psi-ab133b0a5d42

Alom, M. (2021). Adam Optimization Algorithm. ResearchGate.

Amal Mahmoud, A. M. (2021). A Survey on Deep Learning for Time-Series Forecasting. ResearchGate.

Arbel, N. (2018, December 21). How LSTM networks solve the problem of vanishing gradients. Retrieved from Medium: https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577

Bushaev, V. (2018, October 22). Adam — latest trends in deep learning optimization. Retrieved from Medium: https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

Cao, W., Zhu, W., Wang, W., Demazeau, Y., & Zhang, C. (2020). A Deep Coupled LSTM Approach for USD/CNY Exchange Rate Forecasting. IEEE Xplore.

Chalermrat, N. (2022). Improving Time Series Forcasting Using Deep Learning. Thammast University.

Chollet, F. (2018). Deep Learning with Python. New York: Manning Publications Co.

Dash, R. (2018). Performance analysis of a higher order neural network withan improved shuffled frog leaping algorithm for currencyexchange rate prediction. Applied Soft Computing.

Feiping Nie, Z. H. (2018). An investigation for loss functions widely. Communications in Information and Systems, 2.

Filippou, I., Rapach, D. E., Taylor, M. P., & Zhou, G. (2021). Exchange Rate Prediction with Machine Learning.

Gang Wang, T. T. (2021). An improved ensemble learning method for exchange rate forecasting. Elsevier.

Greenwood, C. (2006). How do currency exchange rates influence the price of holidays? Journal of Revenue and Pricing Management.

Gupta, A. (2021, October 7). A Comprehensive Guide on Deep Learning Optimizers. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/#:~:text=The%20results%20of%20the%20Adam,for%20most%20of%20the%20applications.

Gupta, M. (2022, September 14). How feature importance is calculated in Decision Trees? with example. Retrieved from https://medium.com/data-science-in-your-pocket/how-feature-importance-is-calculated-in-decision-trees-with-example-699dc13fc078

Hansika Hewamalage, C. B. (2020). Recurrent Neural Networks for Time Series Forecasting: Current Status. Elsevier.

Investing in Thai Baht. (n.d.). Retrieved from Thai Residential: https://thairesidential.com/an-overview-on-the-thai-baht/

Isaac Kofi Nti, ,. A. (2020). A comprehensive evaluation of ensemble. Springer Open.

Jabeur, S. B., Mefteh-Wali, S., & Viviani, J.-L. (2021). Forecasting gold price with the XGBoost algorithm and SHAP. Springer Nature.

Jimeng Shi, M. J. (2020). Time Series Forecasting (TSF) Using Various. Florida: Florida International University.

JJ. (2016, March 23). MAE and RMSE — Which Metric is Better? Retrieved from https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d

Kingma, D. P., & Ba, J. L. (2015). ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. ICLR.

Kingploy. (5 พฤษภาคม 2565). เฟดขึ้นดอกเบี้ย 0.5% สูงสุดในรอบ 22 ปี เรื่องนี้บอกอะไรกับเรา. เข้าถึงได้จาก workpointTODAY: https://workpointtoday.com/what-does-it-mean-when-fed-raises-interest-rates/

L, M. (2020, July 2018). Deep Learning แบบฉบับคนสามัญชน EP 1 : Neural Network History. Retrieved from medium: https://medium.com/mmp-li/deep-learning-%E0%B9%81%E0%B8%9A%E0%B8%9A%E0%B8%89%E0%B8%9A%E0%B8%B1%E0%B8%9A%E0%B8%84%E0%B8%99%E0%B8%AA%E0%B8%B2%E0%B8%A1%E0%B8%B1%E0%B8%8D%E0%B8%8A%E0%B8%99-ep-1-neural-network-history-f7789236a9a3

Ling Qi, M. K. (2020). Event-Driven LSTM For Forex Price Prediction. IEEE.

M.A. Ganaiea, M. H. (2022). Ensemble deep learning: A review. Elsevier.

M.S. Islam, E. H. (2021). Foreign exchange currency rate prediction using a GRU-LSTM hybrid network. Soft Computing.

Mohammed, A. (2021). A Survey on Deep Learning for Time-Series Forecasting. ResearchGate.

Molnar, C. (2023, March 2). A Guide for Making Black Box Models Explainable. Retrieved from https://christophm.github.io/interpretable-ml-book/feature-importance.html#feature-importance

Nontapa, C. (2022). Improving Time Series Forecasting Using Deep Learning.

Paliari, I., Karanikola, A., & Kotsiantis, S. (2021). A comparison of the optimized LSTM , XGBOOST, ARIMA in Time Series forecasting. International Conference on Information, Intelligence, Systems & Applications.

Panda, M. M., Panda, S. N., & Pattnaik, P. K. (2020). Exchange Rate Prediction using ANN and Deep. Indo-Taiwan ICAN.

Radhakrishnan, P. (10, August 2017). What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network? Retrieved from medium: https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a#:~:text=Deep%20Neural%20Network%3F-

,What%20are%20hyperparameters%3F,optimizing%20the%20weights%20and%20bias)
.

Ramzai, J. (2020, August 11). PSI and CSI: Top 2 model monitoring metrics. Retrieved from https://towardsdatascience.com/psi-and-csi-top-2-model-monitoring-metrics-924a2540bed8#:~:text=Moving%20on%20to%20the%20subject,domains%20especially%20credit%20risk%20domain.

Shaolong Sun, S. W. (2020). A new ensemble deep learning approach for exchange rates forecasting and. Advanced Engineering Informatics.

Singh, A. (2022, June 15). A Comprehensive Guide to Ensemble Learning (with Python codes). Retrieved from analytics vidhya: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/

St´efano Frizzo Stefenon, M. H.-C. (2022). Electric Power Systems Research. Elsevier.

Suna, S., Wang, S., & Wei, Y. (2020). A new ensemble deep learning approach for exchange rates forecasting and. Advanced Engineering Informatics.

Tato, A., & Nkambou, R. (2018). IMPROVING ADAM OPTIMIZER. ICLR.

Thailand’s currency weakens to 38 baht per US dollar. (2022, September 28). Retrieved from Thai PBS World: https://www.thaipbsworld.com/thailands-currency-weakens-to-38-baht-per-us-dollar/

Trilok Nath Pandey, A. K.-B. (2020). A novel committee machine and reviews of neural network and. Journal of King Saud University .

Yunze Li, Y. X. (2022). Feature importance recap and stacking. Sydney.

รุ่งนภา พิมมะศรี. (9 มิถุนายน 2565). ‘เงินเฟ้อ’ มีหลายแบบ ‘การขึ้นดอกเบี้ย’ อาจไม่แก้ปัญหาของไทยและเสี่ยงทำให้เศรษฐกิจถดถอย. เข้าถึงได้จาก Thairath Online: https://plus.thairath.co.th/topic/money/101631

สถานการณ์ "เงินเฟ้อ" ของ "ไทย" เป็นอย่างไร ? (22 มีนาคม 2565). เข้าถึงได้จาก Money Buffalo: https://www.moneybuffalo.in.th/economy/how-is-

%E0%B9%80%E0%B8%87%E0%B8%B4%E0%B8%99%E0%B9%80%E0%B8%9F%E0%B9%89%E0%B8%AD-%E0%B9%84%E0%B8%97%E0%B8%A2-situation

Appendix

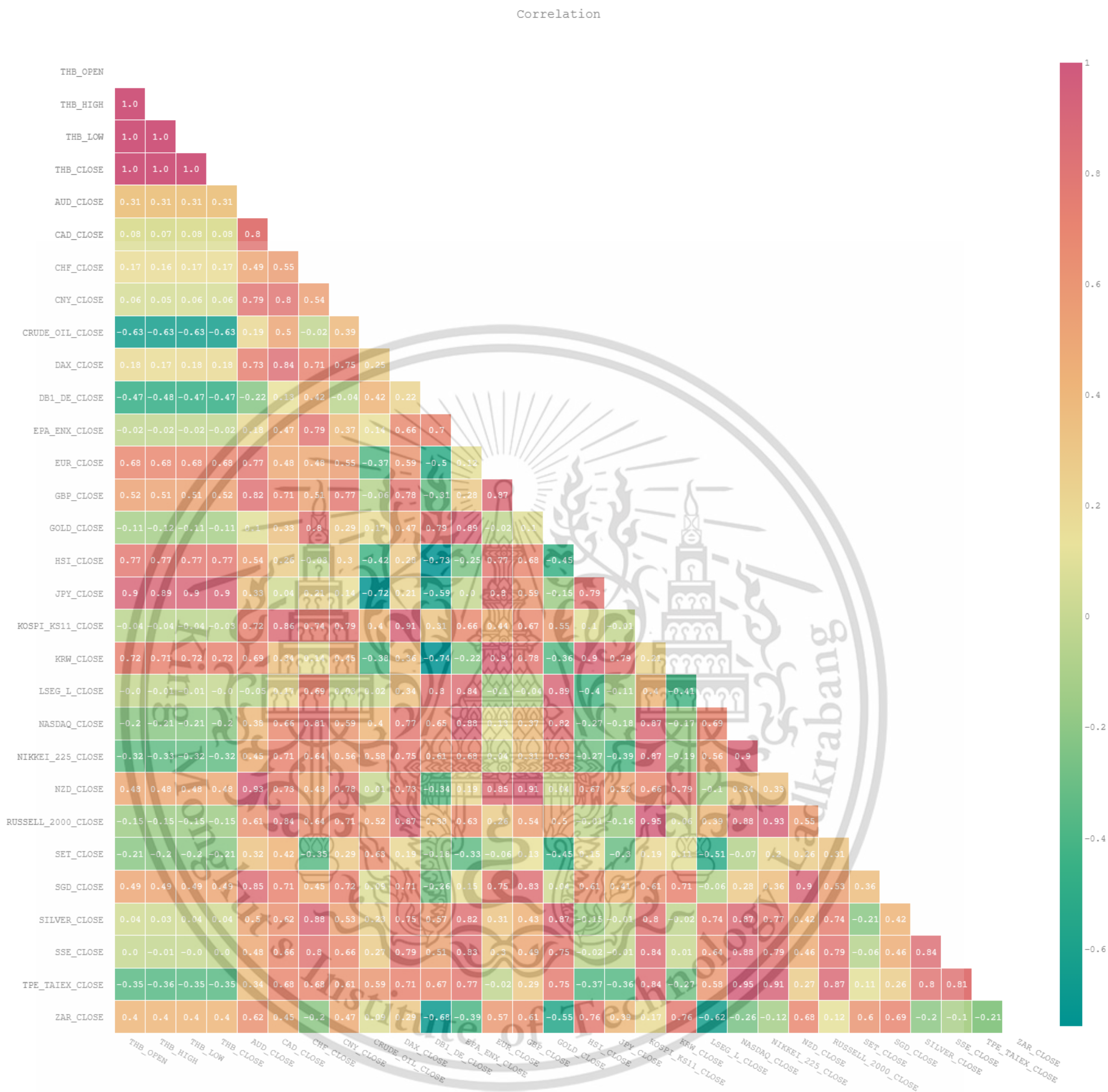

Figure A1 Heat Map showing the correlations among continuous variables before selecting features using Pearson Correlation

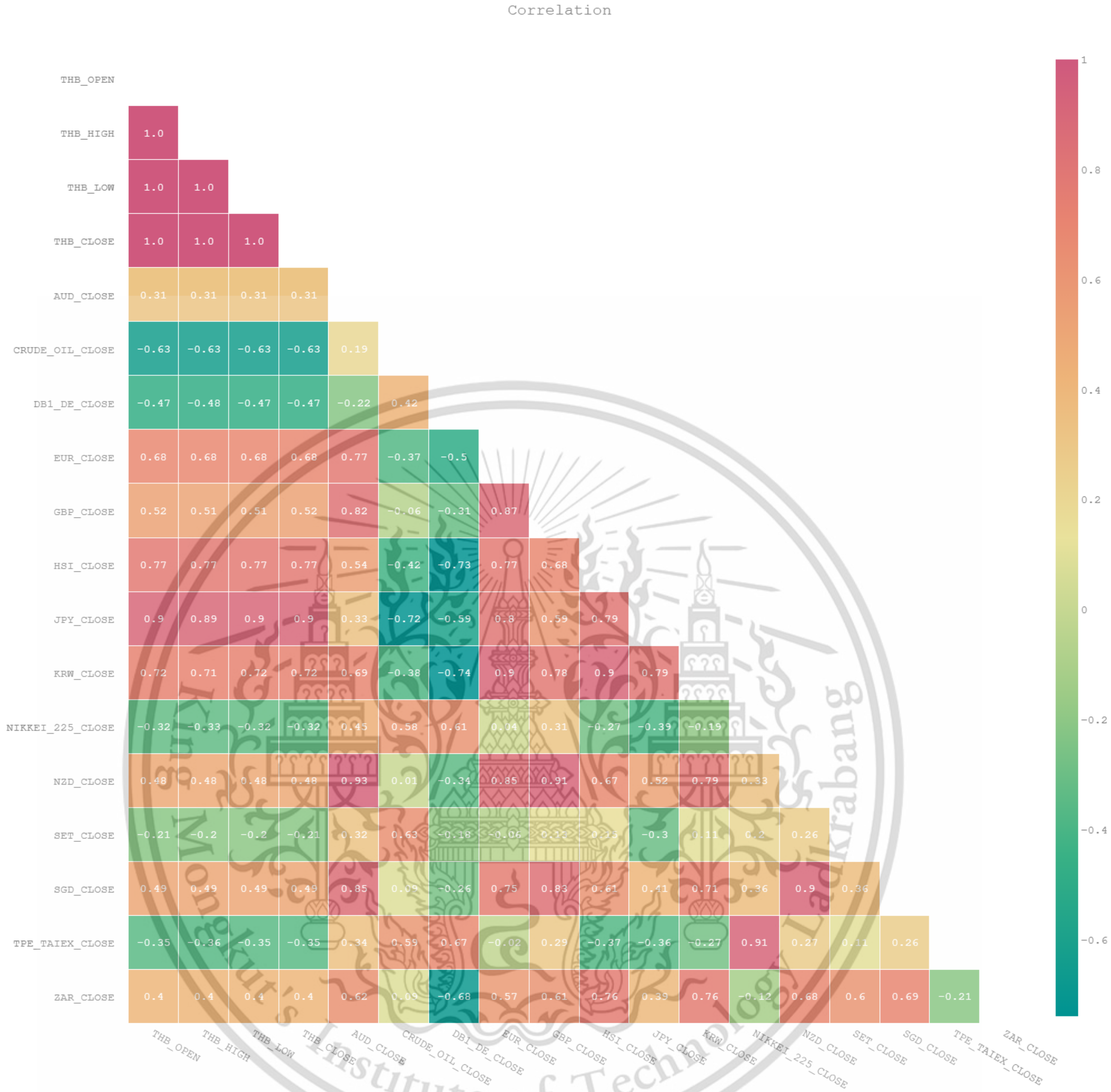

Figure A2 Heat Map showing the correlations among continuous variables after selecting features using Pearson Correlation

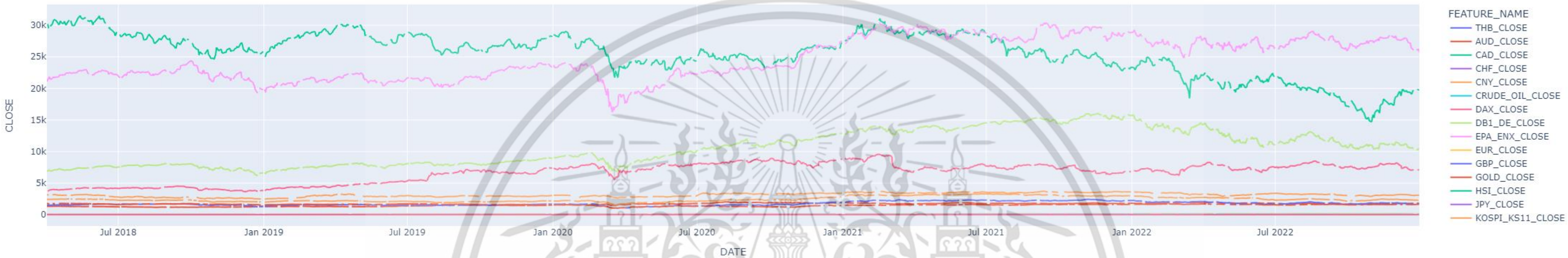

Figure A3 Trend lines of selected continuous variable's price throughout the time between April 2018 and December 2022

Figure A4 Trend line of THB/USD between April 2018 and December 2022

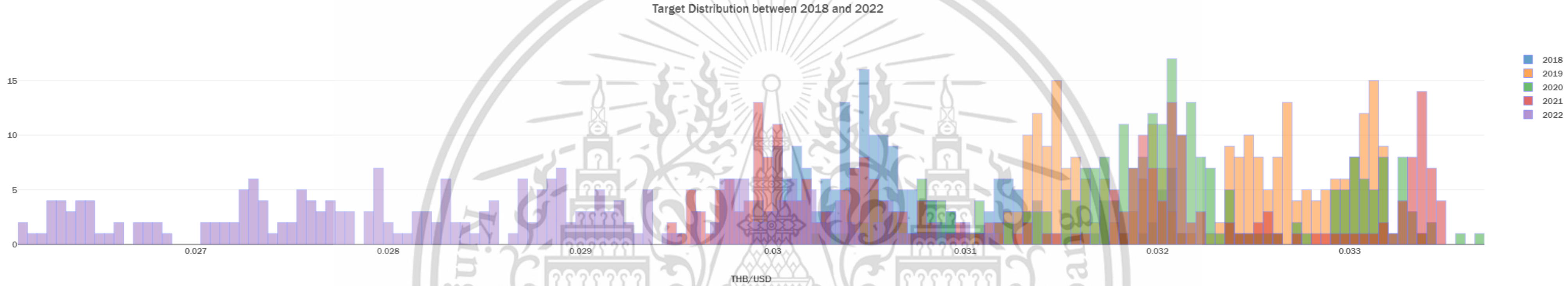

Figure A5 Target Distribution between April 2018 and December 2022

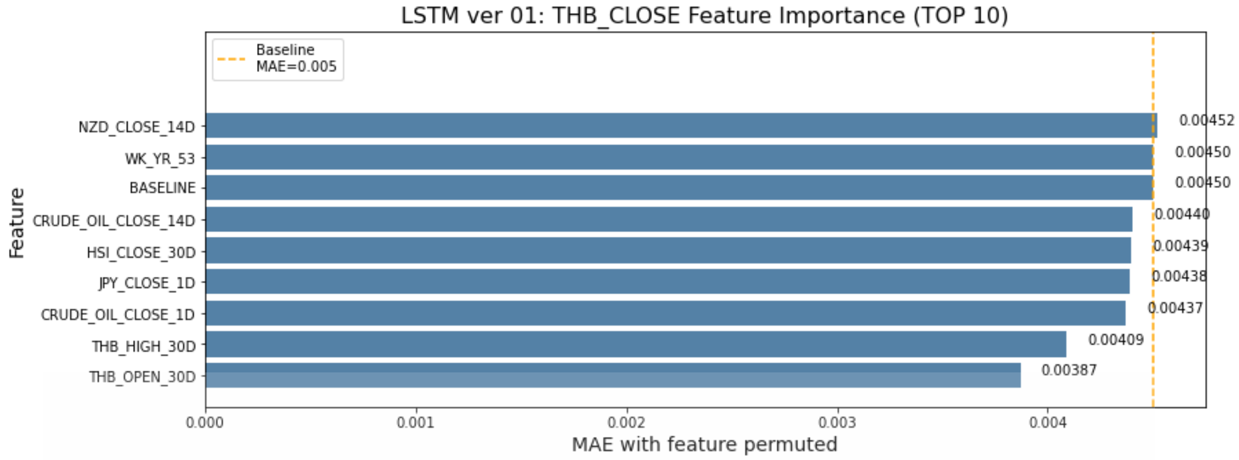

Figure A6 Feature Importance lstm_1

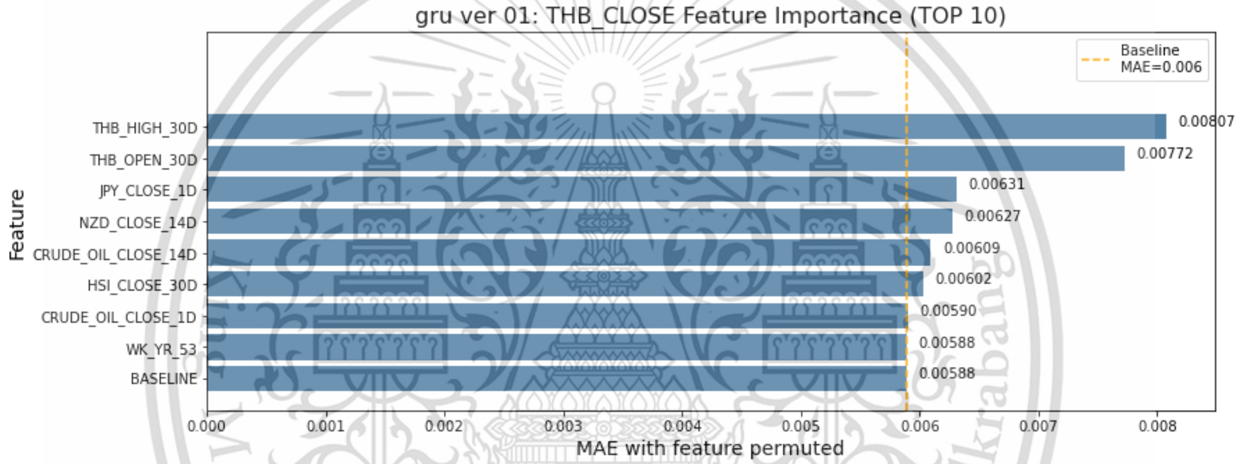

Figure A7 Feature Importance gru_1

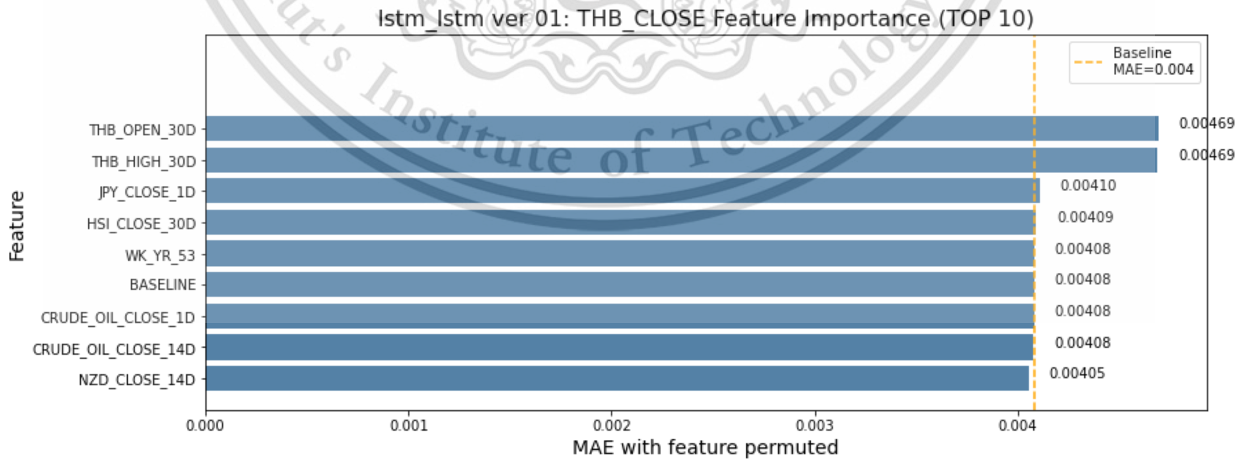

Figure A8 Feature Importance lstm_lstm_1

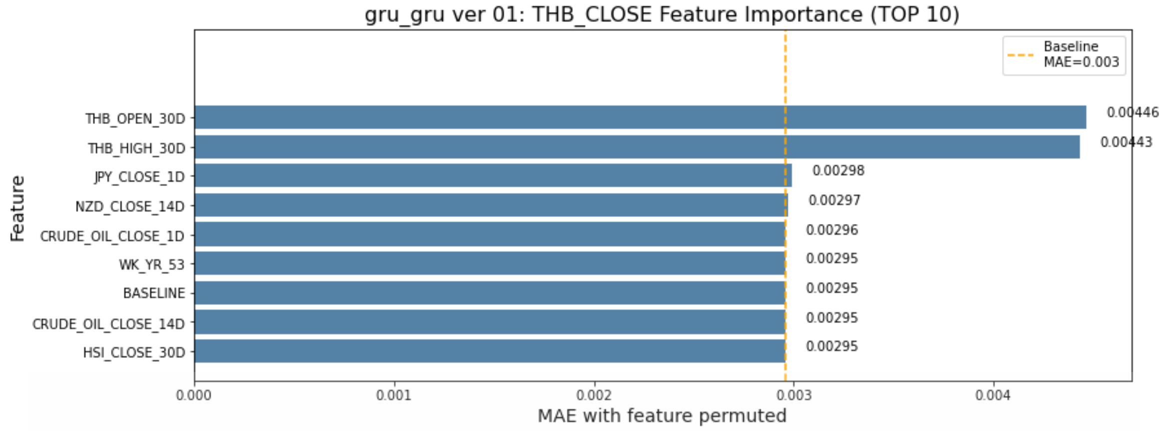

Figure A9 Feature Importance gru_gru_1

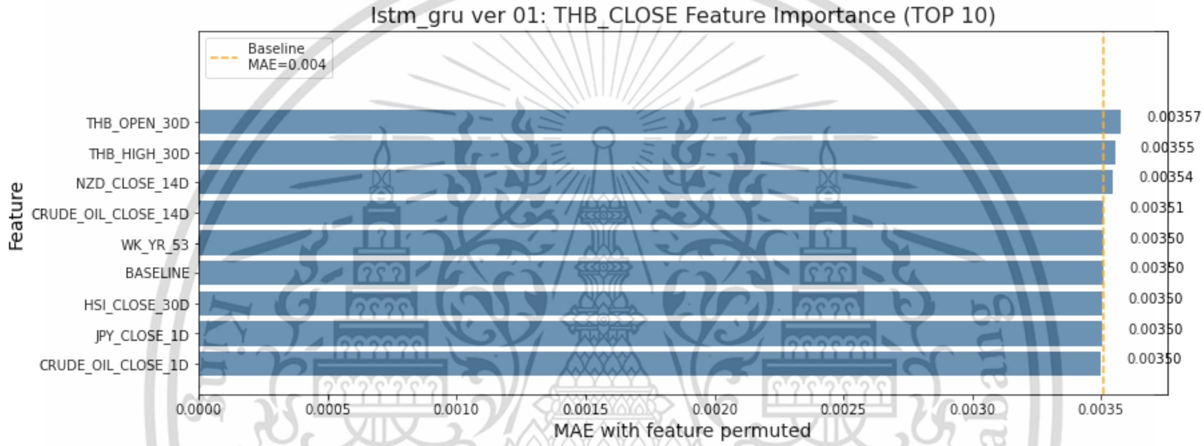

Figure A10 Feature Importance lstm_gru_1

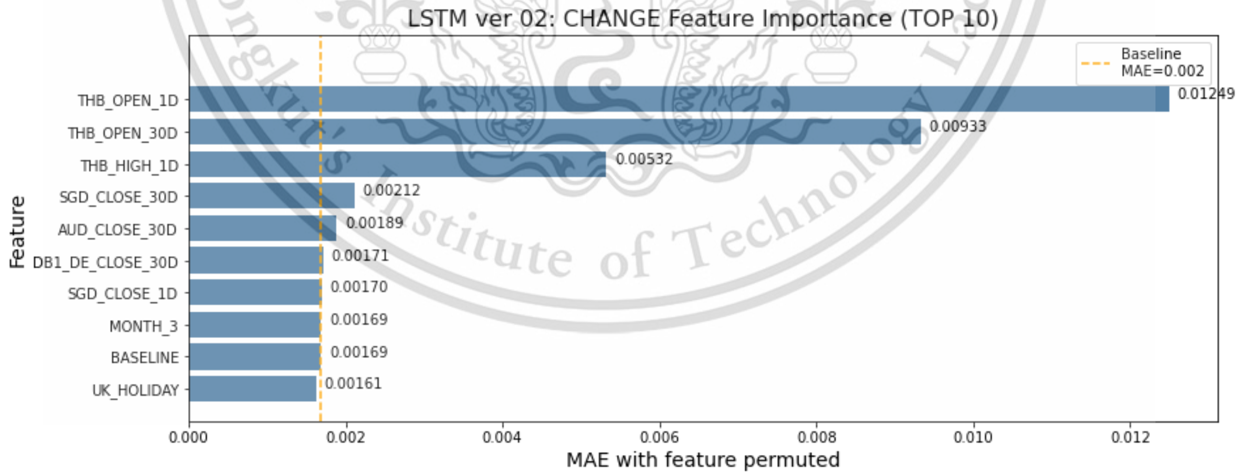

Figure A11 Feature Importance lstm_lstm_1

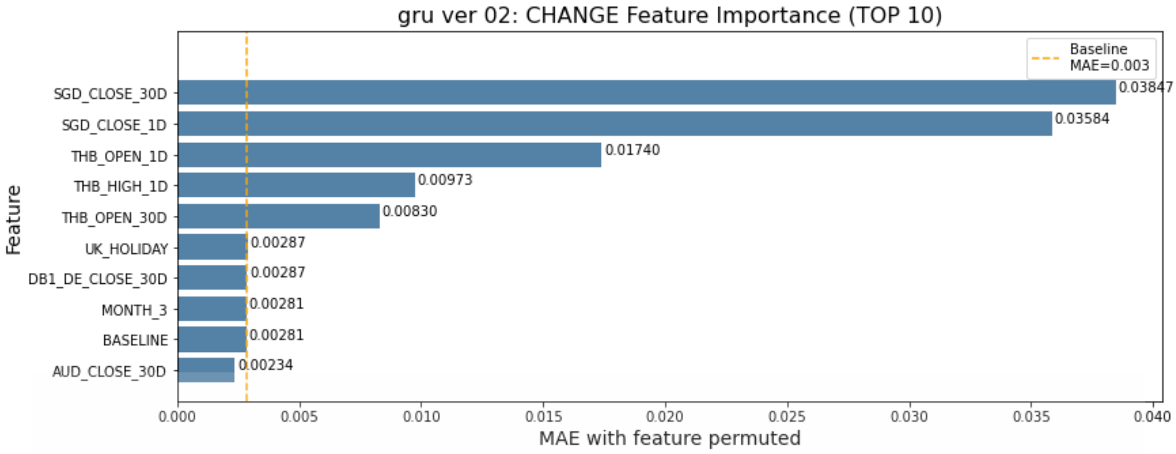

Figure A12 Feature Importance gru_2

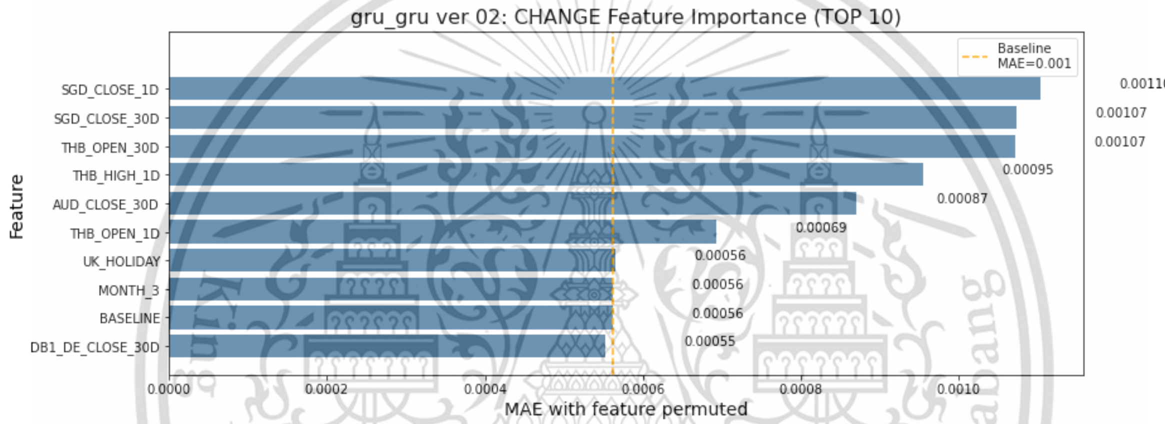

Figure A13 Feature Importance gru_gru_2

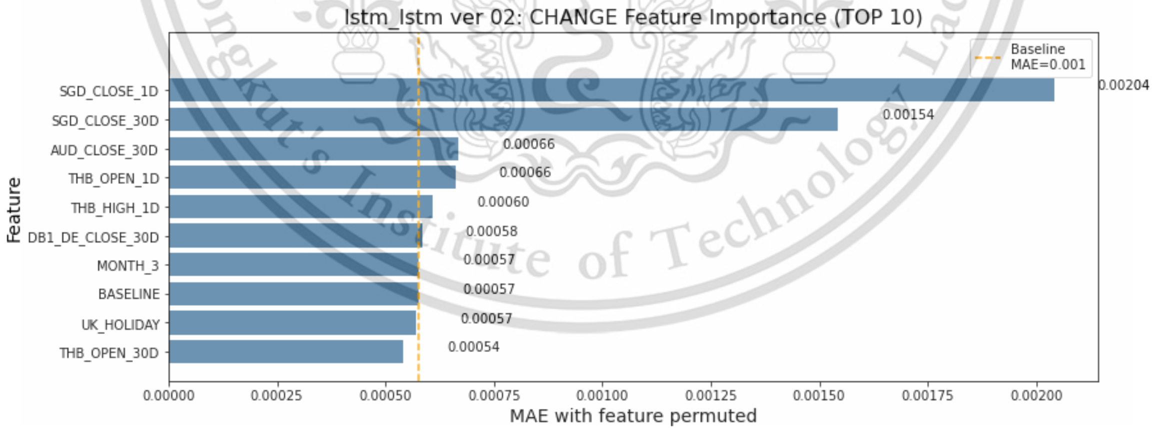

Figure A14 Feature Importance lstm_lstm_2

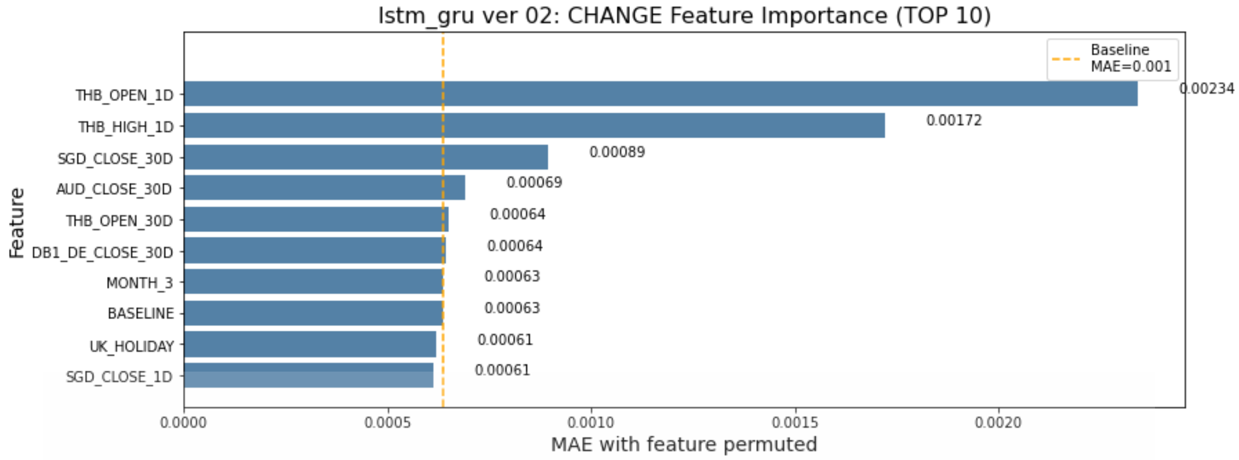

Figure A15 Feature Importance lstm_gru_2

AUTHOR BIOGRAPHY

Name	Paweena Rouyprasert
Date of Birth	June 30, 1996
Address	Klong Toei, Bangkok 10110
Education	(2020) Bachelor's degree of Economics
	Thammasat University
Work Experience	
	(April 2023 – Present) Model Validator (Data Scientist)
	Tri Petch Isuzu Leasing
	(November 2022 – April 2023) Data Scientist
	TD Tawandang – Carabao Group
	(August 2020 –July 2022) Data Analyst
	Siam Kubota Leasing