Comparing Bitcoin's Prediction Model Using GRU, RNN, and LSTM by Hyperparameter Optimization Grid Search and Random Search

Nurhayati Buslim
Informatics Engineering
Universitas Negeri Syarif Hidayatullah Jakarta
Jakarta, Indonesia
nurhayati@uinjkt.ac.id

Imam Lutfi Rahmatullah
Informatics Engineering
Universitas Negeri Syarif Hidayatullah Jakarta
Jakarta, Indonesia
imam.lutfi18@mhs.uinjkt.ac.id

Bayu Aji Setyawan
Informatics Engineering
Universitas Negeri Syarif Hidayatullah Jakarta
Jakarta, Indonesia
bayu.ajisetyawan18@mhs.uinjkt.ac.id

Aryajaya Alamsyah
Informatics Engineering
Universitas Negeri Syarif Hidayatullah Jakarta
Jakarta, Indonesia
aryajaya.alamsyah14@mhs.uinjkt.ac.id

2021 9th International Conference on Cyber and IT Service Management (CITSM) | 978-1-6654-3548-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/CITSM52892.2021.9588947

***Abstract*— Being the most expensive and most popular cryptocurrency, both the business world and the research community have started to study bitcoin development. However, due to the absence of most government regulation, the price of bitcoin has become uncontrollable, resulting in frequent large fluctuations. Using a dataset from 17 August 2017 to 13 April 2021 the GRU, RNN, and LSTM methods will be compared and implement Grid Search and Random Search to find out which one will do better in this research. Those three methods are considered to be the best method to get a prediction, but it also depends on the model that computer could have. The best result is the GRU with Grid Search method with MAE (Mean Absolute Error) of training 0.0043 and testing about 0.0594.**

Keywords—GRU, RNN, LSTM, Random Search, Grid Search

I. Introduction

In recent years, digital currencies have developed so rapidly because the public's need for transactions has increased rapidly. This has made various kinds of innovations appear in online transaction methods. One method that is currently popular in the digital world is transacting using crypto currency, or Cryptocurrency. This currency has many kinds with various advantages offered. Bitcoin is one of the most popular types of Cryptocurrency today. Bitcoin is the first cryptocurrency in the world which was introduced by Sathoshi Nakamoto in 2008.

Bitcoin is the most infamous cryptocurrency. On the paper called "Bitcoin: Peer-to-Peer Electronic Money System." by Satoshi Nakamoto, bitcoin first introduced. Satoshi Nakamoto proposed a currency based on the Peer-to-Peer electronic system.

Cryptocurrency has several advantages and disadvantages if you want to use it as a currency, for now there is no clear and definite law to regulate the circulation of digital currency (cryptocurrency) such as bitcoin for example. If there is an abuse of money in digital form such as fraud, money laundering, or other criminal acts, the institution that will be responsible will not exist. In addition, money that can be used as a means of payment must meet the requirements as a means of payment and be recognized by the government. For now, cryptocurrency digital money does not meet the requirements; there is no recognition from the government as a means of payment, because Bitcoin is a new phenomenon by some people in Indonesia.

The main reason we create this research is to point out what is the best method to forecast and technical analysis bitcoin price which later will be helping traders to determine decision making in bitcoin transactions. Technical analysis is a method that focused on price and volume in the market [1]. On the other hand, forecasting is one of many tools for predicting data effectively and efficiently. By method, forecasting can be divided to subjective and objective [2]. On the subjective method there is a qualitative model, the other method has casual and time series model., namely the time series and causal models. The qualitative model seeks to include subjective factors in the forecasting model, this model will be very useful if accurate quantitative data is difficult to obtain.

The data collected throughout time in sequence to know quarter, month, and week and in some cases days or hours is called Time Series. We can estimate value for the future (forecast) by analyzing Time series data, four components will be seen that will affect past and present data patterns that tend to repeat themselves in the future [3]. Dataset later that we used contains historical data (bitcoin prices history) which is suitable for time series method algorithms such as RNN, GRU, and LSTM.

In a neural network, models can be tuned by adding some Hyperparameters to enhance the model quality. Hyperparameter optimization can be deciphered as an optimization issue where the objective is to discover a value that maximizes the execution and yields a wanted show [4]. Here we are inspired by this journal [5] where this is to use a comparative study to find the best algorithm in their problem. Hence, we use hyperparameter optimization Grid Search and Random Search which is commonly used to find which one would fit better to the model given with GRU, RNN, or the LSTM.

II. Related Work

Yiying and Yeze [6] their research applies ANN and LSTM as algorithms to predict cryptocurrency, especially bitcoin. The results of his research indicate that the two methods show fairly good results in predicting. However, in the memory analysis of the model made, Long Short-Term Memory tends to depend more on the short term meanwhile

Artificial Neural Network depend more on the long-term history.

Phaladisailoed and Numnonda [7] using the RNN, Huber Regression, Theil-Sen Regression and LSTM methods to compare which model is the best for predicting bitcoin prices. There are many predictive features such as Open, Close, High, and Low; the prediction results show that all GRU methods are better than regression. However, the model that is made also depends on the parameters made, because it greatly affects the prediction results.

Shewalkar et al [8] Implementing GRU, LSTM, and RNN in Speech Recognition applications. This is done due to the development of a feedforward neural network that is no longer able to handle speech data properly. This study compares the performance of GRU, LSTM, and RNN using the reduced TED-LIUM speech dataset. These results indicate that the GRU and LSTM methods have similar results, but slightly higher for LSTMs. However, the training time for LSTMs is longer with high accuracy. Therefore, this journal concludes that using GRU is better for speech recognition using the reduced TED-LIUM speech dataset.

III. Problem Formulation

The research method uses a machine learning approach using TensorFlow Keras and directly chooses the variable "Close" to affect forecasting results. Then enter the general stages in the data forecasting process such as data collection, data preprocessing, splitting data into training data and test data, creating data models, and predicting models.

A. Data Acquisition

The data we use is time series data obtained from various websites of historical data providers with Crypto Exchange (providers of cryptocurrency buying or selling), namely from https://www.cryptodatadownload.com. The dataset was downloaded with xlxs format which have some features like UNIX, date, symbol, open, high, low, close, volume btc, volume usd, and tradecount. It consists of 1356 rows of which the row is based on the number of days.

unix	date	symbol	open
1619913600000	2021-05-02 00:00:00	BTC/USDT	57797.35
1619827200000	2021-05-01 00:00:00	BTC/USDT	57697.25
1619740800000	2021-04-30 00:00:00	BTC/USDT	53555.00
1619654400000	2021-04-29 00:00:00	BTC/USDT	54846.23
1619568000000	2021-04-28 00:00:00	BTC/USDT	55011.97

high	low	close	Volume BTC	Volume USDT	tradecount
57911.02	57709.95	57861.01	185.467370	1.071830e+07	6824.0
58458.07	56956.14	57800.37	42600.351836	2.460343e+09	1743013.0
57963.00	53013.01	57694.27	68578.910045	3.808187e+09	2267648.0
55195.84	52330.94	53555.00	52486.019455	2.823220e+09	1763676.0
56428.00	53813.16	54846.22	55130.459015	3.020243e+09	1830042.0

Fig. 1. Dataset features from www.cryptodatadownload.com

The data that was downloaded came from Exchange Gemini with a span of 17 August 2017 to 13 April 2021. In other words, we took the dataset from the original data in the field.

Because the application of forecasting uses time series data, the authors use all existing datasets without trimming them first, because the results of the Prediction model rely on the historical data from the past.

The data that we acquire can be said to be non-defective or dirty. The value of the time span and the completeness of the data is complete. There are no null values or extreme outliers in each of the data indexes.

B. Data Preprocessing

Raw information ordinarily comes with numerous flaws such as irregularities, lost values, commotion and/or redundancies. Execution of ensuing learning calculations will in this way be undermined in the event that they are displayed with low-quality data. In this way by conducting appropriate preprocessing steps, we are ready to essentially impact the quality and unwavering quality of consequent programmed disclosures and choices.

Preprocessing data is pointed at changing crude input into high-quality one that appropriately fits the mining prepares to take after. Preparation is considered as an obligatory step and it incorporates strategies such as integration, normalization, cleaning and transformation [9]. Because the dataset that the author got can be categorized as quite good and there is no dirty data or defective data, then for the preprocessing data we only make the data sorted by date in ascending manner. Because the data received shows its stack shape, every new data is placed at the top (row 1). If left unchecked, the time series plotting will start from the most recent data to the first data input. Therefore, the data is changed in ascending order.

C. Splitting Data

Data sharing is done by dividing the time span into 2 parts. First, for training data, the time span taken is the first 1000 data. Then the test data was taken, the remaining about 356 data.

D. GRU

Gated Recurrent Unit (GRU) is a one step forward improvisation demonstration on Long Short-Term Memory (LSTM). The gating system signaling that regulates how the display input and previous memory are used to alter the present actuation and construct the current state is largely responsible for GRU's triumph [10]. GRU cells differ from LSTM cells in that they combine the Input-Gate and Forget-Gate to form the Update-Gate, and they have two doors, Update-Gate and Reset-Gate.

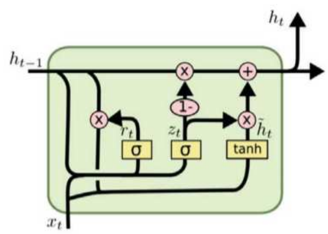

Fig. 2. Gated Recurrent Unit cell

The following equation can be used to represent the connection between input and output:

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \quad (1)$$
$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \quad (2)$$
$$\tilde{h}_t = tanh(W.[r_t \times h_{t-1}, x_t]) \quad (3)$$
$$h_t = (1 - z_t) \times h_{t-1} + z_1 \times \tilde{h}_t \quad (4)$$

where t, z, and t_r are the output of Update-Gate and Reset Gate, W_r and W_z are the weights of the Reset-Gate and Update-Gate; σ (.) and tanh (.) are Sigmoid and Hyperbolic Tangent functions. Reset-Gate helps capture the short-term dependencies on sequence data and Update-Gate helps capture long-term dependencies correspondingly. Where t, z, and t_r are the Update-Gate and Reset Gate outputs, W_r and W_z are the Reset-Gate and Update-Gate weights, σ (.) and tanh (.) are the Sigmoid and Hyperbolic Tangent functions, respectively. Reset-Gate aids in the capture of short-term sequence data dependencies, whereas Update-Gate aids in the capture of long-term dependencies

E. RNN

In a traditional neural network, we accept that all inputs (and yields) are autonomous of each other. In any case, the arrangement and significance of the arrangement need to be considered in numerous cases. RNN (Recurrent Neural Network) are called recurrent since they perform the same assignment for each component of a sequence, with the yield being dependent on the past computations [11]. We will moreover consider RNNs to have a memory which captures data almost what has been calculated.

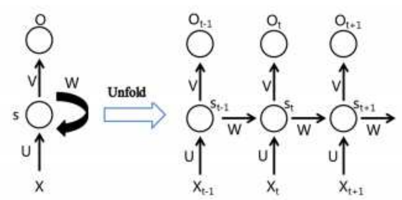

Fig. 3. Illustration of Recurrent Neural Network

Picture over appears a RNN being unrolled (or unfolded) into a full network. By unrolling we essentially mean that we compose out the organization for the total grouping.

F. LSTM

Unlike traditional RNNs, an LSTM (Long Short Term Memory) network is well-suited to memorize from encounter to classify, handle and predict time series when there are exceptionally long time slacks of obscure estimate between critical occasions [12]. The LSTM moreover can handle the issue of vanishing slope and detonating angle which cannot be dodge by RNN amid back propagation optimization.

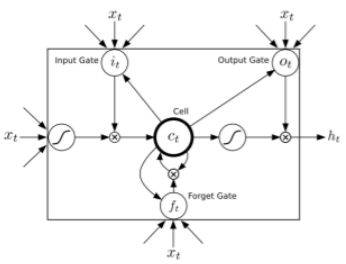

Fig. 4. Long Short-Term Memory Cell

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (5)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (6)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (7)$$
$$g_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (8)$$
$$c_t = F_t \times c_{t-1} + i_t \times g_t \quad (9)$$
$$h_t = o_t \times tanh\,(c_t) \quad (10)$$

The input gate, forget gate, output gate, and cell and cell input activation vectors are represented by i, f, o, and c, respectively. is the logistic sigmoid function, which is defined as (7). The tanh function is represented as (8). The hidden layer outputs ht and ht-1 are the first and last hidden layer outputs, respectively. The linked weight matrix between two units is represented by W_{xi}, W_{hi}, W_{xo}, W_{ho}, W_{xf}, W_{hf}, W_{xc} and W_{hc}. Bias words include b_i, b_o, b_f, b_c.

G. MAE

Mean Absolute error measure the distance between true value and predicted value, but after we measure the distance, it'll be converted to absolute value so the distance will be always positive.

$$mae = \frac{\sum_{i=1}^{n} abs(y_i - \lambda(x_i))}{n} \quad (11)$$

the y_i is the true value and $\lambda(x_i)$ is for predicted value where the n is number of test instances [13].

IV. PROBLEM SOLUTION

Now, to make a better prediction model, we will apply Hyperparameter Optimization to our model. As previously stated, we will focus on two Hyperparameter Optimization, namely Grid Search and Random Search. Each method that we tested was applied one by one Hyperparameter Optimization. In other words, for the GRU method we will apply the Grid Search, Random Search, and without Hyperparameter Optimization respectively. That way, we get 9 comparisons that we will get later. The first, we must know what Grid Search and Random Search are.

A. Grid Search

Grid search is a technique that exhaustively examines a physically designated portion of the hyperparameter space of the focused-on algorithm [14]. Grid search may be a tuning method that endeavors to compute the ideal values of hyperparameters. It is a comprehensive search that's performed on the particular parameter values of a model. The model is additionally known as an estimator.

B. Random Search

Random search is an algorithm where it choose a value for each parameter autonomously employing some sort of probability distribution [14]. Random search is incredible for disclosure and getting hyperparameter combinations that you just would not have speculated naturally, in spite of the fact that it regularly requires more time to execute. The characterizes a search space as a bounded domain of hyperparameter values and arbitrarily sample points in that domain.

V. RESULT AND SIMULATION

This section will experimentally evaluate the performance of two categories which are:

1. Using Hyperparameter Optimization Grid Search
2. Using Hyperparameter Optimization Random Search

First before creating the model and feeding the data to the neural network, the first thing to do is normalize the dataset. To normalize it we can use this formula:

$$x' = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{12}$$

Beside normalization, we also split the dataset to 1000 (74%) training and 356 (26%) as test set. It aims to see how good the model is with data unseen before. After that we used the common parameter to map the dataset by 40 windows and 50 batches. And the last one we did feature selection to choose the "Close" feature of the bitcoin dataset. We only used "Close" because it's the standard feature to account in trading cryptocurrencies. "Close" is the last price of the crypto currency on that day, so to predict for tomorrow "Close" is the key for trading.

In this experiment, every model to train is using the same configurations like down below:

1. Reshape Layer
2. RNN/GRU/LSTM Layer
3. RNN/GRU/LSTM Layer
4. RNN/GRU/LSTM Layer
5. Dense Layer

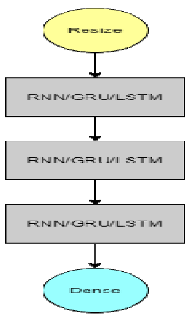

Fig. 5. Illustration of model's architecture that being used in this paper

These are the Hyperparameters will be used for optimization:

- First hidden layer: [16, 32, 64, 128]
- Second hidden layer: [16, 32, 64, 128]
- Third hidden layer: [16, 32, 64, 128]
- Activation Function on Dense Layer: Sigmoid, Linear, ReLu
- Optimizer model: ReLu, Adam, Sigmoid

```
 model =
tf.keras.wrappers.scikit_learn.KerasRegressor(bui
ld_fn=create_model, epochs=150, batch_size =
batch_size, verbose=0)
lstm_unit_1 = [16, 32, 64, 128]
lstm_unit_2 = lstm_unit_1.copy()
lstm_unit_3 = lstm_unit_1.copy()
optimizer = ['SGD', 'Adam', 'Adamax', 'RMSprop']
activation_dense = ['relu', 'linear', 'sigmoid',]
param_grid = dict(lstm_unit_1=lstm_unit_1,
                  lstm_unit_2=lstm_unit_2,
                  lstm_unit_3=lstm_unit_3,
                  optimizer=optimizer,

activation_dense=activation_dense
                  )

grid = GridSearchCV(estimator=model,
param_grid=param_grid, n_jobs=1,
scoring='neg_mean_absolute_error')
grid_result = grid.fit(x_train_model,
y_train_model)
```

Fig. 6. Code Snippet 1. Grid Search Method

```
 model =
tf.keras.wrappers.scikit_learn.KerasRegressor
(build_fn=create_model, epochs=150,
batch_size = batch_size, verbose=0)
lstm_unit_1 = [16, 32, 64, 128]
lstm_unit_2 = lstm_unit_1.copy()
lstm_unit_3 = lstm_unit_1.copy()
optimizer = ['SGD', 'Adam', 'Adamax',
'RMSprop']
activation_dense = ['relu', 'linear',
'sigmoid',]
param_grid = dict(lstm_unit_1=lstm_unit_1,
                  lstm_unit_2=lstm_unit_2,
                  lstm_unit_3=lstm_unit_3,
                  optimizer=optimizer,

activation_dense=activation_dense
                  )

random = RandomizedSearchCV(estimator=model,
param_grid=param_grid, n_jobs=1,
scoring='neg_mean_absolute_error')
random_result= grid.fit(x_train_model,
y_train_model)
```

Fig. 7. Code Snippet 2. Random Search Method

Those are code snippets for creating grid search and as we can see the parameter and hyper parameter that we used are the same as our disclaimer before. We are using negative mean absolute error because this is a time series problem to put the highest MAE in the bottom of the rank method.

All experiments in this research were conducted on a Personal Computer running TensorFlow GPU (CUDA) equipped with Intel Core i9-10900KF 3,7 GHz, 16 GB of memory, and NVIDIA GeForce RTX 3080 GPU 10 GB of Memory.

A. Hyperparameter Optimization Grid Search

RNN, GRU, and LSTM models that are trained using the Grid Search take varying amounts of time. For RNN it takes about 2 hours 5 minutes 10 seconds, for GRU it takes about 4 hours 1 minute 53 seconds and finally for LSTM it takes about 3 hours 31 minutes 7 seconds. On 50 epochs and a total Grid Search iteration of 1537 times.

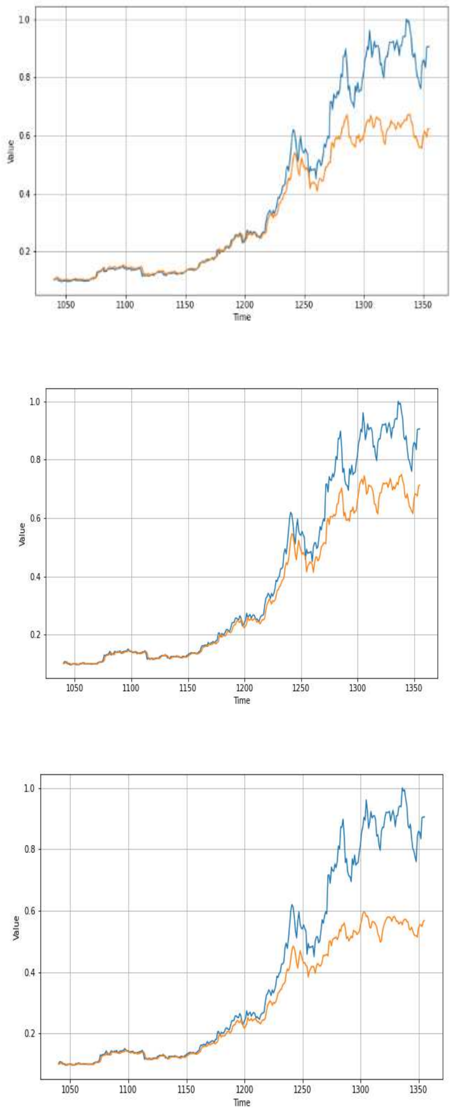

Fig. 8. Testing of RNN, GRU, and LSTM with Grid Search

As we can see from the graphics above, the accuracy is determined by how close is the testing result (orange line) to the actual data (blue line). The testing result of RNN is not bad, but not good. They have slight accuracy when it reaches 'spiky' shape. And the GRU is the best so far, it doesn't match exactly like the real data but it is the closest prediction of the model. And the last one LSTM is surprisingly not good; it is kind of bad because LSTM model can't deal with the actual data. So, we can say that GRU model has more adaptability to the test set than the other 2 models.

B. Hyperparameter Optimization Random Search

Next, the RNN, GRU, and LSTM models that are trained using random search take varying amounts of time. For RNN it takes about 16 minutes 32 seconds, for GRU it takes about 47 minutes 38 seconds and finally for LSTM it takes about 16 minutes 32 seconds. With the number of epochs as many as 150 and Random Search iteration is set 100 times.

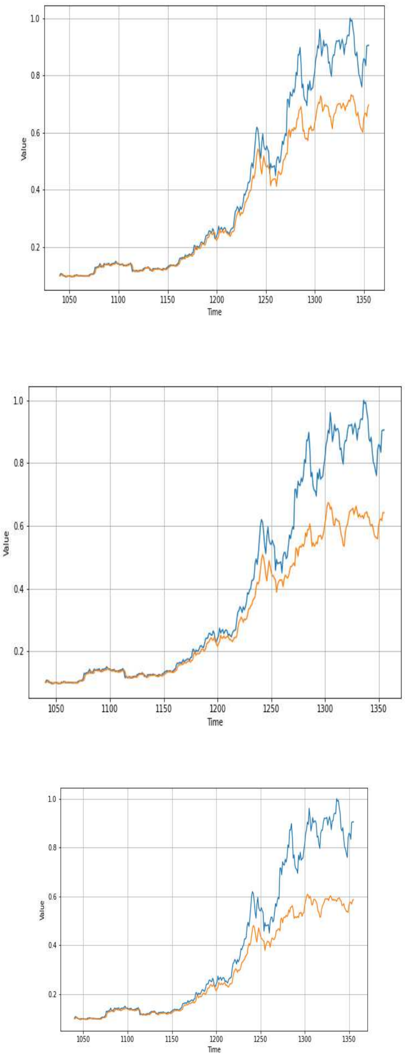

Fig. 9. Testing of RNN, GRU, and LSTM with Random Search

Interestingly, the LSTM model on random search is more accurate than the GRU And that's logical considering that random search is not guaranteed to find the best hyperparameters. Even we can't clearly seen the differences image above, but if we pay attention to the shape it tell us the same pattern as the actual data.

We use MAE (Mean Absolute Error) for metrics because its more common in time series analysis [15] and we put them all results to a table, it look like this:

TABLE I. MAE RESULT ON ALL MODELS

Hyper Param	**Method**	**Train**	**Test**
Grid Search	**RNN**	0.005312362858975959	0.07327416996645006
	GRU	0.004333221323413428	0.05936857883241456
	LSTM	0.004147684353620882	0.09947486805032896
Random Search	**RNN**	0.004252993518323228	0.061953168649061204
	GRU	0.0045455186871690995	0.0839854264687884
	LSTM	0.004265444788501635	0.09666303902669338
Without Hyper Param	**RNN**	0.07493549258039695	0.3936291193184623
	GRU	0.003955552953871398	0.06961754791752854
	LSTM	0.004269957587696327	0.08985289835946308

As we can see from the result above on the test set, the bolded number is the smallest MAE we can get which is from Grid Search on GRU (Gated Recurrent Unit). This might happen because the GRU is the most advanced method developed from the LSTM model and it also will work much better with a small dataset. But if we use Grid Search on these models, it costs much time to train this model. But it is really worth it because of the high accuracy we can get above.

VI. CONCLUSION

Bitcoin is a very valuable commodity on the market today. The exchange rate is high. This makes the fluctuation value of bitcoin difficult to determine. Everyone will have difference experiences an advantages when trading with bitcoin, as example. The methods compared namely GRU, RNN, and LSTM using Hyperparameter Optimization of grid search and random search show the best results, namely by using the GRU method by applying the Grid Search in the test set. For the train data, GRU without Hyperparameter Optimization gets pretty good results even though the test accuracy is not as good as the train, this indicates an overfitting of the model. With the results obtained above, we recommend using GRU with Hyperparameter Optimization Grid Search but not Random Search because the difference in MAE accuracy is consistently higher in models that do not use Hyperparameter Optimization. We can see in the table that there is some anomaly in the model (e.g. higher MAE in test set LSTM with grid search compared to LSTM without Hyperparameter Optimization). This is because there is some variability in deep learning and TensorFlow since their training and optimization of deep learning strongly relies on stochastic procedures and TensorFlow using frameworks cuDNN [16].

REFERENCES

[1] B. Busman, N. Nurhayati, F. Amali, and Z. Muttaqin, "Penerapan Big Data Pada Forex Trading Menggunakan Analisa Statistik Dengan Breakout Strategy," *Pseudocode*, vol. 4, no. 2, pp. 137–143, 2017, doi: 10.33369/pseudocode.4.2.137-143.

[2] H. R. Makridakis S, Wheelwright SC, "1 / the Forecasting Perspective," *Forecast. methods Appl.*, pp. 1–632, 1997.

[3] S. M. Idrees, M. A. Alam, and P. Agarwal, "A Prediction Approach for Stock Market Volatility Based on Time Series Data," *IEEE Access*, vol. 7, no. c, pp. 17287–17298, 2019, doi: 10.1109/ACCESS.2019.2895252.

[4] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Long short term memory hyperparameter optimization for a neural network based emotion recognition framework," *IEEE Access*, vol. 6, no. c, pp. 49325–49338, 2018, doi: 10.1109/ACCESS.2018.2868361.

[5] Nurhayati, Busman, and V. Amrizal, "Big Data Analysis Using Hadoop Framework and Machine Learning as Decision Support System (DSS) (Case Study: Knowledge of Islam Mindset)," *2018 6th Int. Conf. Cyber IT Serv. Manag. CITSM 2018*, no. Citsm, pp. 1–6, 2019, doi: 10.1109/CITSM.2018.8674354.

[6] W. Yiying and Z. Yeze, "Cryptocurrency Price Analysis with Artificial Intelligence," *5th Int. Conf. Inf. Manag. ICIM 2019*, pp. 97–101, 2019, doi: 10.1109/INFOMAN.2019.8714700.

[7] T. Phaladisailoed and T. Numnonda, "Machine learning models comparison for bitcoin price prediction," *Proc. 2018 10th Int. Conf. Inf. Technol. Electr. Eng. Smart Technol. Better Soc. ICITEE 2018*, pp. 506–511, 2018, doi: 10.1109/ICITEED.2018.8534911.

[8] A. Shewalkar, D. nyavanandi, and S. A. Ludwig, "Performance Evaluation of Deep neural networks Applied to Speech Recognition: Rnn, LSTM and GRU," *J. Artif. Intell. Soft Comput. Res.*, vol. 9, no. 4, pp. 235–245, 2019, doi: 10.2478/jaiscr-2019-0006.

[9] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017, doi: 10.1016/j.neucom.2017.01.078.

[10] R. Li, J. Hu, and S. Yang, "Deep Gated Recurrent Unit Convolution Network for Radio Signal Recognition," *Int. Conf. Commun. Technol. Proceedings, ICCT*, pp. 159–163, 2019, doi: 10.1109/ICCT46805.2019.8947225.

[11] X. Zhang and T. Luo, "A RNN decoder for channel decoding under correlated noise," *2019 IEEE/CIC Int. Conf. Commun. Work. China, ICCC Work. 2019*, pp. 30–35, 2019, doi: 10.1109/ICCChinaW.2019.8849949.

[12] J. Li and Y. Shen, "Image describing based on bidirectional LSTM and improved sequence sampling," *2017 IEEE 2nd Int. Conf. Big Data Anal. ICBDA 2017*, pp. 735–739, 2017, doi: 10.1109/ICBDA.2017.8078733.

[13] M. Razzaghi, G. J. McLachan, and K. E. Basford, "Mixture Models," *Technometrics*, vol. 33, no. 3, p. 365, 1991, doi: 10.2307/1268796.

[14] S. Chan and P. Treleaven, *Continuous Model Selection for Large-Scale Recommender Systems*, vol. 33. Elsevier Inc., 2015.

[15] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecast.*, vol. 22, no. 4, pp. 679–688, 2006, doi: 10.1016/j.ijforecast.2006.03.001.

[16] R. D. Peng, *Reproducible Research in Pattern Recognition*, vol. 10214, no. December. 2017.