Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 3 (2024) 377 - 388 e-ISSN: 2580-0760

Comparative Analysis of Recurrent Neural Network Models Performance in Predicting Bitcoin Prices

Zidane Ikkoy Ramadhan[1*], Harya Widiputra[2]
[1,2]Department of Informatics, Faculty of Information Technology, Perbanas Institute Jakarta, Jakarta, Indonesia
[1]ikkoy123@gmail.com, [2]harya@perbanas.id

Abstract

Recurrent Neural Network is a Deep Learning algorithm that is commonly used to develop prediction systems. There are many variants of RNN such as RNN itself, Long Short Term Memory (LSTM), and Gated Recurrent Unit, so it is frequently debatable which algorithm from the RNN family has the most optimal efficiency and computation time. When developing a prediction system, sequential data or time-series data is required so that an accurate prediction can be made. Sequential or time-series data involves data arranged in time sequence, such as weather data, financial data, carbon emission data and traffic data recorded over time. This research will be carried out by predicting the three RNN models against historical Bitcoin value data. The research method used is Experimental Design by comparing the performance between the three models on bitcoin value time series data, testing is done by involving hyperparameters such as Tanh, Sigmoid and ReLU activation functions, batch size, and epochs. The aim of this research is to find out which RNN model can produce the most optimal performance and find out what performance measures can be used to evaluate and compare the performance between the three models. The results of the study show that LSTM is the most effective model with RMSE 0.012441 and MSE 0.000155 but inefficient because it takes 3 minutes 24 seconds to run the computation, in the meantime Tanh activation function gives the most optimal prediction than Sigmoid and RelU and therefore should be the main candidate to be used with RNN models when predicting Bitcoin prices.

Keywords: deep learning; recurrent neural network; LSTM; GRU; bitcoin

How to Cite: Z. I. Ramadhan and H. Widiputra, "Comparative Analysis of Recurrent Neural Network Models Performance in Predicting Bitcoin Prices", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 8, no. 3, pp. 377 - 388, Jun. 2024.
DOI: https://doi.org/10.29207/resti.v8i3.5810

1. Introduction

Throughout the years, Deep Learning technology has been helping to solve complex problems efficiently and has become the foundation of modern Artificial Intelligence. One of the architectures of Deep Learning itself is the Recurrent Neural Network. Recurrent Neural Network or RNN is a widely practised model in forecasting. RNN can perform similar roles for each part of its sequence, and its performance depends on the previous computation [1] the most popular RNN models are the Gated Recurrent Unit and Long Short Term Memory. RNN has assisted many researchers in conducting research such as prediction [2], understanding, and generating natural language [3] along with processing sequential data such as text recognition [4], audio, and video [5], [6].

Time series data is a collection of data that is obtained through observations and is chronological with a certain time interval [7]. Time series are sequential data that usually has a large amount of data and is always up to date from time to time. An example of time-series data can be financial data such as stock prices [8] crude oil prices [9], and currency exchange rates. On the other hand, time-series data can also be non-financial data such as weather data [10], carbon emissions [11], and traffic flow [12]. Bitcoin is one of the financial sequential data where data is captured from time to time. Bitcoin has a unique characteristic in that its price is highly volatile [13], [14]. Due to the volatility of bitcoin's value, several factors have a significant impact on its value, such as the popularity of crypto itself, market trends, legalization, regulation, supply and demand, and the cost of transactions even fake news [15]. The volatile sequential data on Bitcoin is a good topic of discussion for research in predicting the value prices of Bitcoin. The structural diagram of how RNN works with Bitcoin datasets can be seen in Figure 1,

Received: 15-05-2024 | Accepted: 18-06-2024 | Published Online: 20-06-2024

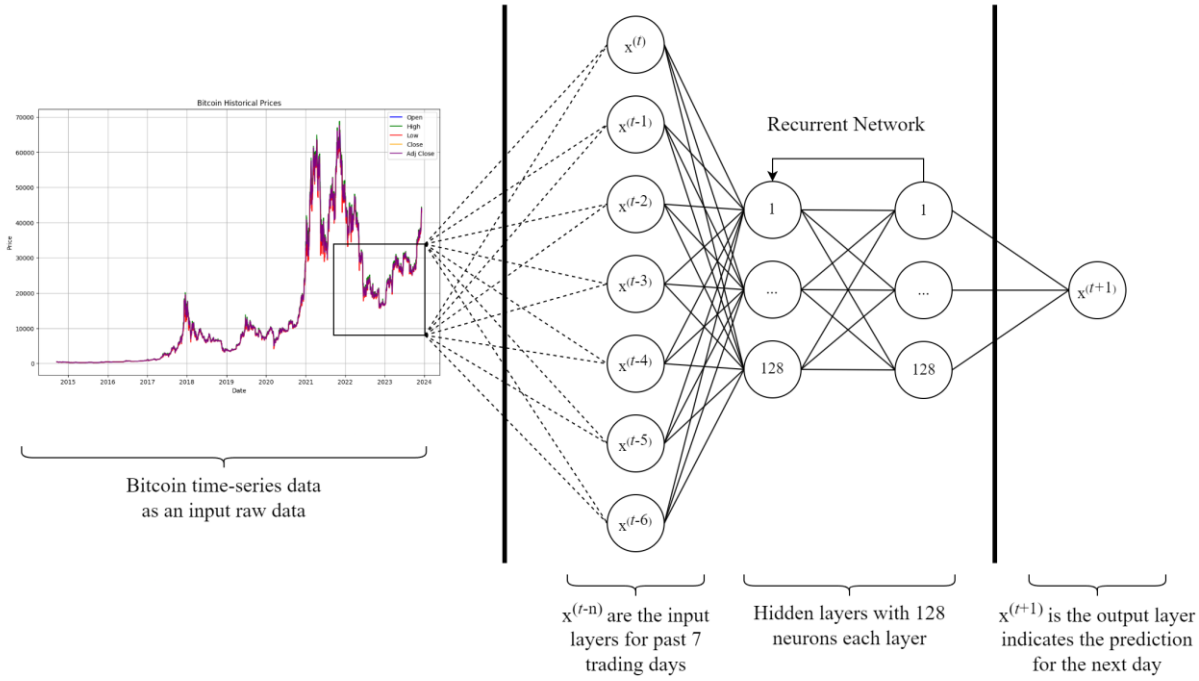

Figure 1 RNN Model Designed for Predicting Bitcoin

where the Bitcoin prices represent the raw data input for the model. The prices over time are utilized to train the model to predict future prices, then the input layers which consist of the past 7 days as timesteps are fed into an RNN with two hidden layers with each layer containing 128 neurons, and then the hidden layers process the input data and extract the features relevant for predicting future prices. Neuron in the hidden layers is interconnected and connected to the neurons in the output layer. Overall, at the end, the prediction is going to predict the next day from the past 7 days.

The RNN algorithm has many variants such as the RNN itself commonly known as traditional RNN. and other popular models that are widely used in research such as Long Short Term Memory and Gated Recurrent Unit. Long Short Term Memory commonly abbreviated as LSTM is the development of a Recurrent Neural Network or RNN which has a strong capacity in complex non-linear interactions and can handle gradient challenges other than that LSTM has the advantage of predicting time-series data [16], [17]. Whereas the Gated Recurrent Unit is another model that evolved from the Recurrent Neural Network, GRU was created to allow each recurrent unit to dynamically understand the dependencies across multiple time scales. Similar to LSTM, GRU also has unit gates that regulate the flow of unit information, but without having separate memory cells. [18].

The number of RNN derivatives and the RNN algorithm itself led to the need for understanding the performance comparison between the three models, by comparing the advantages and disadvantages of each variant of the algorithms. There are several previous research that have been conducted in predicting Bitcoin price using RNN, GRU, and LSTM. Researchers [19] conducted a study using long short-term memory by using 80% data allocation for training and 20% for testing and achieved an RMSE of 0.111 on LSTM with the number of hidden layers 360 and 500 epochs, according to the researchers LSTM is able to predict the price of bitcoin but it needs special treatments for the characteristics of the data set.

Researchers [20] This research was conducted using three Deep Neural Network Methods where LSTM gave the best results for USDT-USD and BNB-USD with an average of 0.0025 for MAE and 0.0034 for RMSE on USDT-USD, on the other hand, Bi-LSTM gave the best results at 100.1383/147.8453 for ETH/USD. While BTC-USD and ADA-USD GRU gave the best results at 1167.3462/1777.3070 and 0.0782/01134. When the performance evaluation was measured using MAPE, GRU gave better results than the three RNNs with a MAPE score of 0.0447. This study also tested the shortest time for the model to execute, where GRU has the shortest time at 7.5791 and LSTM at 7.9768, while Bi-LSTM has the longest execution time than the two RNNs which is 10.8720.

Researchers [21] researched three cryptocurrency prices which are BTC, LTC, and ETH. In this research, GRU has better performance on all Crypto compared to LSTM and Bi LSTM. Using MAPE accuracy performance, GRU produces a performance of 0.2454% on BTC, 0.8267% on ETH and 0.2116% on LTC. LSTM produces MAPE performance of 1.1234% on BTC 1.5498 on ETH and 0.8474% on LTC While Bi-LSTM produces the lowest prediction value compared to GRU and LSTM where the value is 5.990%, 6.85% and 2.332% for BTC, ETH and LTC.

Researchers [22] This research uses LSTM with an epoch parameter of 10 a learning rate of 0.001 and a

batch size of 25. Resulting in an RMSE evaluation value of 77.74 and MAE of 278.33. Researchers [23] This research focuses on Bitcoin price prediction and uses Neural Networks, GRU, and LSTM. GRU with a Hyperparameter dropout rate of 0.1 resulted in an RSME of 0.014. Researcher [24] Conducting research with the number of epochs 1, 10 and 20 in each session, the results of this study are the type of DOGE coin with the number of Epochs 20 results in an RMSE value of 0.0630.

From these various literature reviews, we can draw a preliminary conclusion that there are different perspectives on the performance and efficiency of the RNN model when it comes to predicting Cryptocurrency. This is indicated by the diversity of data sets and the form of models made by previous researchers. There are no models that consistently have absolute superiority, and each model has its own advantages and disadvantages. So, the a need for further research to compare models, especially on bitcoin data.

In this research, we will analyze the performance of the three RNN algorithms namely traditional RNN, GRU, and LSTM to evaluate which model is the most efficient and effective, and provides the highest accuracy value, the selection of Bitcoin as a data set due to its high volatility and chaotic characteristic. So later in this research will configure each model with parameters such as activation function, batch size, and epochs. Furthermore, the evaluation metrics used are Root Mean Squared Error and Mean Squared Error because these metrics can provide insight into how close the model's predictions are to the true value. In the end, this research is aimed to select the model that best suits the specific needs of the problem at hand, both in accuracy, and computational efficiency in predicting the closing value of Bitcoin. Therefore, the main objective of this research is to find out which RNN model has the most effective and efficient performance, as well as the highest accuracy in predicting Bitcoin prices, and see what performance measures can be used to compare the performance between the three RNN models, specifically the traditional RNN, LSTM, and GRU. To provide the best model, we will use daily Bitcoin data collected from Yahoo! Finance, consisting of 3370 trading days from September 17, 2014, to December 8, 2023.

2. Research Methods

Figure 2 illustrates the sequential flow of research methods that will be executed throughout this study. The research process unfolds across three main stages: data collection, data preprocessing, and model performance evaluation. These stages form the foundational framework upon which our investigation into the dataset's characteristics, especially for the Bitcoin data set and predictive modelling techniques will unfold. Each stage is meticulously designed to ensure methodological rigour and facilitate meaningful insights into the research objectives.

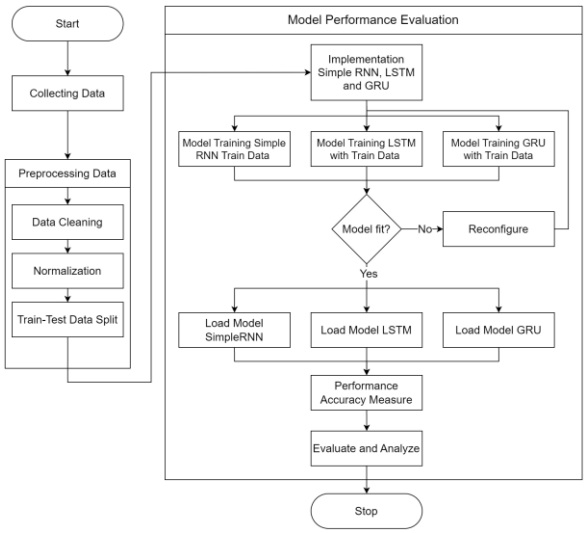

Figure 2 Processing Stages

The first step in building a deep learning model is to collect data. At this stage, the original data is collected, and the data is obtained through the Yahoo! Finance page. After the data has been collected, the data will be processed at the next stage where the data will go through a cleaning and normalization process, in this stage the data will be cleaned to remove irrelevant or unnecessary information, so the research is more focused, then do normalization. Large amounts of data are highly vulnerable, affecting the algorithm's performance and prediction outcomes. Therefore, normalization needs to be done to transform the data into an appropriate range. In this study, normalization will be carried out using Min-Max Scaler where the value will be changed linearly from the original data to a scale of 0 to 1 using Formula 1.

$$\tilde{x} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

$\tilde{x}$ is the result of normalization, x_i is the initial data x_{min} is the minimum value of the original data and x_{max} is the maximum value of the original data.

The next stage of data preprocessing is to separate the data into train test data split. In this step, the data that has been through the cleaning and normalization process will be divided into two parts. The data will be divided into two sets, the first is data for training and the other for testing. in this study, the ratio used is 80% for training data and 20% for testing data.

After doing preprocessing the next stage is to evaluate performance, first implementing or building models for SimpleRNN, LSTM, and GRU. Then determine the number of layers, the number of neurons in each layer, the activation function, batch sizes, and epochs to be used. in this study, there will be nine model developments where each model gets different treatments such as using batch sizes 64, 128, and 256, epochs 100 250 500 with activation functions Tanh, Sigmoid, and ReLU, with these hyperparameters, it is

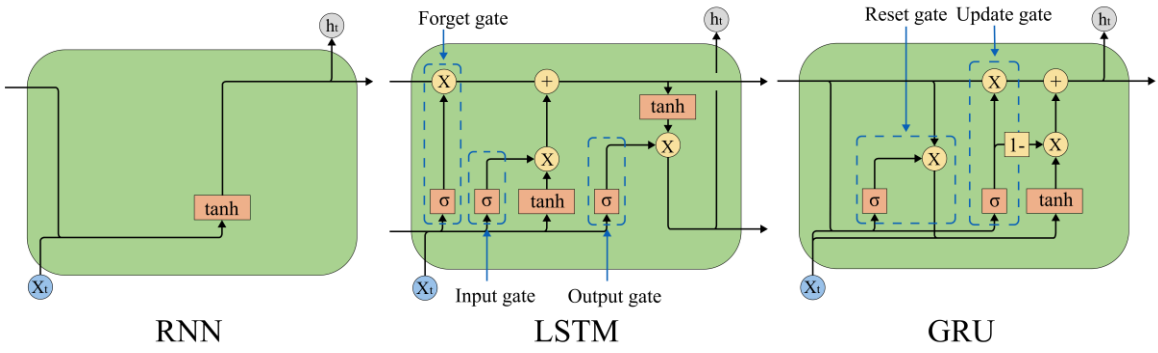

Figure 3 RNN, LSTM and GRU Structure [25]

expected that the models can provide the best prediction results effectively and efficiently.

Recurrent Neural Networks is a model often used in forecasting research, RNN models have a feedback connection between nodes and layers, which allows RNN to process input sequences of varying lengths [25], The activation of a recurrent node consists of feedback to itself from one time-step to the subsequent step [26], in the hidden layers of RNN, the structure is converted into a repeating pattern and the input data is organized during the non-linear learning process, However, the RNN is vulnerable to problems such as gradient disappearance and gradient explosion during backpropagation [27]. The structure of an RNN can be seen in Figure 3.

In the meantime, Long short-term memory in Figure 3 was proposed by Hochreiter & Schmidhuber in 1997 [28] a derivative of Recurrent Neural Network is developed to overcome problems such as gradient disappearance and gradient explosion during backpropagation within RNN, this was achieved by using a more efficient gradient-based algorithm for architecture that implements a constant error flow so that there is no explosion or disappearance through the internal state of particular units as long as the gradient calculation is intersected at certain point according to the architecture. LSTMs are very suitable for classifying, processing, and making predictions based on single time series data [29]. LSTM has 3 main gates namely input gate, output gate and forget gate. The cell remembers the value in a specific time interval and the remaining three gates regulate the flow of information associated with that cell [30]. In LSTM there are computational steps to calculate predictive time-series data starting with calculating the output value from the previous time, while the input value from the current time is used as input for the forget gate.

The processing results of the forget gate are then calculated using Formula 2.

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \quad (2)$$

f_t has a ranging from (0, 1), W_f is the weight of the forget gate, bf is the bias value, x_t is the input value of the current time and h_{t-1} is the output of the previous calculation value.

The calculation of the output value from the previous calculation and the input value of the current time is the input value for the input gate and for the output value and condition of the candidate cell of the input gate can be achieved with Formulas 3 and 4.

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = tanh(W_c . [h_{t-1}, x_t] + b_c) \quad (4)$$

with the value range for i_t is (0, 1) W_i is the weight of the input gate, b_i is the bias value of the input gate, W_c is the weight for the candidate input gate value, and b_c represents the bias value of the candidate input gate, after calculating for the condition of the candidate cell has been completed, the process continues by calculating the cell value using Formula 5.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

The range of values for C_t is (0, 1). t is a process of calculating time, the output value of h_{t-1} and the input value of xt becomes the input value for the output gate, and to calculate the gate output value can be calculated by using Formula 6.

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o) \quad (6)$$

o_t has a range of (0,1), W_o is the weight of the gate output, and the output bias value is symbolized by the variable b_o and the result of the LSTM is achieved from the output gate value and to calculate the result is by using Formula 7

$$h_t = o_t * tanh + C_t \quad (7)$$

tanh is the activation function of the LSTM which can be changed according to the requirements and characteristics of the problems to be resolved [31]. In general, there is no guideline on determining the number of layers or the number of hidden layers in LSTM. The number of layers required in LSTM depends on the complexity of the datasets [32]. LSTM has the advantage of being able to predict time patterns [17] and is specifically designed to perceive long-term dependency by changing part of the memory cell state, thus overcoming the drawbacks found in RNN [33].

The Gated Recurrent Unit or GRU which can be seen in Figure 3 proposed by Chung et al., 2014 [34] aims to maintain the safety of each recurrent unit, which can

adaptively capture different time-scale dependencies. Like LSTM units, GRUs are also provided with gates that regulate the flow of information within the unit, although without separate memory cells. GRU has a more simplified baseline structure than LSTM, thus making it easier to train and less computationally intensive [35]. The GRUs structure can be seen in Figure 3 In GRU there are several steps in performing computational calculations which are described in Formula 8.

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h}_t \quad (8)$$

h_t is the activation of the GRU at time t which interpolates between the previous activation h_{t-1} and the candidate activation $\tilde{h}_t$, where the update gate z_t decides how many units to update the activation or its content. The update gate can be computed with Formula 9.

$$Z_t = \sigma(W_z x_t) + U_z h_{t-1} \quad (9)$$

In this procedure, the linear computation between the existing state and the newly computed state is similar to the unit of LSTM. However, in GRU there is no mechanism to control the degree of exposed states, but rather exposes all parts of the state at each time. The activation candidate $\tilde{h}_t$ is calculated and computed similar to the traditional recurrent unit which is obtained by Formula 10.

$$h_t = tanh(Wx_t + U(r_t \odot h_{t-1})) \quad (10)$$

r_t is the reset gates set and the symbol $\odot$ is the element of the multiplication. When closed (r_t is close to zero), the reset gate will effectively make the unit act as if it is reading the first symbol of the input circuit, thus allowing it to forget the pre-calculated state. The reset gate r_t is calculated similarly to the update gate by using Formula 11.

$$r_t = \sigma(W_r x_t) + U_r h_{t-1} \quad (11)$$

However, the GRU model still has several problems such as slow convergence rate and low learning efficiency, which can result in a long training time and may even cause underfitting [36].

The last stage is to perform performance evaluation metrics or error measures. This stage is a vital component in making predictions. The evaluation metrics that will be used are Root Means Squared error and Mean Squared Error. The RMSE value can be achieved using Formula 12.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (12)$$

$\hat{y}_i$ is the predicted value at a specific time point and y_i is the actual value. RMSE measures each sequence and compares each model, the smaller RMSE results the higher the accuracy of the prediction.

Mean Squared Error (MSE) is the average of the squared errors between the actual and predicted values. This is a commonly used method to check the value of the estimated error in forecasting. Mean Squared Error can be achieved with Formula 13.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (13)$$

$\hat{y}_i$ is the value of the prediction at a certain point in time and yi is the actual value. actual value. The MSE value, which is close to zero indicates that the forecasting results are in line with the actual data values.

3. Results and Discussions

The data will be used to carry out training and testing, which obtained through the Yahoo! finance website, this data is the value of Bitcoin recorded in a daily period with a total of 3370 trading days starting from September 17, 2014, to December 9, 2023, there are several fields such as date, open, high, low, close, adj close and volume. However, this research will be conducted univariate where bitcoin predictions will only learn the pattern of movement of the value itself in the past. which later the model will perform forecasting for the past 7 days to predict the next day.

3.1 Results

In the preceding chapter, we delved into the intricate modelling nuances of SimpleRNN, LSTM, and GRU architectures. In this chapter, we undertake a comparative analysis of these models by systematically adjusting hyperparameters such as activation functions, batch sizes, and epochs. The configurations for RNN parameters are outlined in Table 1 by providing details about those hyperparameters.

Table 1 RNN Parameters

Parameter	Value
RNN layer Activation	Tanh, Sigmoid, ReLU
RNN Hidden Unit	128
Dense Layer	32
Dense Layer	1
Batch Size	64, 128, 256
Optimizers	Adam
Loss Function	MAE
Epochs	100, 250, 500

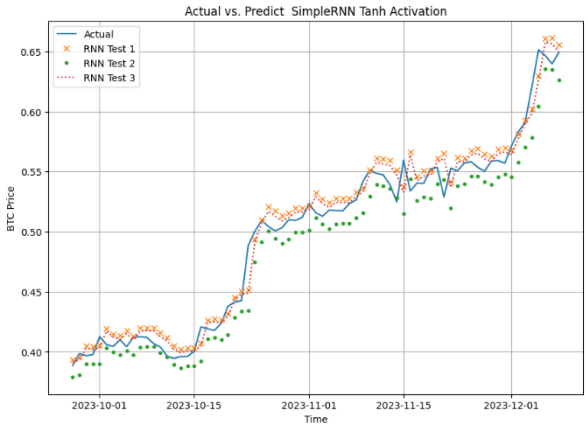

Figure 4 Actual vs. Predict RNN Tanh Activation

After carrying out the implementation using the parameters mentioned in the table above, the next step is to test the model and make predictions.

In Figure 4, the RNN model with Tanh activation was tested in three different trials. In the first experiment, the model was tested using a batch size of 64 and trained for 100 epochs. In the second experiment, the test was conducted with a batch size of 128 and trained for 250 epochs. While in the third experiment, the test used a batch size of 256 and was trained for 500 epochs.

Table 2 Experiment Result of RNN with Tanh Activation

Test	Epoch	Batch Size	Tanh		Execution Time
			RMSE	MSE	
1	100	64	0.014788	0.000219	42.4s
2	150	128	0.015024	0.000226	1m 23s
3	500	256	0.013445	0.000181	1m

Table 2 provides a more granular breakdown of the RMSE, MSE, and Execution Time metrics. Our analysis revealed intriguing insights across the three experiments conducted. In the initial and subsequent trials, the model demonstrated remarkable accuracy, closely aligning predicted values with their originals.

Notably, the first experiment boasted an execution time of approximately 42.4 seconds, while the third experiment slightly extended to a minute, yet still within reasonable bounds.

Conversely, the second experiment unveiled a significant deviation between actual and predicted values, despite the longest execution time clocking in at 1 minute and 23 seconds, therefore the second test doesn't provide optimal results.

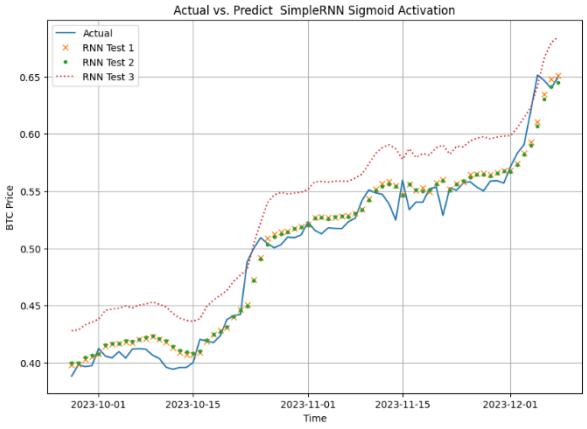

Figure 5 Actual vs. Predict RNN Sigmoid Activation

Table 3 Experiment Result of RNN with Sigmoid Activation

Test	Epoch	Batch Size	Sigmoid		Execution Time
			RMSE	MSE	
1	100	64	0.017945	0.000322	1m 23s
2	150	128	0.018860	0.000356	1m 23s
3	500	256	0.042982	0.001847	1m 3s

Figure 5 shows the comparison between the actual and predicted values of the RNN model with Sigmoid activation. In contrast to the results of the test using Tanh activation, the first test was conducted with a batch size of 64 for 100 epochs, while the second test used a batch size of 128 for 250 epochs.

The results show that the use of Sigmoid activation does not provide an optimal level of accuracy, but the execution time generated in the first and second tests is the same, which is 1 minute 23 seconds.

However, when performing the third test using a batch size of 256 and 500 epochs, despite having the lowest execution time of 1 minute 3 seconds, the prediction results far exceeded the actual values. Full details of the RMSE, MSE, and execution time are provided in Table 3. Figure 6 shows the comparison between the actual and predicted values of the RNN model with ReLU activation.

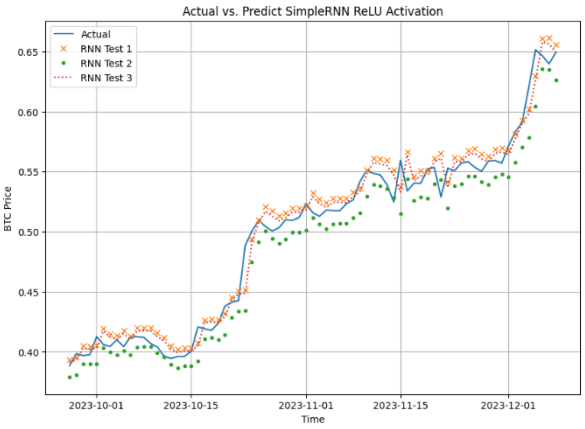

Figure 6 Actual vs. Predict ReLU Activation

In the first test, using batch size 64 for 100 epochs, the predicted values are close to the results of the third test with batch size 256 and epoch 500. However, the execution time of the first test is shorter, 42.1 seconds, while the third test requires 1 minute 23 seconds. In the second test with batch size 128 and epoch 250, the predicted graph weakened and was below the actual value. Although it has the same execution time as the third test, which is 1 minute 23 seconds.

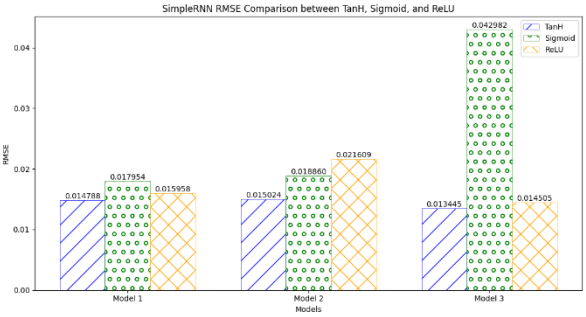

Figure 7 RMSE RNN Comparison Between Tanh, Sigmoid, ReLU

Table 4 presents a detailed breakdown of RMSE, MSE and execution time. Meanwhile Figures 7 and 8 show a comparison of RMSE and MSE between three activation functions. The next phase involves applying and assessing the dataset utilizing LSTM architecture. Following a similar protocol to our RNN experiments,

LSTM will undergo testing with identical parameter configurations.

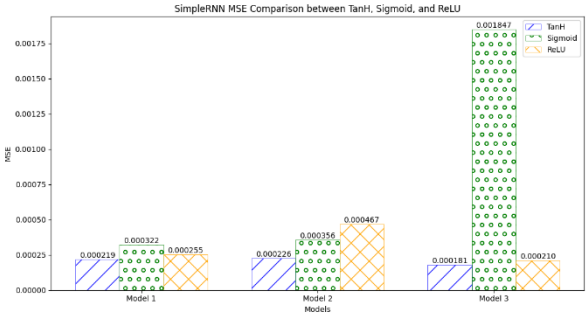

Figure 8 RNN MSE Comparison Between Tanh, Sigmoid, ReLU

Table 4 Experiment Result of RNN with ReLU Activation

Test	Epoch	Batch Size	ReLU		Execution Time
			RMSE	MSE	
1	100	64	0.015958	0.000255	42.1s
2	150	128	0.021609	0.000467	1m 23s
3	500	256	0.014505	0.000210	1m 23s

However, this time, the focus shifts to the LSTM algorithm. Table 5 outlines the configuration of the LSTM Parameter

Table 5 LSTM Parameters

Parameter	Value
RNN layer Activation	Tanh, Sigmoid, ReLU
RNN Hidden Unit	128
Dense Layer	32
Dense Layer	1
Batch Size	64, 128, 256
Optimizers	Adam
Loss Function	MAE
Epochs	100, 250, 500

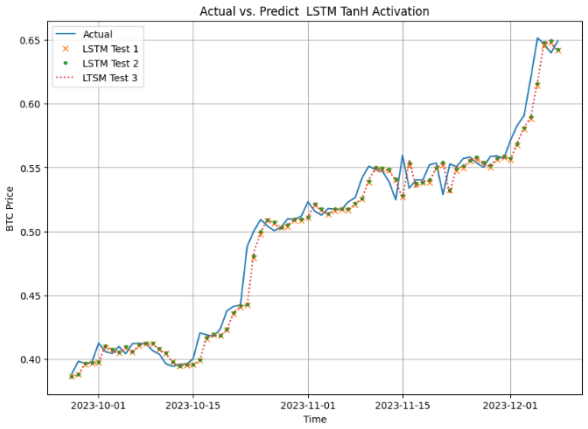

Figure 9 Actual vs. Predict LSTM Tanh Activation

Figure 9 shows the prediction results of LSTM against Tanh activation. The first test was conducted using a batch size of 64 with 100 epochs, the second test batch size of 128 and 250 epochs, and the third test was conducted with a batch size of 256 and 500 epochs.

In the graph, the three models produce predictions that are not too significant compared to the actual values. However, there is a variation in execution time between the three models. The first run took 1 minute 30 seconds, the second run took 2 minutes 26 seconds, and the third run took 3 minutes 24 seconds. Detailed results of RMSE and MSE as well as execution time can be seen in Table 6.

Table 6 Experiment Result of LSTM with Tanh Activation

Test	Epoch	Batch Size	Tanh		Execution Time
			RMSE	MSE	
1	100	64	0.012626	0.000159	1m 30s
2	150	128	0.012557	0.000158	2m 26s
3	500	256	0.012441	0.000155	3m 24s

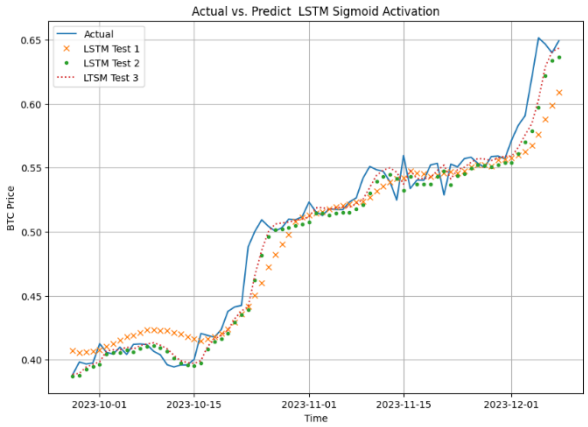

Figure 10 Actual vs. Predict LSTM Sigmoid Activation

Figure 10 presents the results of the LSTM prediction against Sigmoid activation. The first test was carried out with a batch size configuration of 64 with an epoch of 100, the second test used a batch size of 128 and an epoch of 250, and the third test used a batch size of 256 and an epoch of 250.

In the first test, the results of the predicted and actual values given were very far away but had the fastest execution time of 36.4 seconds. When the second and third test results are obtained, the predicted value is close to the actual value even though there is a difference in execution time where the second test takes 1 minute 21 seconds while the third test lasts for 1 minute 51 seconds. Details of RMSE, MSE and execution time can be seen in Table 7.

Table 7 Experiment Result of LSTM with Sigmoid Activation

Test	Epoch	Batch Size	Sigmoid		Execution Time
			RMSE	MSE	
1	100	64	0.027474	0.000755	36.4s
2	150	128	0.014578	0.000213	1m 21s
3	500	256	0.014463	0.000209	1m 51s

Figure 11 provides the prediction results of LSTM against ReLU activation. The first test was conducted using a batch size of 64 with an epoch of 100, the second test a batch size of 128 and epoch 250, and the third test was conducted with a batch size of 256 and epoch 500. In the first test, the use of ReLU did not produce satisfactory performance, with an execution time of 1 minute 23 minutes.

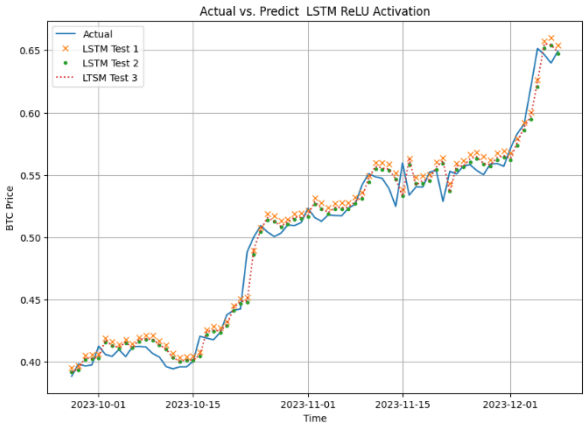

Figure 11 Actual vs. Predict LSTM ReLU Activation

However, there were improved results in the second and third tests, where the predicted results were almost close to the actual values. The third test had a faster execution time compared to the second test, with execution times of 2 minutes 11 seconds and 2 minutes 23 seconds respectively.

More details about the results of RMSE, MSE and execution time can be found in Table 8, and the comparison of RMSE and MSE on the three activations can be seen in Figure 12 and Figure 13.

Table 8 Experiment Result of LSTM with ReLU Activation

Test	Epoch	Batch Size	ReLU		Execution Time
			RMSE	MSE	
1	100	64	0.027474	0.000755	36.4s
2	150	128	0.014578	0.000213	1m 21s
3	500	256	0.014463	0.000209	1m 51s

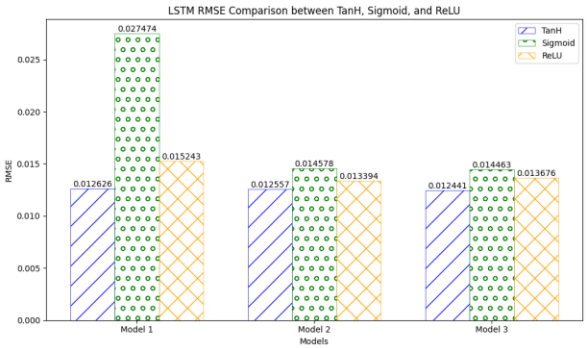

Figure 12 RMSE LSTM Comparison Between Tanh, Sigmoid ReLU

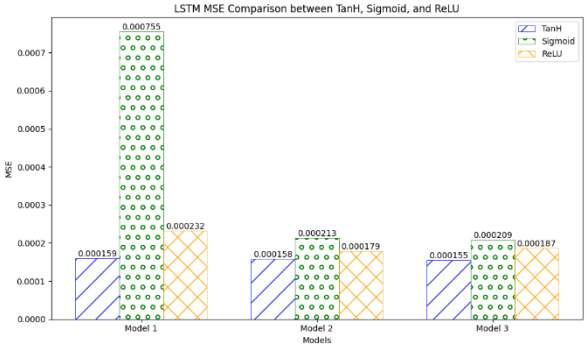

Figure 13 MSE Comparison Between Tanh, Sigmoid, ReLU

The final stage of the research experiment entails constructing the third model, GRU. Similar to approaches with RNN and LSTM, the next procedure will vary several parameters to identify the most optimal configuration. Table 9 shows the parameters of the GRU model test and compared in this research.

Table 9 GRU Parameters

Parameter	Value
RNN layer Activation	Tanh, Sigmoid, ReLU
RNN Hidden Unit	128
Dense Layer	32
Dense Layer	1
Batch Size	64, 128, 256
Optimizers	Adam
Loss Function	MAE
Epochs	100, 250, 500

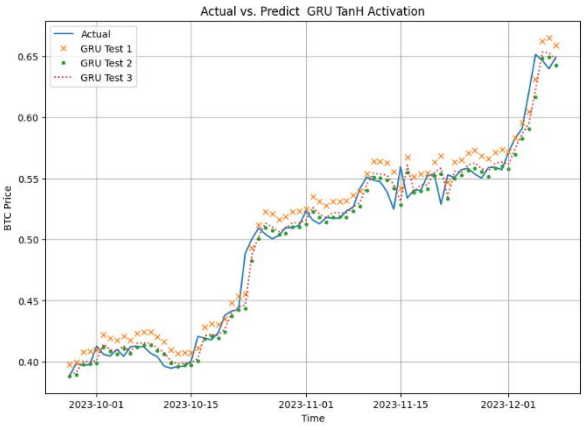

Figure 14 Actual vs. Predict GRU Tanh Activation

Figure 14 depicts the prediction results of GRU against Tanh activation. The first test was conducted using a batch size of 64 with epoch 100, the second test batch size of 128 and epoch 250, and the third test was conducted with a batch size of 256 and epoch 500.

Table 10 Experiment Result of GRU with Tanh Activation

Test	Epoch	Batch Size	Tanh		Execution Time
			RMSE	MSE	
1	100	64	0.017145	0.000294	54.7s
2	150	128	0.012568	0.000158	1m 25s
3	500	256	0.012845	0.000165	1m 23s

The first test has results that are far from the actual value and has the fastest execution time of 54.7 seconds, followed by the second test with an execution time of 1 minute 25 seconds and the third test with 2 minutes 23 seconds. The prediction result that is closest to the actual value is obtained in the second test. Details about RMSE and MSE, as well as execution time can be seen in Table 10.

Figure 15 shows the prediction results of GRU towards Sigmoid activation. The first test was conducted using a batch size of 64 with an epoch of 100, the second test batch size of 128 and an epoch of 250, and the third test was conducted with a batch size of 256 and an epoch of 500.

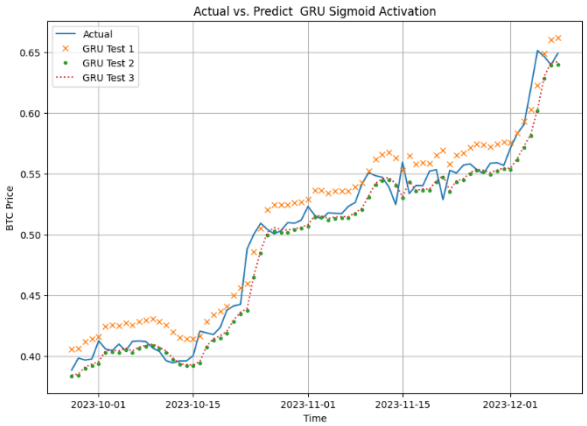

Figure 15 Actual vs. Predict GRU Sigmoid Activation

Using Sigmoid activation does not provide optimal results in the first test the prediction results are above the actual value with an execution time of 1 minute 23 seconds, while in the second and third tests, the graph is below the actual value with a test time of 1 minute 11 seconds and 1 minute 40 seconds respectively, where the second test has the fastest execution time compared to the first and third tests. The details of RMSE, MSE and Execution time can be seen in Table 11.

Table 11 Experiment Result of GRU with Sigmoid Activation

Test	Epoch	Batch Size	Sigmoid		Execution Time
			RMSE	MSE	
1	100	64	0.017145	0.000294	54.7s
2	150	128	0.012568	0.000158	1m 25s
3	500	256	0.012845	0.000165	1m 23s

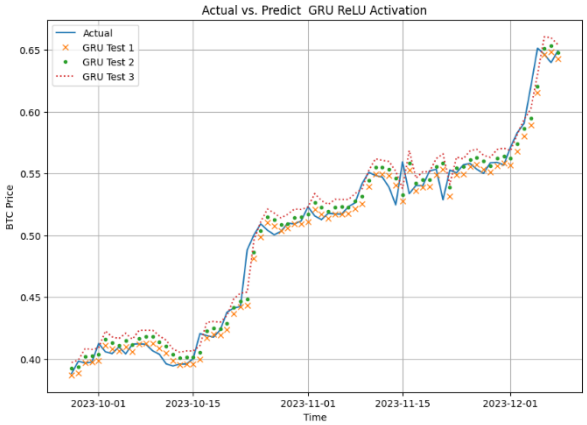

Figure 16 Actual vs. Predict GRU ReLU Activation

Figure 16 depicts the prediction results of GRU against ReLU activation. The first test was conducted using a batch size of 64 with epoch 100, the second test batch size of 128 and epoch 250, and the third test was conducted with a batch size of 256 and epoch 500. In ReLU activation, the best results were achieved in the first test with an execution time of 1 minute 22 seconds, which was followed by the second test with the same time as the first test of 1 minute 22 seconds, but there was a significant change in the third test with the longest execution time taking 2 minutes 23 seconds.

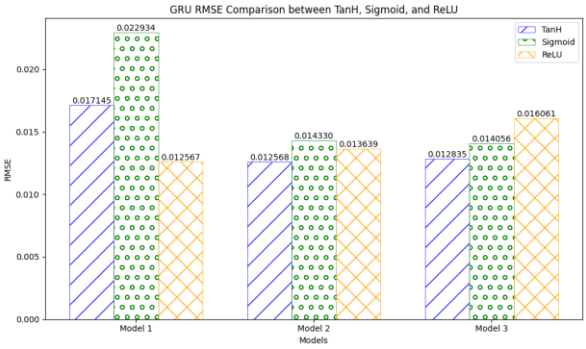

Figure 17 RMSE GRU Comparison Between Tanh, Sigmoid ReLU

Information on RMSE and MSE, as well as execution time can be seen in Table 12. In addition, the comparison of RMSE and MSE between the three activations can be seen in Figures 16 and 17.

Table 12 Experiment Result of GRU with ReLU Activation

Test	Epoch	Batch Size	ReLU		Execution Time
			RMSE	MSE	
1	100	64	0.012567	0.000158	1m 22s
2	150	128	0.013639	0.000186	1m 25s
3	500	256	0.016061	0.000258	2m 23s

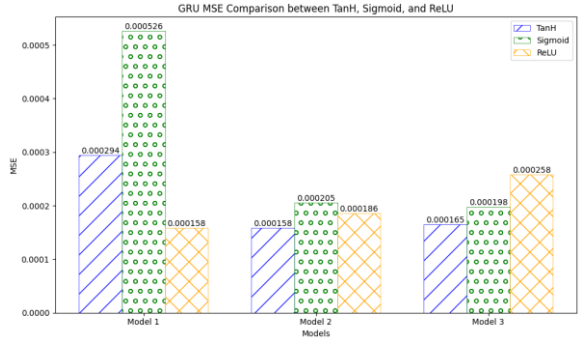

Figure 18 MSE GRU Comparison Between Tanh, Sigmoid ReLU

3.2 Discussions

Different treatment of parameters such as the number of epochs, activation function and batch size can change the results very significantly, the models that have been tested have gotten the best attention, in LSTM, the best model is to use Tanh activation compared to Sigmoid and ReLU.

The third model LSTM with the number of epochs 500, as well as batch size 256, produces an RMSE value of 0.012441 and MSE 0.000155 but has the disadvantage that the execution time is very long at 3 minutes 24 seconds, this is because LSTM has a high complexity algorithm in performing the learning process at each iteration and meanwhile RNN and GRU provide shorter execution time compared to LSTM.

RNN reaches its best model in the third model with epochs of 500, batch size 256 getting RMSE results of 0.013445 and MSE of 0.000181 with an execution time of 1 minute, using Tanh activation, then followed by GRU with the second model where there are differences between selecting parameters for LSTM and RNN

because GRU only requires epochs of 250 and batch size 128 with an execution time of 1 minute 25 seconds to reach its optimum and getting RMSE results of 0.12568 and MSE 0.000158 using Tanh activation.

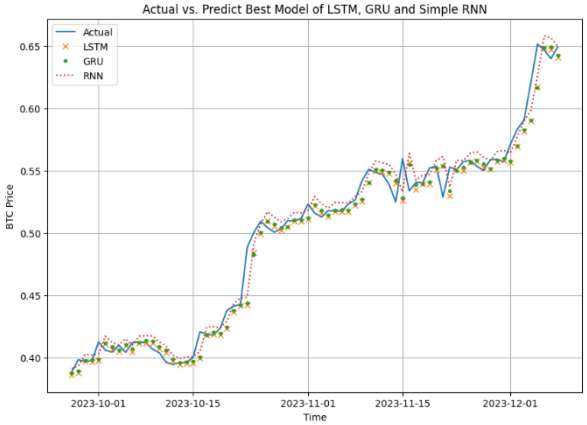

Figure 19 Actual vs. Predict Best Model of Three Models

RNN can execute quickly due to its simple algorithm structure compared to LSTM. On the other hand, although GRU is more complex than RNN, it has a more compact structure and some reliable mechanisms for regulating the flow of information, which ultimately results in a faster execution time than LSTM but still slightly slower than RNN.

LSTM is the model that performs most effectively but not very efficiently this is because LSTM has the smallest error value but has the longest execution time compared to GRU providing computationally efficient results, on the other hand, RNN which has the fastest execution time has the highest error value compared to both algorithms.

This actuality can be seen in Figure 19 which shows a comparison of the best between three models of RNN, LSTM and GRU against Tanh activation. Tanh is the most optimal activation compared to Sigmoid and ReLU. Although LSTM has a lag between actual and predicted values, it can be close to actual data, on the other hand, GRU is almost close to the performance of LSTM this is due to the simpler structure of GRU compared to LSTM utilizing fewer gating mechanisms, which can reduce computational complexity.

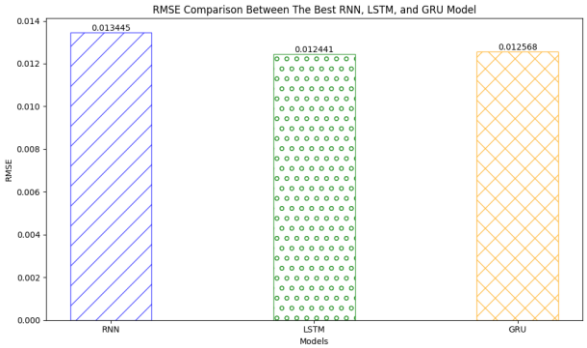

Figure 20 RMSE Comparison Best of Three Models

Figure 21 MSE Comparison Best of Three Models

Figure 20 and Figure 21 show the best three models with each using Tanh activation being analyzed on the RMSE and MSE score for each prediction.

Based on the plotted error values between the three models, an initial conclusion can be made that the model that has the smallest error value is LSTM followed by GRU and RNN, in terms of using activation function Tanh is the most optimal choice compared to Sigmoid and ReLU.

This is due to the characteristics of Tanh which can handle data centred around zero better, as well as being able to overcome the gradient disappearances problem that often occurs in Sigmoid activation. ReLU activation, although simple, can cause dead neuron problems which are neurons that fail to activate in the learning process because the activation is outside the usable range.

Thus, for modelling that requires high accuracy and is ready to sacrifice computation time, LSTM can be a suitable choice. However, if computational time efficiency is a top priority, GRU can be a better alternative with almost comparable performance to LSTM.

4. Conclusions

LSTM is the most effective algorithm in predicting the closing price of Bitcoin which is measured by the RMSE value of 0.012441 and MSE 0.000155, but it is not very efficient because of the high computational level so it took 3 minutes 24 seconds to make a prediction, followed by GRU which gets RMSE results of 0.12568 and MSE 0.000158 with a computation time of 1 minute 25 seconds, then the RNN is the fastest algorithm with a computation time of 1 minute 23 seconds, but it is not effective because it has the largest error value compared to the GRU and LSTM algorithms with RMSE values of 0.013445 and MSE 0.000181. Hence, the use of Tanh activation is the most optimal activation in predicting Bitcoin prices, compared to other activations. However, currently, the conducted research is still limited to measuring and comparing the performance of RNN models based on accuracy (RMSE and MSE values) and computing time only. Therefore,

further exploration involving other performance measures, such as Mean Absolute Error (MAE), Prediction Interval Coverage Probability (PICP) and Prediction Interval Width (PIW) to obtain a more comprehensive understanding of the performance of RNN models in predicting time-series data, especially Bitcoin price movements should be carried on. Apart from that, testing of other deep learning models, such as BI-LSTM or CNN-LSTM, which are accompanied by larger datasets, also needs to be reviewed to obtain more complete performance characteristics of deep learning models in predicting time-series data. Finally, integrating several RNN models to produce an ensemble also provides space for further research in the future.

Acknowledgements

The research team would like to thank the Faculty of Information Technology, Perbanas Institute for the support provided during the research and preparation of this article.

References

[1] A. Pranolo, Y. Mao, Y. Tang, Haviluddin, and A. P. Wibawa, "A Long Short Term Memory Implemented for Rainfall Forecasting," *2020 6th Int. Conf. Sci. Inf. Technol. Embrac. Ind. 4.0 Towar. Innov. Disaster Manag. ICSITech 2020*, pp. 194–197, 2020, doi: 10.1109/ICSITech49800.2020.9392056.

[2] Z. Shen, Q. Wan, and D. J. Leatham, "Bitcoin Return Volatility Forecasting: A Comparative Study between GARCH and RNN," *J. Risk Financ. Manag.*, vol. 14, no. 7, 2021, doi: 10.3390/jrfm14070337.

[3] M. Liu, W. Luo, Z. Cai, X. Du, J. Zhang, and S. Li, "Numerical-discrete-scheme-incorporated recurrent neural network for tasks in natural language processing," *CAAI Trans. Intell. Technol.*, vol. 8, no. 4, pp. 1415–1424, 2023, doi: 10.1049/cit2.12172.

[4] Y. Liu, Y. Wang, and H. Shi, "A Convolutional Recurrent Neural-Network-Based Machine Learning for Scene Text Recognition Application," *Symmetry (Basel).*, vol. 15, no. 4, 2023, doi: 10.3390/sym15040849.

[5] A. ur Rehman, S. B. Belhaouari, M. A. Kabir, and A. Khan, "On the Use of Deep Learning for Video Classification," *Appl. Sci.*, vol. 13, no. 3, p. 2007, Feb. 2023, doi: 10.3390/app13032007.

[6] P. Rani, J. Kaur, and S. Kaswan, "Automatic Video Classification: A Review," *EAI Endorsed Trans. Creat. Technol.*, vol. 7, no. 24, p. 163996, 2020, doi: 10.4108/eai.13-7-2018.163996.

[7] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool, "Transformers in Time-Series Analysis: A Tutorial," *Circuits, Syst. Signal Process.*, vol. 42, no. 12, pp. 7433–7466, Dec. 2023, doi: 10.1007/s00034-023-02454-8.

[8] C. Zhao, P. Hu, X. Liu, X. Lan, and H. Zhang, "Stock Market Analysis Using Time Series Relational Models for Stock Price Prediction," *Mathematics*, vol. 11, no. 5, 2023, doi: 10.3390/math11051130.

[9] H. Abdollahi, "A novel hybrid model for forecasting crude oil price based on time series decomposition," *Appl. Energy*, vol. 267, no. February, p. 115035, 2020, doi: 10.1016/j.apenergy.2020.115035.

[10] I. Salehin, I. M. Talha, M. Mehedi Hasan, S. T. Dip, M. Saifuzzaman, and N. N. Moon, "An Artificial Intelligence Based Rainfall Prediction Using LSTM and Neural Network," *Proc. 2020 IEEE Int. Women Eng. Conf. Electr. Comput. Eng. WIECON-ECE 2020*, pp. 5–8, 2020, doi: 10.1109/WIECON-ECE52138.2020.9398022.

[11] Y. Li, "Forecasting Chinese carbon emissions based on a novel time series prediction method," *Energy Sci. Eng.*, vol. 8, no. 7, pp. 2274–2285, 2020, doi: 10.1002/ese3.662.

[12] J. Zheng and M. Huang, "Traffic flow forecast through time series analysis based on deep learning," *IEEE Access*, vol. 8, pp. 82562–82570, 2020, doi: 10.1109/ACCESS.2020.2990738.

[13] A. F. M. Shahen Shah *et al.*, "On the Vital Aspects and Characteristics of Cryptocurrency—A Survey," *IEEE Access*, vol. 11, no. February, pp. 9451–9468, 2023, doi: 10.1109/ACCESS.2023.3240103.

[14] I. E. Livieris, N. Kiriakidou, S. Stavroyiannis, and P. Pintelas, "An advanced CNN-LSTM model for cryptocurrency forecasting," *Electron.*, vol. 10, no. 3, pp. 1–16, 2021, doi: 10.3390/electronics10030287.

[15] Y. Li, Z. Zheng, and H.-N. Dai, "Enhancing Bitcoin Price Fluctuation Prediction Using Attentive LSTM and Embedding Network," *Appl. Sci.*, vol. 10, no. 14, p. 4872, Jul. 2020, doi: 10.3390/app10144872.

[16] M. Musfiroh, D. C. R. Novitasari, P. K. Intan, and G. G. Wisnawa, "Penerapan Metode Principal Component Analysis (PCA) dan Long Short-Term Memory (LSTM) dalam Memprediksi Prediksi Curah Hujan Harian," *Build. Informatics, Technol. Sci.*, vol. 5, no. 1, pp. 1–11, 2023, doi: 10.47065/bits.v5i1.3114.

[17] D. Z. Haq *et al.*, "Long Short-Term Memory Algorithm for Rainfall Prediction Based on El-Nino and IOD Data," *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 829–837, 2021, doi: 10.1016/j.procs.2021.01.071.

[18] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, "Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach," *Fractal Fract.*, vol. 7, no. 2, pp. 1–18, 2023, doi: 10.3390/fractalfract7020203.

[19] R. Sari, K. Kusrini, T. Hidayat, and T. Orphanoudakis, "Improved LSTM Method of Predicting Cryptocurrency Price Using Short-Term Data," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 17, no. 1, p. 33, 2023, doi: 10.22146/ijccs.80776.

[20] S. Hansun, A. Wicaksana, and A. Q. M. Khaliq, "Multivariate cryptocurrency prediction: comparative analysis of three recurrent neural networks approaches," *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00601-7.

[21] M. J. Hamayel and A. Y. Owda, "A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms," *AI*, vol. 2, no. 4, pp. 477–496, 2021, doi: 10.3390/ai2040030.

[22] M. L. Pratama and H. Utama, "Pendekatan Deep Learning Menggunakan Metode LSTM Untuk Prediksi Harga Bitcoin," vol. 2, no. 2, pp. 43–50, 2023.

[23] A. Dutta, S. Kumar, and M. Basu, "A Gated Recurrent Unit Approach to Bitcoin Price Prediction," *J. Risk Financ. Manag.*, vol. 13, no. 2, 2020, doi: 10.3390/jrfm13020023.

[24] Moch Farryz Rizkilloh and Sri Widiyanesti, "Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM)," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 1, pp. 25–31, 2022, doi: 10.29207/resti.v6i1.3630.

[25] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *Eur. Phys. J. Spec. Top.*, vol. 228, no. 10, pp. 2313–2324, 2019, doi: 10.1140/epjst/e2019-900046-x.

[26] Merinda Lestandy, Abdurrahim Abdurrahim, and Lailis Syafa'ah, "Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naïve Bayes," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 802–808, 2021, doi: 10.29207/resti.v5i4.3308.

[27] F. Zhou, Y. Chen, and J. Liu, "Application of a New Hybrid Deep Learning Model That Considers Temporal and Feature Dependencies in Rainfall–Runoff Simulation," *Remote Sens.*, vol. 15, no. 5, 2023, doi: 10.3390/rs15051395.

[28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[29] H. Widiputra, Adele Mailangkay, and Elliana Gautama, "Prediksi Indeks BEI dengan Ensemble Convolutional Neural Network dan Long Short-Term Memory," *J. RESTI (Rekayasa*

Sist. dan Teknol. Informasi), vol. 5, no. 3, pp. 456–465, 2021, doi: 10.29207/resti.v5i3.3111.

[30] F. Azizi and W. C. Wibowo, "Intermittent Demand Forecasting Using LSTM With Single and Multiple Aggregation," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 5, pp. 855–859, 2022, doi: 10.29207/resti.v6i5.4435.

[31] H. Widiputra, A. Mailangkay, and E. Gautama, "Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/9903518.

[32] Sabar Sautomo and Hilman Ferdinandus Pardede, "Prediksi Belanja Pemerintah Indonesia Menggunakan Long Short-Term Memory (LSTM)," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 99–106, 2021, doi: 10.29207/resti.v5i1.2815.

[33] T. Sap *et al.*, "Long-Short Term Memory Technique for Monthly Rainfall," *Symmetry (Basel).*, vol. 14, no. 1599, pp. 1–24, 2022.

[34] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555

[35] Cornelius Stephanus Alfredo and D. A. Adytia, "Time Series Forecasting of Significant Wave Height using GRU, CNN-GRU, and LSTM," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 5, pp. 776–781, 2022, doi: 10.29207/resti.v6i5.4160.

[36] X. Wang, J. Xu, W. Shi, and J. Liu, "OGRU: An Optimized Gated Recurrent Unit Neural Network," *J. Phys. Conf. Ser.*, vol. 1325, no. 1, 2019, doi: 10.1088/1742-6596/1325/1/012089.

