



## Implementation of Gated Recurrent Unit, Long Short-Term Memory and Derivatives for Gold Price Prediction

Amanda Iksanul Putri<sup>1\*</sup>, Yulia Syarif<sup>2</sup>,  
Nasywa Rihadatul Aisyi<sup>3</sup>, Nuralisa Waeyusoh<sup>4</sup>

<sup>1</sup>Department of Information System, Faculty of Science and Technology,  
Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia

<sup>3</sup>Department of International Trade and Fianance, Faculty of Business and Management Sciences,  
Istanbul Sabahattin Zaim University, Turkey

<sup>4</sup>Department of Islamic Business Innovation, Faculty of Islamic Science,  
Prince of Songkla University, Thailand

E-Mail: <sup>1</sup>amandaiksanulputri@gmail.com, <sup>2</sup>yuliasyarif18@gmail.com,  
<sup>3</sup>nrihadatulaisyi@gmail.com, <sup>4</sup>sa.alisa19825@gmail.com

Received Aug 8th 2024; Revised Nov 18th 2024; Accepted Dec 28th 2024; Available Online Jan 12th 2025, Published Jan 18th 2025

Corresponding Author: Amanda Iksanul Putri

Copyright © 2025 by Authors, Published by Institute of Research and Publication Indonesia (IRPI)

### Abstract

Gold is a precious metal with high resale value, often considered a safe investment as its price typically rises with inflation, attracting investors. However, even slight changes in gold prices can have significant impacts. To build an accurate forecasting model, this study applies and compares Gated Recurrent Unit (GRU), Bidirectional Gated Recurrent Unit (Bi-GRU), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (Bi-LSTM) algorithms on global gold prices. GRU and LSTM are recurrent neural networks designed to capture patterns in sequential data, where GRU uses a simplified gating mechanism to retain essential information, and LSTM, with its more complex gates, helps manage long-term dependencies in data. Bi-GRU and Bi-LSTM process data bidirectionally, capturing context from both past and future sequences for better prediction accuracy. This research uses data from Yahoo Finance (01-01-2014 to 12-06-2024) and experiments with optimization techniques (Adam, AdamW, Adamax, and Nadam), batch sizes (8, 16, and 32), time steps (10, 20, and 30), and a learning rate of 0.0001, trained for 1000 epochs with checkpoints and early stopping. Bi-GRU with Nadam, batch size 8, and 20 time steps proved most effective, with MSE of 4.1153, RMSE of 2.0286, MAE of 1.5881, and MAPE of 0.8857%. Forecasts using this model predict a 20-day decline in gold prices.

Keyword: Derivatives, Gated Recurrent Unit, Gold, Long Short-Term Memory, Prediction

### 1. INTRODUCTION

One valuable metal that has a high resale value is gold. Both inflation and economic downturns cannot affect gold [1]. As a reserve held by a country's central bank, gold plays a crucial role in strengthening the national economy through strong collateral policies and increasing global market confidence. According to research data from the World Gold Council, demand for gold investments has increased by almost 10% annually over the past 20 years. Gold is a precious metal with a high resale value. Both inflation and economic downturns cannot affect gold [2]. Gold was chosen by 46% of respondents in the 2019 retail gold market study, placing it third among preferred investment products, behind life insurance (54%) and savings (78%) [3].

Gold investment is often considered a safe option because its price tends to rise with inflation, making it a favorite among investors [4]. Moreover, gold may be easily liquidated or converted into cash because it is a liquid asset [5]. Historically, the adequate support and liquidity of gold have attracted the attention of many financial specialists. However, even small potential changes in gold prices can have significant impacts, resulting in either substantial gains or significant losses. Therefore, predicting gold behavior crucial for financial investors and central banks in making appropriate investment decisions and managing potential risks [6].

Given the important role of gold for financial investors and central banks, it is necessary to predict daily gold prices (time series data) to optimize investor investments as a basic reference [5]. Time series data



may be predicted using a variety of machine learning techniques. One machine learning approach for forecasting periodic data is the Recurrent Neural Network (RNN), which creates new values for usage in the future by processing previous data. Some of the variations of RNN itself include Gated Recurrent Unit (GRU), Bidirectional Gated Recurrent Unit (Bi-GRU), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (Bi-LSTM) [7].

The architectures of GRU, Bi-GRU, LSTM, and Bi-LSTM each have their unique advantages in handling time series and sequential data, making them particularly suitable for predicting gold prices, which are influenced by various complex temporal factors [8]. On the other hand, XGBoost is more suitable for simpler data and does not handle temporal dependencies well [9], while ARIMA is less flexible in handling the dynamic fluctuations of gold prices [10]. GRU is known for its simplicity and computational efficiency, offering faster training while effectively addressing the vanishing gradient problem, which is important when modeling financial data that involves long-term dependencies [8]. Bi-GRU enhances contextual understanding by processing data in both time directions, which is crucial for gold price prediction, as market movements are often affected by both past and future events or trends, thus improving accuracy [11]. LSTM, with its three gates and cell state, excels at retaining long-term information, making it ideal for predicting gold prices, as such data often exhibits long-range dependencies. Additionally, LSTM's ability to handle the vanishing gradient problem makes it suitable for time series prediction tasks like forecasting gold prices [12]. Bi-LSTM further enhances performance by incorporating bidirectional processing, allowing the model to better understand the temporal dependencies and contextual relationships within the price data, which is essential for applications like gold price forecasting that rely on capturing both past and future patterns [13]. Choosing the right optimizer is essential to maximizing the performance of these architectures, as it influences the efficiency of training and convergence, thereby enhancing the model's predictive accuracy for gold prices [14].

The optimizers Adam, AdamW, Adamax, and Nadam each have their advantages in machine learning. Adaptive Moment Estimation (Adam) is renowned for its effective adaptive learning, efficient memory usage, and fast convergence, combining the benefits of AdaGrad and RMSProp [15]. Adam with Weight Decay (AdamW) improves regularization by introducing weight decay independent of gradient updates, resulting in more stable and often better convergence in overfitting scenarios [16]. Adamax uses the infinity norm, making it more stable and suitable for problems with large gradients, capable of handling various parameter scales [15]. Combining Adam with Nesterov Accelerated Gradient (NAG) allows for faster convergence and improved avoidance of local plateaus in Nesterov-accelerated Adaptive Moment Estimation (Nadam) [17].

In the study by Rahmadeyan and Mustakim, the GRU model was the best performer with an MSE of 4958.9168, RMSE of 70.4195, and MAPE of 1.1699% in stock investment. The GRU model predicted a decline in stock prices for the following month [15]. The subsequent study by Fayyad et al., with an Mean Squared Error (MSE) of 3741.6999, RMSE of 61.1694, Mean Absolute Error (MAE) of 45.6246, and Mean Absolute Percent Error (MAPE) of 0.3054%, showed that the GRU model with the Nadam optimizer, batch size of 16, and a time step of 30 performed the best. This study predicted the Rupiah to USD exchange rate for the next 30 days [16].

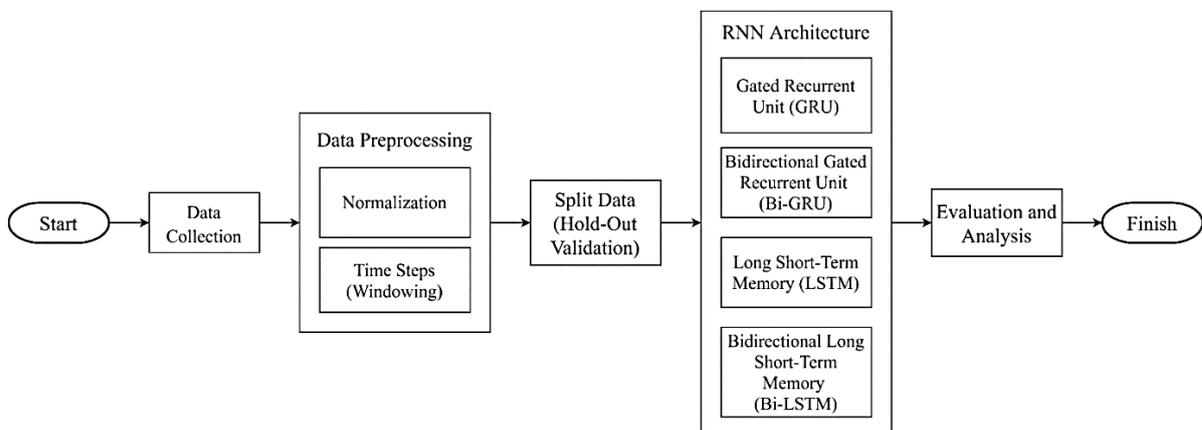
In the study by Rahmadeyan and Mustakim, the GRU model was the best performer with an MSE of 4958.9168, Root Mean Squared Error (RMSE) of 70.4195, and MAPE of 1.1699% in stock investment. The GRU model predicted a decline in stock prices for the following month [18]. The subsequent study by Fayyad et al., with an MSE of 3741.6999, RMSE of 61.1694, MAE of 45.6246, and MAPE of 0.3054%, showed that the GRU model with the Nadam optimizer, batch size of 16, and a time step of 30 performed the best. This study predicted the Rupiah to USD exchange rate for the next 30 days [19]. In the study by Ma and colleagues, to improve oil production forecast accuracy, a combination of Sparrow Search Algorithm (SSA) and Bi-GRU was used. Bi-GRU leverages both past and future information to optimize production, with parameters adjusted through SSA. The results showed better performance compared to traditional methods, unidirectional recurrent neural networks, and decline curve analysis [20].

In the study Yurtsever on gold price prediction, the results showed that, with a MAPE of 3.48, RMSE of 61.728, and MAE of 48.85, the LSTM model performed better than the others [21]. In another study on the same topic, the RMSE for LSTM training data was 391.95, and for testing data, it was 412.36, making LSTM the best-performing model [5]. Further research on rice crop monitoring yielded high accuracy and Kappa values (>97% for all techniques and measures). With statistically significant differences in the McNemar test ( $p < 0.05$ ), Bi-LSTM was found to be the best model [22]. Lastly, this research discovered that Bi-LSTM outperformed other models in the COVID-19 context based on the  $r_2\_score$ , MAE, and RMSE indices. With the best  $r_2\_score$  value of 0.9997 for recovery cases in the same region, Bi-LSTM achieved the lowest MAE and RMSE values, 0.0070 and 0.0077, respectively, for death cases in China. Bi-LSTM proved advantageous for pandemic prediction due to its resilience and predictive accuracy, which enhanced planning and management [23].

From previous studies, several algorithms have been identified as the best for making predictions. In this study, a comparison will be conducted on the best algorithms from prior research, namely GRU, Bi-GRU, LSTM, and Bi-LSTM. In addition to comparing these four algorithms, this research will also optimize model performance using Adam, AdamW, Adamax, and Nadam optimizers. Various experiments will be conducted in this study, using batch sizes (8, 16, and 32) and time steps (10, 20, and 30). The best resulting model will be tested on gold prices and can be used as a recommendation for investors in making investment decisions.

## 2. MATERIAL AND METHOD

The methodology for predicting gold prices involves several key steps. First, historical gold prices from Yahoo Finance over the past 10 years (01-01-2014 to 12-06-2024) are collected. The data then undergoes processing, including normalization (using Date and Close attributes) and windowing with a sliding window approach. The data set is split 8:2, with 20% designated for testing and the remaining 80% for training. Time steps of 10, 20, and 30 are used to create series for predicting the target value. Four architectures RNN, LSTM, BiLSTM, GRU, and BiGRU are used to train the model, alongside four optimizers: Adam, AdamW, Adamax, and Nadam. Experiments with different batch sizes (8, 16, and 32) are conducted to generate the optimal model. Next, based on the models' lowest MSE, RMSE, and MAPE values, the optimal model is tested and assessed. In the end, the most accurate model is used to forecast gold prices in the future. An illustration of the research steps may be seen in Figure 1.



**Figure 1.** Research Methodology

### 2.1. Data Collecting

The Yahoo! Finance dataset [24] includes the closing price, highest price, opening price, and lowest price of gold. The price at which gold is initially traded when the market opens is known as the opening price, and the price at which the market closes at the conclusion of a trading day or session is known as the closing price. Additionally, the Yahoo! Application Programming Interface (API) allows us to retrieve historical gold price data [25]. The historical gold price data included in this study spans a ten-year period, from January 1, 2014, to June 12, 2024. In order to encompass a wide variety of market situations, including times of stability, volatility, and noteworthy international events, a 10-year span was selected. This extended timeframe ensures robust analysis by encompassing diverse market dynamics and trends.

### 2.2. Data Preprocessing

Data preprocessing is crucial for model training as it emphasizes relevant features, speeds up convergence, and reduces the impact of irrelevant information [26]. The preprocessing stage includes normalization and windowing techniques to prepare the dataset. The Date attribute is transformed into a time sequence, ensuring that the temporal aspect of the data is preserved. This prevents any future information from leaking into the model during training by keeping the time order intact. The Close attribute is scaled using min-max normalization to bring all values into a consistent range, ensuring that this feature contributes equally to the learning process. To capture temporal dependencies, the dataset is divided using a sliding window approach with overlapping segments. Each segment contains a fixed number of time steps (10, 20, and 30), ensuring that the model learns from past information without incorporating future data, thereby maintaining the integrity of the time-series data.

### 2.3. Split Data

Using hold-out validation, the data is divided into 80% training and 20% testing sets. An out-of-sample assessment technique called hold-out evaluation divides the available data into two sets: a training set and a test set. Hence, the test set also referred to as the holdout set or holdout data is out of sample data. This method allows for an objective assessment of the model's performance and its generalization to new, untested data by guaranteeing that the model is trained on one subset of the data and evaluated on another [27].

### 2.4. Gated Recurrent Unit (GRU)

One common neural network component used to control input flow during learning is the GRU [28]. The GRU network consists of only two gating layers: the reset gate and the update gate. The update gate determines how much information from the previous memory should be kept, much as the forget gate and input gate in an LSTM cell, while the reset gate determines how much information from the previous memory should be deleted [29]. These gates,  $z_t$  and  $r_t$ , along with their parameters,  $W_z$ ,  $W_r$ , and  $W$ , are updated during the training process. The relevant equations can be seen equation 1 to 4 [30].

$$Z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (1)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (2)$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_{t-1}, x_t] + b_c) \quad (3)$$

$$h_t = (1 - Z_t \times h_{t-1} + Z_t \times \tilde{h}_t) \quad (4)$$

### 2.5. Bidirectional Gated Recurrent Unit (Bi-GRU)

Two GRUs coupled to the same output layer but orientated in different directions make up the Bi-GRU. As a result, every point in the input sequence has entire past and future information available to the output layer, enabling it to include additional sequential information [31]. The output vectors from the hidden layer of the forward layer in the first and second layers of the Bi-GRU neural network at time  $t$  are represented as  $h_{\leftarrow t} \in \mathbb{R}^H$  and  $h_{t \rightarrow} \in \mathbb{R}^H$ , where  $H$  stands for the number of units in the GRU cell. Similar to this, the output vectors from the hidden layer of the backward layer in the first and second layers of the Bi- are  $h_{t \leftarrow} \in \mathbb{R}^H$  and  $h_{\leftarrow t} \in \mathbb{R}^H$ , where  $x_t$  is the neural network input at time  $t$  and  $n$  is the number of labels. Function  $(\cdot)$  represents the activation function, and Function  $(\cdot)$  represents the GRU neural network processing, where  $(x)_i = \text{exp}(\sum_k x_k) = \text{exp}(x)$ . The weights and biases that need to be learned are represented by the variables  $w$  and  $b$ , respectively. The relevant equations can be seen equation 5 to 9 [32].

$$\vec{h}_t^1 = f(w_{xh^1} \vec{x}_t + w_{h^1 h^1} \vec{h}_{t-1}^1 + b_{h^1}) \quad (5)$$

$$\overleftarrow{h}_t^1 = f(w_{xh^1} x_t + w_{h^1 h^1} \overleftarrow{h}_{t-1}^1 + b_{h^1}) \quad (6)$$

$$\vec{h}_t^2 = f(w_{h^1 h^2} \vec{h}_t^1 + w_{h^1 h^2} \vec{h}_{t-1}^2 + b_{h^2}) \quad (7)$$

$$\overleftarrow{h}_t^2 = f(w_{h^1 h^2} \overleftarrow{h}_t^1 + w_{h^2 h^2} \overleftarrow{h}_{t+1}^2 + b_{h^2}) \quad (8)$$

$$y_t = g(w_{h^2 y} \vec{h}_t^2 + w_{h^2 y} \overleftarrow{h}_t^2 + b_y) \quad (9)$$

### 2.6. Long Short-Term Memory (LSTM)

Long-term dependencies are handled by LSTM, which stores historical data and retrieves it as required [33]. LSTM consists of several memory cells, each equipped with three gates and states to regulate the flow of data through the cell. This design is particularly useful for predicting values in the context of time series data [34]. As listed below, there are six formal equations related to the gating processes in an LSTM cell: The current cell has the forget gate  $ft$ , the input gate  $it$ , the output gate value  $ot$ , the current cell state  $\tilde{C}t$ , the candidate cell state (next state) and the network weights  $Wf$ ,  $Wi$ ,  $Ua$ ,  $UG$ ,  $Ua$ ,  $Ua$  are the bias variables,  $bZ$ ,  $bi$ ,  $ba$ ,  $bn$  are the bias variables,  $ht$  is the current hidden state,  $ht-1$  is the previous hidden state, and  $xt$  is the new input value [35]. The relevant equations can be seen equation 10 to 15.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (10)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (11)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (12)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (13)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (14)$$

$$h_t = o_t \odot \tanh(c_t) \quad (15)$$

### 2.7. Bidirectional Long Short-Term Memory (Bi-LSTM)

An improved variant of the LSTM algorithm is called BiLSTM. The bidirectional RNN and LSTM ideal functions are combined in the BiLSTM method. This is accomplished by combining two concealed states together, enabling data from both the forward and backward layers [36]. The six independent weight matrices represented by  $w_i$  ( $1, 2, \dots, 6$ ) are as follows:  $w_1, w_3$  represents the input-to-hidden weights for the forward and backward layers;  $w_2, w_5$  represents the hidden-to-hidden weights; and  $w_4, w_6$  represents the forward and backward hidden-to-output weights. At every time step, these six weights are utilized repeatedly. Two values,  $h_t$ , must be stored in the hidden layer of the BiLSTM model:  $h_t \rightarrow$  for forward computation and  $h_t \leftarrow$  for backward calculation. The forward and backward layers' outputs are combined to produce the final output value, which has the following mathematical expression (Eqs. 16–18) [37].

$$\vec{h}_t = f(w_1 x_t + w_2 \vec{h}_{t-1}) \quad (16)$$

$$\overleftarrow{h}_t = f(w_3 x_t + w_5 \overleftarrow{h}_{t+1}) \quad (17)$$

$$O_t = g(w_4 \vec{h}_t + w_6 \overleftarrow{h}_t) \quad (18)$$

### 2.8. Sliding Window

The time series sequence is divided into two segments using the sliding window technique. The input window size is represented by the first segment, and the anticipated output size is represented by the second segment. This approach generates multiple samples for the training set by moving one step at a time along the sequence [38].

### 2.9. Optimizer

Examining the impact of variance in optimization methods, as emphasized by Ruder [14], is crucial for enhancing the performance of optimizers [39]. Adam is a well-known optimizer whose primary benefit is that parameter update magnitudes are invariant to changes in gradient size. Still, alternative optimizers are frequently used because of their superior generalization [40]. Using a neural network topology designed to reduce uncertainty, we constructed a gold price forecasting model to give optimizers in this study a comparison scenario. In this context, the performance of four optimizers Adam, AdaMax, AdamW, and Nadam was examined.

### 2.10. Evaluation Matrix

The assessment measures that were used were Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Squared Error (MSE). Because these empirical techniques successfully gauge the bias and accuracy of models, they are frequently utilized for merging and choosing forecasts[41]. These metrics are calculated using Equations (19), (20), (21), and (22). The anticipated value at time  $t$  is denoted by  $P_t$ , the observed value at time  $t$  is represented by  $Z_t$ , and the number of predictions is [42][44].

$$MAE = \frac{1}{T} \sum_{t=1}^T |P_t - \hat{Z}_t| \quad (19)$$

$$MAPE = \frac{1}{T} \sum_{t=1}^T \left| \frac{P_t - \hat{Z}_t}{Z_t} \right| \quad (20)$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (P_t - \hat{Z}_t)^2} \quad (21)$$

$$\text{MSE} = \frac{1}{T} \sum_{t=1}^T (P_t - \hat{Z}_t)^2 \quad (22)$$

### 3. RESULTS AND DISCUSSION

The GRU, Bi-GRU, LSTM, and Bi-LSTM algorithms were modeled to predict gold prices through several experiments aimed at generating the optimal model. These experiments included optimization techniques (Adam, AdamW, Adamax, and Nadam), batch sizes (8, 16, and 32), time steps (10, 20, and 30), and a fixed learning rate of 0.0001, trained over 1000 epochs using callbacks (checkpoint and early stopping). The initial step was to implement the GRU algorithm with 4 optimizers, 3 time steps, and 3 batch sizes.

#### 3.1. Evaluation Results of GRU Algorithm Modeling

**Tabel 1.** Evaluation Result of GRU Algorithm Modeling

Optimizer	Experiment		GRU			
	Batch Size	Timestep	MSE	RMSE	MAE	MAPE
Adam	8	10	6.2724	2.5045	1.9609	1.0913
Adam	8	20	4.5614	2.1357	1.6323	0.9065
Adam	8	30	5.9780	2.4450	1.9337	1.0799
Adam	16	10	5.6661	2.3804	1.8787	1.0480
Adam	16	20	7.6934	2.7737	2.2120	1.2330
Adam	16	30	5.9829	2.4460	1.9511	1.0910
Adam	32	10	8.5831	2.9297	2.3457	1.3065
Adam	32	20	8.0502	2.8373	2.2255	1.2364
Adam	32	30	7.7677	2.7871	2.2028	1.2264
AdamW	8	10	5.5395	2.3536	1.8687	1.0444
AdamW	8	20	4.8064	2.1923	1.7204	0.9602
AdamW	8	30	5.9798	2.4454	1.9088	1.0641
AdamW	16	10	7.3352	2.7084	2.1450	1.1947
AdamW	16	20	5.0157	2.2396	1.7577	0.9794
AdamW	16	30	8.6809	2.9463	2.3234	1.2913
AdamW	32	10	6.9304	2.6326	2.0839	1.1621
AdamW	32	20	6.7107	2.5905	2.0518	1.1428
AdamW	32	30	7.2060	2.6844	2.1276	1.1864
Adamax	8	10	8.2166	2.8665	2.2867	1.2739
Adamax	8	20	6.9855	2.6430	2.1124	1.1797
Adamax	8	30	8.1215	2.8498	2.2403	1.2453
Adamax	16	10	8.4393	2.9050	2.2636	1.2565
Adamax	16	20	8.9676	2.9946	2.3988	1.3364
Adamax	16	30	7.6339	2.7629	2.2104	1.2317
Adamax	32	10	8.2975	2.8805	2.2931	1.2777
Adamax	32	20	9.3000	3.0496	2.4025	1.3344
Adamax	32	30	9.4126	3.0680	2.4369	1.3565
Nadam	8	10	6.2569	2.5014	1.9861	1.1077
Nadam	8	20	4.5621	2.1359	1.6431	0.9130
Nadam	8	30	5.0733	2.2524	1.8085	1.0094
Nadam	16	10	7.0525	2.6557	2.1561	1.2053
Nadam	16	20	5.7489	2.3977	1.9133	1.0698
Nadam	16	30	7.9895	2.8266	2.2458	1.2492
Nadam	32	10	7.7352	2.7812	2.1983	1.2229
Nadam	32	20	7.9360	2.8171	2.1719	1.2048
Nadam	32	30	6.9658	2.6393	2.0979	1.1703

With an MSE of 4.5614, RMSE of 2.1357, MAE of 1.6323, and MAPE of 0.9065%, the GRU model with Adam optimization, a batch size of 8, and a time step of 20 provided the best model according to the trials. Table 1 displays the outcomes of the GRU modeling.

#### 3.2. Evaluation Results of Bi-GRU Algorithm Modeling

**Tabel 2.** Evaluation Result of Bi-GRU Algorithm Modeling

Optimizer	Experiment		Bi-GRU			
	Batch Size	Timestep	MSE	RMSE	MAE	MAPE
Adam	8	10	4.7659	2.1831	1.7329	0.9691
Adam	8	20	4.6819	2.1638	1.7370	0.9714

Experiment			Bi-GRU			
Optimizer	Batch Size	Timestep	MSE	RMSE	MAE	MAPE
Adam	8	30	5.4362	2.3316	1.8221	1.0152
Adam	16	10	8.8455	2.9741	2.2736	1.2557
Adam	16	20	6.9040	2.6275	2.0909	1.1679
Adam	16	30	6.5067	2.5508	2.0676	1.1583
Adam	32	10	5.7263	2.3930	1.9101	1.0662
Adam	32	20	5.8747	2.4238	1.9344	1.0813
Adam	32	30	9.6192	3.1015	2.5696	1.4416
AdamW	8	10	6.9075	2.6282	2.0991	1.1690
AdamW	8	20	4.8022	2.1914	1.7406	0.9733
AdamW	8	30	4.8892	2.2112	1.7007	0.9442
AdamW	16	10	4.2985	2.0733	1.5925	0.8841
AdamW	16	20	5.5130	2.3480	1.8984	1.0619
AdamW	16	30	5.2069	2.2819	1.8412	1.0315
AdamW	32	10	6.2945	2.5089	2.0330	1.1362
AdamW	32	20	6.6169	2.5723	2.0233	1.1276
AdamW	32	30	7.4548	2.7303	2.2403	1.2536
Adamax	8	10	6.8418	2.6157	2.0762	1.1556
Adamax	8	20	4.5747	2.1389	1.6675	0.9284
Adamax	8	30	4.9038	2.2144	1.7049	0.9475
Adamax	16	10	9.9533	3.1549	2.4869	1.3818
Adamax	16	20	6.3590	2.5217	1.9882	1.1082
Adamax	16	30	6.5623	2.5617	2.0473	1.1420
Adamax	32	10	8.9498	2.9916	2.3864	1.3293
Adamax	32	20	15.4491	3.9305	3.1651	1.7711
Adamax	32	30	5.0221	2.2410	1.7917	0.9992
Nadam	8	10	5.1700	2.2738	1.7831	0.9916
Nadam	8	20	4.1153	2.0286	1.5881	0.8857
Nadam	8	30	4.5362	2.1298	1.6617	0.9268
Nadam	16	10	4.4466	2.1087	1.6364	0.9099
Nadam	16	20	5.1593	2.2714	1.7907	0.9989
Nadam	16	30	4.4652	2.1131	1.6787	0.9379
Nadam	32	10	6.8298	2.6134	2.0199	1.1196
Nadam	32	20	7.6788	2.7711	2.2146	1.2372
Nadam	32	30	8.6755	2.9454	2.4070	1.3491

The most effective Bi-GRU model was identified using the Nadam optimizer with a batch size of 8 and 20 time steps, achieving an MSE of 4.1153, RMSE of 2.0286, MAE of 1.5881, and MAPE of 0.8857%. A detailed overview of all Bi-GRU modeling results is available in Table 2.

### 3.3. Evaluation Results of LSTM Algorithm Modeling

**Table 3.** Evaluation Result of LSTM Algorithm Modeling

Experiment			LSTM			
Optimizer	Batch Size	Time Step	MSE	RMSE	MAE	MAPE
Adam	8	10	11,9450	3,4562	2,7697	1,5421
Adam	8	20	7,0031	2,6463	2,0660	1,1477
Adam	8	30	17,1460	4,1408	3,4172	1,9082
Adam	16	10	19,1385	4,3748	3,3898	1,8597
Adam	16	20	21,0114	4,5838	3,7597	2,0871
Adam	16	30	12,8330	3,5823	2,9147	1,6244
Adam	32	10	14,6458	3,8270	2,9794	1,6441
Adam	32	20	20,1335	4,4870	3,5419	1,9599
Adam	32	30	23,2409	4,8209	3,7762	2,0913
AdamW	8	10	8,5343	2,9214	2,2912	1,2742
AdamW	8	20	10,2896	3,2077	2,5597	1,4248
AdamW	8	30	11,0760	3,3281	2,7055	1,5069
AdamW	16	10	11,2665	3,3566	2,6282	1,4573
AdamW	16	20	18,2461	4,2716	3,3767	1,8765
AdamW	16	30	11,7009	3,4207	2,6821	1,4871
AdamW	32	10	15,3812	3,9219	3,0137	1,6600
AdamW	32	20	23,7634	4,8748	3,9002	2,1516
AdamW	32	30	20,6120	4,5400	3,7183	2,0685
Nadam	8	10	15,7960	3,9744	3,0436	1,6803

Optimizer	Experiment		LSTM			
	Batch Size	Time Step	MSE	RMSE	MAE	MAPE
Nadam	8	20	7,1177	2,6679	2,1111	1,1757
Nadam	8	30	16,0979	4,0122	3,0871	1,7007
Nadam	16	10	23,8502	4,8837	3,8374	2,0991
Nadam	16	20	20,2478	4,4998	3,6374	2,0209
Nadam	16	30	12,8561	3,5855	2,8640	1,5929
Nadam	32	10	20,7939	4,5600	3,5025	1,9133
Nadam	32	20	22,7048	4,7650	3,9765	2,2158
Nadam	32	30	25,7033	5,0698	4,2468	2,3607
Adamax	8	10	14,3783	3,7919	2,9670	1,6408
Adamax	8	20	17,5479	4,1890	3,3728	1,8751
Adamax	8	30	37,3567	6,1120	5,1720	2,8740
Adamax	16	10	20,8709	4,5685	3,6040	1,9788
Adamax	16	20	21,7591	4,6647	3,8158	2,1176
Adamax	16	30	32,2058	5,6750	4,3437	2,3839
Adamax	32	10	18,8228	4,3385	3,3878	1,8596
Adamax	32	20	19,0917	4,3694	3,4929	1,9382
Adamax	32	30	18,9196	4,3497	3,5388	1,9674

From the LSTM modeling analysis, the most effective model was achieved using the Adam optimizer, a batch size of 8, and 20 time steps. This model demonstrated significant metrics, including an MSE of 7.0031, RMSE of 2.6463, MAE of 2.0660, and MAPE of 1.1477%. The complete details of all LSTM modeling results are documented in Table 3.

### 3.4. Results of LSTM Algorithm Modeling

**Table 4.** Evaluation Result of Bi-LSTM Algorithm Modeling

Optimizer	Experiment		Bi-LSTM			
	Batch Size	Time Step	MSE	RMSE	MAE	MAPE
Adam	8	10	7,9231	2,8148	2,2581	1,2574
Adam	8	20	5,4659	2,3379	1,8437	1,0297
Adam	8	30	8,2640	2,8747	2,2425	1,2438
Adam	16	10	9,1646	3,0273	2,2920	1,2647
Adam	16	20	18,4291	4,2929	3,5321	1,9641
Adam	16	30	11,4785	3,3880	2,8112	1,5715
Adam	32	10	13,5864	3,6860	2,9673	1,6414
Adam	32	20	20,3252	4,5083	3,7222	2,0724
Adam	32	30	7,8108	2,7948	2,2374	1,2485
AdamW	8	10	7,5441	2,7467	2,0930	1,1557
AdamW	8	20	6,6934	2,5872	2,0803	1,1631
AdamW	8	30	7,7974	2,7924	2,2575	1,2622
AdamW	16	10	8,7949	2,9656	2,3887	1,3313
AdamW	16	20	10,8027	3,2867	2,6848	1,5013
AdamW	16	30	6,3198	2,5139	1,9810	1,1019
AdamW	32	10	7,8582	2,8032	2,1610	1,1953
AdamW	32	20	23,2707	4,8240	3,7210	2,0544
AdamW	32	30	11,7582	3,4290	2,7818	1,5473
Nadam	8	10	5,5633	2,3587	1,8515	1,0316
Nadam	8	20	7,5843	2,7540	2,2254	1,2433
Nadam	8	30	9,3038	3,0502	2,4531	1,3708
Nadam	16	10	5,4099	2,3259	1,8117	1,0093
Nadam	16	20	16,1391	4,0174	3,3507	1,8693
Nadam	16	30	7,3001	2,7019	2,1657	1,2081
Nadam	32	10	14,6588	3,8287	3,0268	1,6646
Nadam	32	20	10,8511	3,2941	2,6739	1,4933
Nadam	32	30	8,2833	2,8781	2,2532	1,2511
Adamax	8	10	14,1674	3,7640	3,0257	1,6726
Adamax	8	20	7,4240	2,7247	2,1150	1,1721
Adamax	8	30	33,0851	5,7520	4,4067	2,4099
Adamax	16	10	17,4347	4,1755	3,3070	1,8142
Adamax	16	20	24,6397	4,9638	3,8224	2,2052
Adamax	16	30	28,9168	5,3774	4,5572	2,5585
Adamax	32	10	11,2138	3,3487	2,6043	1,4395
Adamax	32	20	19,3872	4,4031	3,5030	1,9431

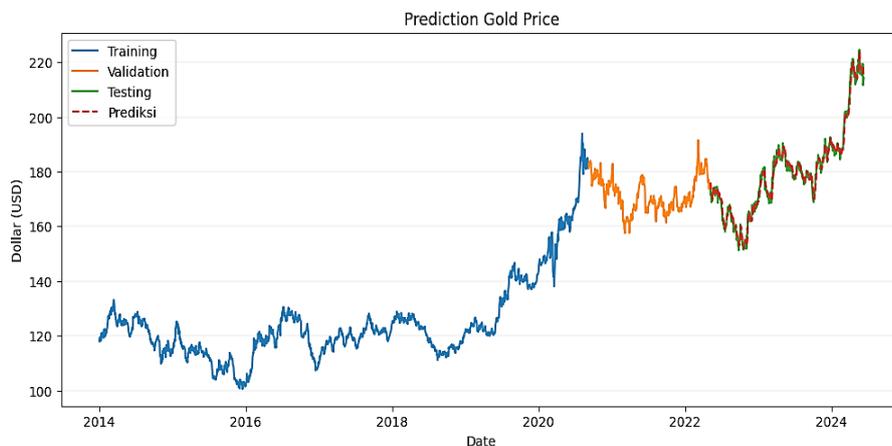
Experiment			Bi-LSTM			
Optimizer	Batch Size	Time Step	MSE	RMSE	MAE	MAPE
Adamax	32	30	28,2247	5,3127	4,2822	2,3861

From the Bi-LSTM modeling analysis, the most effective model was achieved using the Nadam optimizer, with a batch size of 16 and 10 time steps. This model demonstrated the following metrics: an MSE of 5.4099, RMSE of 2.3259, MAE of 1.8117, and MAPE of 1.0093%. The complete details of all Bi-LSTM modeling results are documented in Table 4.

### 3.5. Evaluation and Analysis

The Bi-GRU model with Nadam optimization, a batch size of 8, and 20 time steps emerged as the most effective model from all the tests conducted. The combination of these hyperparameters allowed the model to balance between computational efficiency and accuracy. With an MSE of 4.1153, RMSE of 2.0286, MAE of 1.5881, and MAPE of 0.8857%, the Bi-GRU model performed well, particularly considering that a MAPE value in the 0-0.5 range is considered excellent in time series forecasting [43]. The choice of Nadam optimization helped improve convergence by combining the benefits of both Adam and Nesterov accelerated gradient, which enhances the model’s ability to learn from sequential data without getting stuck in local minima. The batch size of 8 and 20 time steps were also key factors, as they allowed the model to capture temporal dependencies effectively while maintaining training stability. While the MAE and RMSE values may not be considered low (falling in the 10-100 range), the MAPE value suggests that the model’s ability to predict relative changes in gold prices was very accurate. Figure 2 shows the Bi-GRU model’s prediction results on the testing data, highlighting its accuracy in forecasting trends.

The projected gold prices in US dollars (USD) from 2014 to 2024 are displayed in the figure 2. The time period is represented by the X-axis, while the gold values in USD are displayed on the Y-axis. The historical data used to train the prediction model is displayed in blue, covering the period from 2014 to early 2020. The validation data is shown in orange, covering early 2020 to mid-2021, and the testing data is displayed in green from mid-2021 to mid-2023. The gold price predictions by the model are indicated by the red dashed line running from mid-2021 to early 2024. This graph illustrates that gold prices experienced a significant increase from 2018 to early 2020, followed by major fluctuations during the validation and testing periods. The predictions show a trend of increasing gold prices with minor fluctuations up to early 2024.



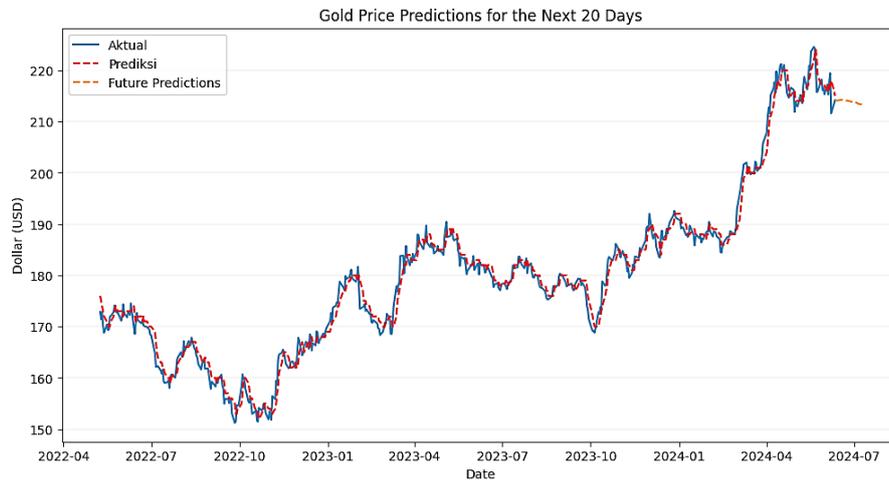
**Figure 2.** Gold Price Prediction Trends Using the Bi-GRU Algorithm

**Table 5.** Gold Price Prediction Results Using the Bi-GRU Algorithm

Date	Close	Prediction
09/05/2022	172.880005	175
10/05/2022	171.419998	174
11/05/2022	172.820007	173
12/05/2022	170.169998	173
13/05/2022	168.789993	172
...	...	...
05/06/2024	217.820007	215
06/06/2024	219.429993	216
07/06/2024	211.600006	217
10/062024	213.539993	215
11/06/2024	214.149994	214

Table 5 presents the gold price prediction results using the Bi-GRU algorithm from Figure 2. At the beginning of the period, the predictions are relatively close to the actual values, such as on May 9, 2022, where the actual price was 172.88, and the predicted value was 175. The prediction trend continues to closely follow the actual price trend with minor deviations. Toward the end of the period, from June 5 to June 11, 2024, the predictions indicate stable prices in the range of 214 to 217, demonstrating the model's consistent accuracy in forecasting price movements. Overall, the Bi-GRU algorithm shows strong performance in tracking the gold price trend.

The best model, Bi-GRU, will be used to forecast gold prices for the next twenty days. The findings suggest that a drop in gold prices is anticipated. Figure 3 displays the Bi-GRU model's prediction graph.



**Figure 3.** Gold Price Prediction for the Next 20 Days

**Tabel 6.** Detailed Gold Price Predictions for the Next 20 Days

Date	Prediction
13/06/2024	216.0
14/06/2024	216.0
17/06/2024	218.0
18/06/2024	218.0
19/06/2024	220.0
20/06/2024	221.0
21/06/2024	222.0
24/06/2024	221.0
25/06/2024	219.0
26/06/2024	217.0
27/06/2024	217.0
28/06/2024	216.0
01/07/2024	216.0
02/07/2024	215.0
03/07/2024	216.0
04/07/2024	215.0
05/07/2024	216.0
08/07/2024	217.0
09/07/2024	215.0
10/07/2024	214.0

Table 6 presents the gold price predictions based on Figure 3, where the forecast for the next 20 days indicates relatively stable movement with minor fluctuations. The price is projected to remain at 216 on June 13 and 14, 2024, before slightly increasing to reach 222 on June 21, 2024. However, the price is expected to decline thereafter, reaching 217 on June 26 and 27, 2024, and returning to 216 on June 28 and July 1, 2024. The downward trend continues, with the price predicted to fall to 214 by July 10, 2024. Overall, the forecast indicates a slight increase at the beginning of the period, followed by a gradual decline toward the end.

#### 4. CONCLUSION

It is possible to infer that GRU, Bi-GRU, LSTM, and Bi-LSTM were successful in forecasting gold prices based on the data and analysis performed. With an MSE of 4.1153, RMSE of 2.0286, MAE of 1.5881,

and MAPE of 1.5881%, Bi-GRU was determined to be the best model after these four algorithms were implemented through a series of trials. In this study, Nadam outperformed the other optimizers and improved the Bi-GRU model's performance, demonstrating its effectiveness as an optimizer. Then, the best model, the Bi-GRU model, was used to forecast gold prices for the ensuing 20 days. The study's findings showed that the Bi-GRU model forecasts a drop in gold prices. As a result, investors are able to make wise and well-informed investment selections.

## REFERENCES

- [1] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks," Sep. 2019, [Online]. Available: <http://arxiv.org/abs/1909.09586>
- [2] "The Case for Gold | World Gold Council." Accessed: Nov. 16, 2024. [Online]. Available: <https://invest.gold/why-invest-in-gold>
- [3] J. Chai, C. Zhao, Y. Hu, and Z. G. Zhang, "Structural analysis and forecast of gold price returns," *Journal of Management Science and Engineering*, vol. 6, no. 2, pp. 135–145, Jun. 2021, doi: 10.1016/j.jmse.2021.02.011.
- [4] G. D. Mishra et al., "EMAS position statement: Predictors of premature and early natural menopause," *Maturitas*, vol. 123, pp. 82–88, May 2019, doi: 10.1016/j.maturitas.2019.03.008.
- [5] P. S. A. Surya Pangestu, A. Rochman, and A. Zuhdi, "Comparison of Gold Price Prediction Techniques Using Long Short Term Memory (LSTM) And Fuzzy Time Series (FTS) Method," *Intelmatias*, vol. 3, no. 2, pp. 57–62, Aug. 2023, doi: 10.25105/itm.v3i2.17325.
- [6] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," *Neural Comput Appl*, vol. 32, no. 23, pp. 17351–17360, Dec. 2020, doi: 10.1007/s00521-020-04867-x.
- [7] I. Banerjee et al., "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification," *Artif Intell Med*, vol. 97, pp. 79–88, Jun. 2019, doi: 10.1016/j.artmed.2018.11.004.
- [8] Y. Wang, W. Liao, and Y. Chang, "Gated Recurrent Unit Network-Based Short-Term Photovoltaic Forecasting," *Energies (Basel)*, vol. 11, no. 8, p. 2163, Aug. 2018, doi: 10.3390/en11082163.
- [9] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [10] M. G. Sobol, "Panel Mortality and Panel Bias," *J Am Stat Assoc*, vol. 54, no. 285, p. 52, Mar. 1959, doi: 10.2307/2282139.
- [11] F. Meng, T. Song, D. Xu, P. Xie, and Y. Li, "Forecasting tropical cyclones wave height using bidirectional gated recurrent unit," *Ocean Engineering*, vol. 234, p. 108795, Aug. 2021, doi: 10.1016/j.oceaneng.2021.108795.
- [12] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [13] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.
- [14] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, pp. 1–14, Sep. 2016.
- [15] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ICLR*, pp. 1–15, Dec. 2014.
- [16] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *ICLR*, pp. 1–19, Nov. 2017.
- [17] P. Radhakrishnan and G. Senthilkumar, "Nesterov-accelerated Adaptive Moment Estimation NADAM-LSTM based text summarization1," *Journal of Intelligent & Fuzzy Systems*, vol. 46, no. 3, pp. 6781–6793, Mar. 2024, doi: 10.3233/JIFS-224299.
- [18] A. Rahmadyan and Mustakim, "Long Short-Term Memory and Gated Recurrent Unit for Stock Price Prediction," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 204–212. doi: 10.1016/j.procs.2024.02.167.
- [19] M. F. Fayyad, V. Kurniawan, M. R. Anugrah, B. H. Estanto, and T. Bilal, "Application of Recurrent Neural Network Bi-Long Short-Term Memory, Gated Recurrent Unit and Bi-Gated Recurrent Unit for Forecasting Rupiah Against Dollar (USD) Exchange Rate," *Public Research Journal of Engineering, Data Technology and Computer Science*, vol. 2, no. 1, pp. 1–10, Apr. 2024, doi: 10.57152/predatecs.v2i1.1094.
- [20] X. Li, X. Ma, F. Xiao, C. Xiao, F. Wang, and S. Zhang, "Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA)," *J Pet Sci Eng*, vol. 208, p. 109309, Jan. 2022, doi: 10.1016/j.petrol.2021.109309.

- 
- [21] M. Yurtsever, "Gold Price Forecasting Using LSTM, Bi-LSTM and GRU," *European Journal of Science and Technology*, vol. 31, no. 1, pp. 341–347, Dec. 2021, doi: 10.31590/ejosat.959405.
- [22] H. Crisóstomo de Castro Filho et al., "Rice Crop Detection Using LSTM, Bi-LSTM, and Machine Learning Models from Sentinel-1 Time Series," *Remote Sens (Basel)*, vol. 12, no. 16, p. 2655, Aug. 2020, doi: 10.3390/rs12162655.
- [23] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos Solitons Fractals*, vol. 140, p. 110212, Nov. 2020, doi: 10.1016/j.chaos.2020.110212.
- [24] Yahoo Finance, "Historical Gold Prices."
- [25] M. Javed Awan, M. Shafry Mohd Rahim, H. Nobanee, A. Munawar, A. Yasin, and A. Mohd Zain Azlanmz, "Social Media and Stock Market Prediction: A Big Data Approach," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2569–2583, 2021, doi: 10.32604/cmc.2021.014253.
- [26] A. Boukerche and J. Wang, "Machine Learning-based traffic prediction models for Intelligent Transportation Systems," *Computer Networks*, vol. 181, p. 107530, Nov. 2020, doi: 10.1016/j.comnet.2020.107530.
- [27] J. Awwalu and F. Ogwueleka, "On Holdout and Cross Validation: A Comparison between Neural Network and Support Vector Machine," *International Journal of Trend in Research and Development*, vol. 6, no. 2, pp. 2394–9333, Jun. 2019.
- [28] N. Zhai, P. Yao, and X. Zhou, "Multivariate Time Series Forecast in Industrial Process Based on XGBoost and GRU," in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, IEEE, Dec. 2020, pp. 1397–1400. doi: 10.1109/ITAIC49862.2020.9338878.
- [29] S. B. Primananda and S. M. Isa, "Forecasting Gold Price in Rupiah using Multivariate Analysis with LSTM and GRU Neural Networks," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 2, pp. 245–253, Mar. 2021, doi: 10.25046/aj060227.
- [30] L. Bi, G. Hu, M. M. Raza, Y. Kandel, L. Leandro, and D. Mueller, "A Gated Recurrent Units (GRU)-Based Model for Early Detection of Soybean Sudden Death Syndrome through Time-Series Satellite Imagery," *Remote Sens (Basel)*, vol. 12, no. 21, p. 3621, Nov. 2020, doi: 10.3390/rs12213621.
- [31] F. Meng, T. Song, D. Xu, P. Xie, and Y. Li, "Forecasting tropical cyclones wave height using bidirectional gated recurrent unit," *Ocean Engineering*, vol. 234, pp. 1–11, Aug. 2021, doi: 10.1016/j.oceaneng.2021.108795.
- [32] P. Li et al., "Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation," *ISPRS Int J Geoinf*, vol. 9, no. 11, p. 635, Oct. 2020, doi: 10.3390/ijgi9110635.
- [33] N. T. Luchia, E. Tasia, I. Ramadhani, A. Rahmadyan, and R. Zahra, "Performance Comparison Between Artificial Neural Network, Recurrent Neural Network and Long Short-Term Memory for Prediction of Extreme Climate Change," *Public Research Journal of Engineering, Data Technology and Computer Science*, vol. 1, no. 2, pp. 62–70, Feb. 2024, doi: 10.57152/predatecs.v1i2.864.
- [34] A. Alsharaf, K. Aggarwal, Sonia, M. Kumar, and A. Mishra, "Review of ML and AutoML Solutions to Forecast Time-Series Data," *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5297–5311, Nov. 2022, doi: 10.1007/s11831-022-09765-0.
- [35] S. Suryadibrata, "Gold Price Prediction in COVID-19 Era," *International Journal of Computational Intelligence in Control Copyrights @Muk Publications*, vol. 13, no. 2, pp. 1–4, 2021.
- [36] N. N. Aung, J. Pang, M. C. H. Chua, and H. X. Tan, "A novel bidirectional LSTM deep learning approach for COVID-19 forecasting," *Sci Rep*, vol. 13, no. 1, p. 17953, Oct. 2023, doi: 10.1038/s41598-023-44924-8.
- [37] J. Yin, Z. Deng, A. V. M. Ines, J. Wu, and E. Rasu, "Forecast of short-term daily reference evapotranspiration under limited meteorological variables using a hybrid bi-directional long short-term memory model (Bi-LSTM)," *Agric Water Manag*, vol. 242, Dec. 2020, doi: 10.1016/j.agwat.2020.106386.
- [38] S. Aryal, D. Nadarajah, D. Kasthurirathna, L. Rupasinghe, and C. Jayawardena, "Comparative analysis of the application of Deep Learning techniques for Forex Rate prediction," in *2019 International Conference on Advancements in Computing (ICAC)*, IEEE, Dec. 2019, pp. 329–333. doi: 10.1109/ICAC49085.2019.9103428.
- [39] C. Arthur, N. Yudistira, and C. Dewi, "AutoCyclic: Deep Learning Optimizer for Time Series Data Prediction," *IEEE Access*, vol. 12, pp. 14014–14026, 2024, doi: 10.1109/ACCESS.2024.3356553.
- [40] R. Llugsi, S. El Yacoubi, A. Fontaine, and P. Lupera, "Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito," in *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, IEEE, Oct. 2021, pp. 1–6. doi: 10.1109/ETCM53643.2021.9590681.
-

- [41] C. Christodoulos, C. Michalakelis, and D. Varoutas, "On the combination of exponential smoothing and diffusion forecasts: An application to broadband diffusion in the OECD area," *Technol Forecast Soc Change*, vol. 78, no. 1, pp. 163–170, Jan. 2011, doi: 10.1016/j.techfore.2010.08.007.
- [42] X. Zhang, T. Zhang, A. A. Young, and X. Li, "Applications and Comparisons of Four Time Series Models in Epidemiological Surveillance Data," *PLoS One*, vol. 9, no. 2, p. e88075, Feb. 2014, doi: 10.1371/journal.pone.0088075.
- [43] Z. Hu, Y. Zhao, and M. Khushi, "A Survey of Forex and Stock Price Prediction Using Deep Learning," *Applied System Innovation*, vol. 4, no. 1, p. 9, Feb. 2021, doi: 10.3390/asi4010009.
- [44] Suryani and Mustakim, "An Intelligent Chatbot for Faculty Administration Using Bidirectional LSTM and Seq2Seq Architecture," *2024 International Conference on Smart Computing, IoT and Machine Learning (SIML)*, Surakarta, Indonesia, 2024, pp. 226-231, doi: 10.1109/SIML61815.2024.10578161.