Weighted Deep Learning Approach For Better Forecasting

Seng Hansun* and Farica Perdana Putri

Informatics Department, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia

*Corresponding author. E-mail: seng.hansun@lecturer.umn.ac.id

Received: January 29, 2023; Accepted: June 09, 2023

There are different kinds of time series analysis and forecasting techniques can be found in the literature. The prediction of unknown future values based on known historical data is one of the goals to be achieved. Here, another approach by combining well-known Deep Learning methods with Weighted Moving Average method is introduced. The Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) from the Recurrent Neural Networks family are utilized in this study. We also compare the prediction results of the proposed approach, namely weighted LSTM (w-LSTM) and weighted GRU (w-GRU), with the original implementation of LSTM and GRU. Different scenarios using real-world import values dataset are developed in the experimentation phase. It was found that the proposed approach could get lower Root Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error at 1143.242, 999.028, and 0.155 respectively than the original Deep Learning methods.

Keywords: Gated Recurrent Unit; Long Short-Term Memory; Weighted Moving Average; Time series forecasting

http://dx.doi.org/10.6180/jase.202404_27(4).0013

1. Introduction

A regularly time-ordered observations of quantitative properties of an individual or an object, usually taken at successive equidistant points of time is the common definition for time series data [1]. It is one of the most encountered data types by data scientists [2]. In order to understand the underlying structure or characteristics of a time series data, various methods have been developed, ranging from fundamental to technical and statistical to soft computing methods [3–5]. One of the main goals of time series analysis is to forecast future events or data.

Moving Average (MA) is probably one of the most popular technical methods used by people to predict the future values of time series data. It is relatively simple, easy to compute, objectively managed, and has a robust result [6]. There are some derivative methods included in this MA family. The simplest one is known as Simple Moving Average (SMA). It assumes that every data point in the time series data has the same weight regardless of its position. Another version of MA is the Weighted Moving Average (WMA) which gives different weights for each data point in the time series data. Hence, recent data points will have greater effects than the older ones. A more advanced version is the Exponential Moving Average (EMA) method which uses an exponential approach for each data point weight calculation. Other hybrid MA methods also have been introduced by numerous researchers, such as Autoregressive Integrated Moving Average (ARIMA) and Weighted Exponential Moving Average (WEMA) methods [7–9].

Recently, a new sub-domain in Artificial Intelligence, or more specifically in Machine Learning, has emerged. It is known as Deep Learning which takes further development of Neural Networks in Machine Learning by incorporating deeper layer structure. It has been deployed in different fields to solve different problems and got notable achievements, especially in image classification, natural language processing, and reinforcement learning [10]. Particularly for regression tasks, such as predicting future values of time series data, a special Deep Learning method, known as Recurrent Neural Networks (RNN) has been popularly employed [11]. Similar to MA, RNN also has derived several other well-known methods, such as the Long Short-Term Memory (LSTM) and the Gated Recurrent Unit (GRU). Both of those derivative methods were introduced to handle RNN weaknesses, such as the vanishing or exploding gradient problem [12] and the long-term dependency problem [13].

Both LSTM and GRU had been employed in various cases related to time series analysis. Hansun and Suryadibrata [14], for example, implemented LSTM networks in predicting gold prices during the COVID-19 crisis. They proposed a relatively simple networks architecture that could achieve high prediction results comparable with other complex networks architectures. Biswas et al. [15] tried to predict the stock market price by using several algorithms, including LSTM, Extreme Gradient Boosting (XGBoost), Linear Regression, MA, and the Last Value model. They found LSTM had exceeded all other methods in the experimental phase conducted. In another study, Zheng et al. [16] implemented GRU for short-term power load forecasting in a residential community. They compared the results with LSTM and traditional RNN approaches. Based on the numerical simulation, they concluded GRU model has better performance, convergence speed, and robustness. GRU also had been incorporated by Becerra-Rico et al. [17] to predict the airborne particle pollution in the atmosphere. They found that the proposed method could accurately forecast the behavior of Particulate Matter (PM10) as the object of the study. Both LSTM and GRU had been applied in other studies to predict the electricity load [18–20], cryptocurrency prediction [21, 22], and disease progression [23, 24] to name a few.

Here, we introduce a novel approach to time series forecasting using Deep Learning methods, particularly LSTM and GRU, which are combined with Weighted Moving Average (WMA) method. WMA as a smoothing technique could positively contribute to the overall prediction of time series data, especially the one with high volatility. Hence, we prepare a time series dataset by using the import values in Indonesia and several experimentation scenarios. For convenience, we will call the proposed methods as w-LSTM for the combination of WMA and LSTM, while w-GRU represents WMA and GRU combination.

2. Theory and formula

2.1. Long Short-Term Memory

In 1997, Hochreiter and Schmidhuber introduced a special type of Recurrent Neural Networks (RNN) that could address the limitations found in RNN [25]. The method is known as Long Short-Term Memory (LSTM) that has been widely accepted and implemented in various areas, especially where the data is sequential in nature [26].

LSTM differs from traditional RNN where it uses three gates in the block cell. Those gates are called forget gate (f_t), input gate (i_t), and output gate (o_t). Each gate contributes to the information distribution and value in the LSTM networks. There are six equations incorporated in an LST cell calculation as shown in equations 1 to 6 [27].

$$f_t = \sigma\left(W_f h_{t-1} + U_f x_t + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i h_{t-1} + U_i x_t + b_i\right) \tag{2}$$

$$\tilde{C}_t = \tanh\left(W_C h_{t-1} + U_C x_t + b_C\right) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

$$o_t = \sigma\left(W_o h_{t-1} + U_o x_t + b_o\right) \tag{5}$$

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{6}$$

There are two activation functions (AFs) utilized in an LSTM cell, namely the tangent hyperbolic (tanh) and sigmoid (σ) functions. Both of them are the most frequently used nonlinear AFs [28]. $W_f, W_i, W_o, W_C, U_f, U_i, U_o, U_C$ are the respective weights for forget, input, output gates, and the candidate cell. Similarly, b_f, b_i, b_o, b_C are bias values for forget, input, output gates, and the candidate cell. $\tilde{C}_t$ represents the candidate cell state, C_t represents the current cell state, h_t represents the current hidden state value, x_t indicates the new input at the current cell, and $\odot$ is the element-wise multiplication operator.There are two activation functions (AFs) utilized in an LSTM cell, namely the tangent hyperbolic (tanh) and sigmoid (σ) functions. Both of them are the most frequently used nonlinear AFs [28]. $W_f, W_i, W_o, W_C, U_f, U_i, U_o, U_C$ are the respective weights for forget, input, output gates, and the candidate cell. Similarly, b_f, b_i, b_o, b_C are bias values for forget, input, output gates, and the candidate cell. $\tilde{C}_t$ represents the candidate cell state, C_t represents the current cell state, h_t represents the current hidden state value, x_t indicates the new input at the current cell, and $\odot$ is the element-wise multiplication operator.

2.2. Gated Recurrent UnitGated Recurrent Unit

GRU is another type of RNN recently introduced by Cho et al. [29] in 2014. It has a pretty similar procedure to

LSTM but incorporates fewer gates and hence fewer equations [30]. There are only four equations used in GRU cell calculation as shown in equations 7 to 10 [31].

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{7}$$

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{8}$$

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{9}$$

$$h_t = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{10}$$

In those four equations, z_t is the update gate, while r_t is the reset gate. $\tilde{h}_t$ represents the candidate's hidden state, h_t represents the final output of a GRU unit. Similar to LSTM, $W_z, W_r, W_h, U_z, U_r, U_h, b_z, b_r, b_h$ are the networks' weights and bias values and x_t is the new information to the networks.In those four equations, z_t is the update gate, while r_t is the reset gate. $\tilde{h}_t$ represents the candidate's hidden state, h_t represents the final output of a GRU unit. Similar to LSTM, $W_z, W_r, W_h, U_z, U_r, U_h, b_z, b_r, b_h$ are the networks' weights and bias values and x_t is the new information to the network.

2.3. Proposed Weighted Deep Learning Methods

We propose a novel approach by incorporating Weighted Moving Average (WMA) within Deep Learning methods, particularly LSTM and GRU, for time series forecasting. WMA could act as a smoothing technique that positively contributes to the overall forecasting results of time series data with high volatility. WMA can be formulated as shown in equation 11 where n represents the time span used, A_t and w_t are the real data value and linearly weighted value at time t, and k is the relative position of the current data point [32].

$$WMA_t = \frac{\sum_{t=k-n+1}^{k} w_t A_t}{\sum_{t=k-n+1}^{k} w_t} \tag{11}$$

Figure 1 shows the schematic diagram of the proposed methods, namely weighted LSTM (w-LSTM) and weighted GRU (w-GRU). First of all, the time series data will be preprocessed to handle any missing data. WMA is applied to the cleaned data before the data splitting process (80:20 ratio) is done for train and test sets. Both train and test sets are normalized and reshaped, where the feature scaling applied in the test set is based on the feature scaling result on the train set. To handle any missing values from WMA calculation, zero data imputation is applied to the train set before deep learning model development.

There are two deep learning approaches utilized in this study, viz. LSTM and GRU. Both of them use the same basic architecture and are referred to as w-LSTM and w-GRU. We use three layers architecture, one LSTM (GRU) layer

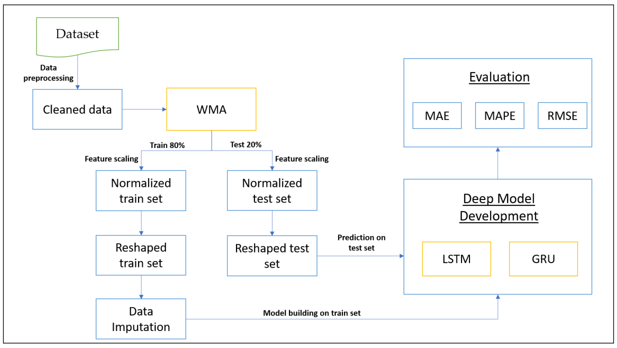

Fig. 1. Schematic diagram of the proposed methods.

with 300 nodes, one Dropout layer to prevent overfitting and drops 10% of processed information, and one Dense layer with one node to represent the final output. The loss function is calculated using Mean Square Error while Adam optimizer is utilized during the training phase which runs for 30 epochs on 32 batch sizes each run. Lastly, the built deep models are used to predict the future values of the test set, which are then evaluated by using three error criteria measurements that will be explained in the following sub-section.

2.4. Error Criteria

In this study, three error criteria are used as the performance metrics, namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Those three criteria are popularly used and can be represented as shown in equation 12 to equation 14 [33].

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(A_t - F_t\right)^2} \tag{12}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|A_t - F_t\right| \tag{13}$$

$$MAPE = \left(\frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|\right) \times 100\% \tag{14}$$

where A_t is the actual data value at time t, F_t is the forecasted value at time t, and n is the total number of compared data.

3. Experimental setup

We prepare a time series data from the official publication of Statistics Indonesia (Badan Pusat Statistik Indonesia) [34]. The data contains Indonesia's import values for both Oil-Gas (OG) products and non-Oil-Gas (nOG) products in millions of US$. We recorded the import values from January 1993 to February 2022. Table 1 summarizes the dataset.

Table 1. Dataset summary.

Characteristics	Oil-Gas	non-Oil-Gas
count	350	350
mean	1515.112000	6679.183429
std	1161.049468	4433.731597
min	108.200	1522.300
25%	431.425	2578.400
50%	1417.000	4494.600
75%	2235.250	10942.525
max	4221.600	17974.200

All deep learning models are trained and tested on Google Colab. Intel ® Xeon ® Processors ®2.20 GHz, 12.69 GB RAM, and 107.72 GB disk spaces are utilized. Python 3 programming language and a number of core libraries, viz. NumPy, Matplotlib, Pandas, Keras, and sklearn are used in the experimental setup. The source code can be accessed on the GitHub repository at https://github.com/senghansun/w-LSTM-w-GRU.

There are two scenarios prepared in the experimental setup. One for the time series data forecasting by using LSTM and GRU methods and another for time series forecasting by using w-LSTM and w-GRU methods. As explained in the previous section, all deep learning methods applied the same architecture, i.e., simple three layers architecture. Moreover, each scenario will be run on two different datasets, one for the import values of Oil-Gas products (OG) and another one for the import values of non-Oil-Gas products (nOG). The prepared scenarios can be seen in Table 2.

4. Result discussions

The forecasting results for each type of time series data (OG and nOG) by using LSTM and GRU deep learning methods are shown in Figure 2 (scenario 1). Similarly, Figure 3 depicts the results for scenario 2 where forecasting results of both OG and nOG by using w-LSTM and w-GRU are shown. The blue color represents actual values, while the red color represents the forecasted values. Both LSTM and GRU could follow the data pattern pretty well in both scenarios despite the relatively small number of data available. Moreover, w-LSTM and w-GRU have successfully smoothed the time series data.

The performance results of applied methods for the experiment conducted is shown in Table 3. Three error criteria are used, namely RMSE, MAE, and MAPE. The lower the score of each criterion implies a smaller prediction error between the actual and forecasted values, and hence, the better the forecasting method. Moreover, the execution time of applied methods for each scenario in the experimental phase is shown in Table 4.

5. Conclusions

We introduce a novel approach for time series forecasting utilizing two popularly used Deep Learning methods, namely LSTM and GRU, combine with WMA method. The proposed methods are referred to as w-LSTM and w-GRU for simplicity. We then prepare a time series dataset containing two types of data and two scenarios to conduct the experiment.

Based on the experimental results, it is confirmed that the proposed methods have smaller error values measured by RMSE, MAE, and MAPE. By using the same deep learning architecture, the original LSTM and GRU applied on both Oil-Gas (OG) and non-Oil-Gas (nOG), got average values of 1550.866, 1322.575, and 0.200 for RMSE, MAE, and MAPE respectively. Meanwhile, the proposed w-LSTM and w-GRU got significantly lower average values of RMSE, MAE, and MAPE at 1143.242, 999.028, and 0.155 respectively. This implies that the introduced approach could give better prediction results.

Moving Average (MA) methods, in this case, WMA, seem to positively contribute to the overall prediction results of a time series data by smoothing the data pattern. This behavior of various MA family methods for financial data smoothing had been extensively discussed in Raudys et al. [35]. Interested readers are encouraged to read the publication.

From the experimental results, both original Deep Learning methods implementation in Scenario 1 and introduced w-LSTM and w-GRU methods in Scenario 2, have relatively the same execution time. Since the data was preprocessed early in the data preparation phase as shown in Fig. 1, there are not many resources and time differences in implementing both original and proposed methods. Although the average time execution for Scenario 1 is slightly faster than Scenario 2, the proposed w-LSTM method is slightly faster for Oil-Gas (OG) and w-GRU is slightly faster for non-Oil-Gas (nOG) than their original methods.

Another finding worth mentioning is the Deep Learning architecture being used in this study. We used a relatively simple Deep Learning architecture for all applied methods (both original and proposed ones) which consist of one LSTM (or GRU) layer, one Dropout layer, and one Dense layer as the output layer. As can be seen in the forecasting results (Figs. 2 and 3), all applied Deep Learning methods have successfully followed the data pattern. Moreover, the prediction results are pretty good despite the relatively small number of data being used in the experiment. Hence, this finding strengthened the argument that Deep Learning

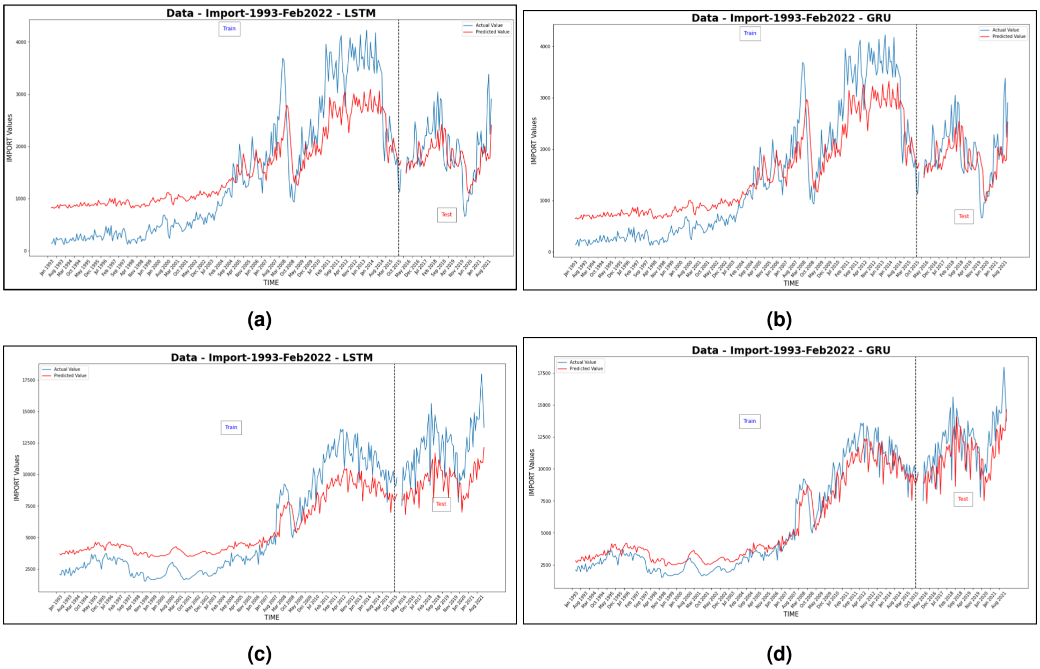

Fig. 2. Forecasting results of Scenario 1: (a) LSTM-OG; (b) GRU-OG; (c) LSTM-nOG; (d) GRU-nOG.

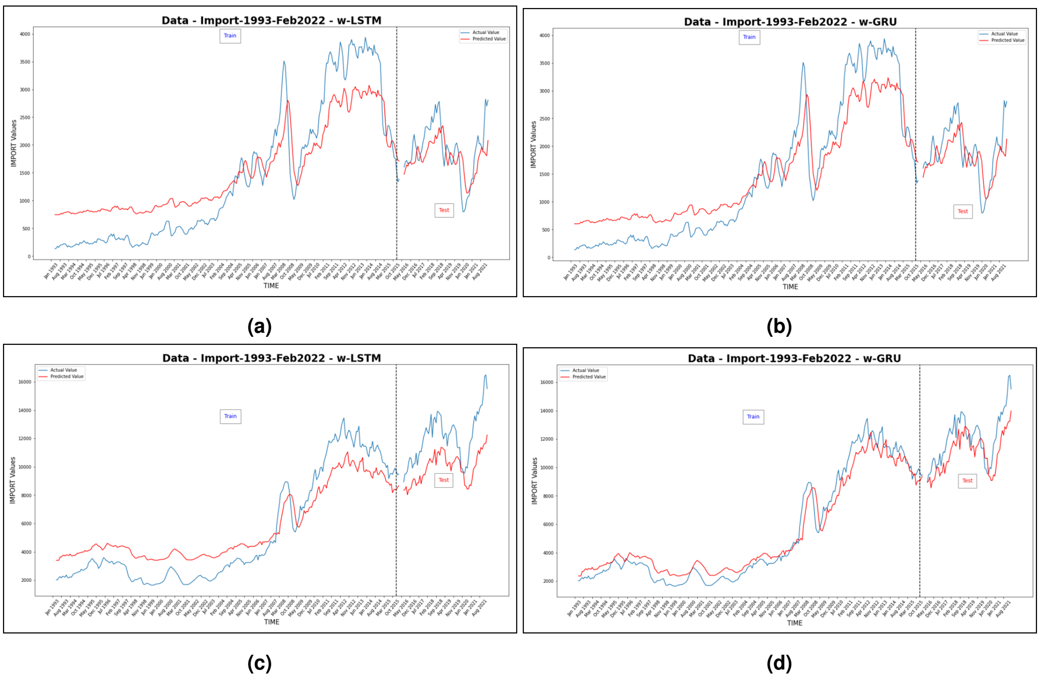

Fig. 3. Forecasting results of Scenario 2: (a) w-LSTM-OG; (b) w-GRU-OG; (c) w-LSTM-nOG; (d) w-GRU-nOG.

Table 2. Prepared scenarios.

Scenario 1		Scenario 2	
LSTM	GRU	w-LSTM	w-GRU
LSTM for Oil and Gas (LSTM-OG)	GRU for Oil and Gas (GRU-OG)	weighted LSTM for Oil and Gas (w-LSTM-OG)	weighted GRU for Oil and Gas (w-GRU-OG)
LSTM for non-Oil and Gas (LSTM-nOG)	GRU for non-Oil and Gas (GRU-nOG)	weighted LSTM for non-Oil and Gas (w-LSTM-nOG)	weighted GRU for non-Oil and Gas (w-GRU-nOG)

Table 3. Performance results measured by RMSE, MAE, and MAPE.

Method	RMSE	MAE	MAPE	Method	RMSE	MAE	MAPE
	Scenario 1				Scenario 2		
LSTM-OG	499.468	383.701	0.220	w-LSTM-OG	389.990	310.053	0.176
GRU-OG	497.662	380.649	0.219	w-GRU-OG	384.310	305.158	0.173
LSTM-nOG	3169.994	2805.236	0.220	w-LSTM-nOG	2380.818	2155.905	0.172
GRU-nOG	2036.338	1720.714	0.141	w-GRU-nOG	1417.849	1224.995	0.098
Average	**1550.866**	**1322.575**	**0.200**	**Average**	**1143.242**	**999.028**	**0.155**

Table 4. Time execution.

Scenario 1	Time (s)	Scenario 2	Time (s)
LSTM-OG	9.043439	LSTM-OG	8.755312
GRU-OG	6.836447	GRU-OG	7.037054
LSTM-nOG	8.323306	LSTM-nOG	12.632214
GRU-nOG	12.732269	GRU-nOG	12.650054
Average	**9.233865**	Average	**10.268659**

methods can effectively learn the hidden pattern in a given dataset, especially the one in the form of a time series data, without the need to add too much complexity [11, 36].

As the last note, incorporating Moving Average within Deep Learning methods could enhance the forecasting result. There are several other MA techniques, despite WMA which was used in this study, that can be incorporated into future research. The hybrid Exponential Smoothing (ES) method [37], Weighted Exponential Moving Average (WEMA) method [7], and even the recently developed Self-Attentive Moving Average (SAMA) method [38] are some prospective methods to be used and compared with this study's results.

Acknowledgements

We would like to acknowledge the support and facilities given by Universitas Multimedia Nusantara during this study.

References

[1] OECD. *Data and Metadata Reporting and Presentation Handbook*. Paris, France: OECD Publishing, 2007, 61.

[2] S. Hansun, V. Charles, C. R. Indrati, and Subanar, (2019) *"Revisiting the Holt-Winters' Additive Method for Better Forecasting"* **International Journal of Enterprise Information Systems** ***15***(2): 43–57. DOI: 10.4018/IJEIS.2019040103.

[3] Y. Uygun, M. O. Erboy, M. S. Aktas, O. Kalipsiz, and I. Aykurt. "Technical Analysis on Financial Time Series Data Based on Map-Reduce Programming Model: A Case Study". In: *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*. Ankara, Turkey: IEEE, 2018, 92–97. DOI: 10.1109/IBIGDELFT.2018.8625357.

[4] V. M. Kureychick and T. G. Kaplunov, (2019) *"Time series forecasting method based on genetic algorithm for predicting the conditions of technical systems"* **Journal of Physics: Conference Series** ***1333***(3): 032046. DOI: 10.1088/1742-6596/1333/3/032046.

[5] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, (2020) *"The M4 Competition: 100,000 time series and 61 forecasting methods"* **International Journal of Forecasting** ***36***(1): 54–74. DOI: 10.1016/j.ijforecast.2019.04.014.

[6] S. Hansun and S. Subanar, (2016) *"H-WEMA: A New Approach of Double Exponential Smoothing Method"* **TELKOMNIKA (Telecommunication Computing Electronics and Control)** ***14***(2): 772. DOI: 10.12928/telkomnika.v14i2.3096.

[7] S. Hansun. "A new approach of moving average method in time series analysis". In: *2013 Conference on New Media Studies (CoNMedia)*. IEEE, 2013, 1–4. DOI: 10.1109/CoNMedia.2013.6708545.

[8] G. Zhang, (2003) *"Time series forecasting using a hybrid ARIMA and neural network model"* **Neurocomputing** ***50***: 159–175. DOI: 10.1016/S0925-2312(01)00702-0.

[9] R. H. Shumway and D. S. Stoffer. "ARIMA Models". In: *Time Series Analysis and Its Applications*. Springer, Cham, 2017, 75–163. DOI: 10.1007/978-3-319-52452-8_3.

[10] B. Lim and S. Zohren, (2021) *"Time-series forecasting with deep learning: a survey"* **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences** ***379***(2194): 20200209. DOI: 10.1098/rsta.2020.0209.

[11] S. Hansun and J. C. Young, (2021) *"Predicting LQ45 Financial Sector Indices using RNN-LSTM"* **Journal of Big Data** ***8***(1): 104. DOI: 10.1186/s40537-021-00495-x.

[12] D. Yi, S. Bu, and I. Kim, (2019) *"An Enhanced Algorithm of RNN Using Trend in Time-Series"* **Symmetry** ***11***(7): 912. DOI: 10.3390/sym11070912.

[13] F. Shahid, A. Zameer, and M. Muneeb, (2020) *"Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM"* **Chaos, Solitons & Fractals** ***140***: 110212. DOI: 10.1016/j.chaos.2020.110212.

[14] S. Hansun and A. Suryadibrata, (2021) *"Gold Price Prediction in COVID-19 Era"* **International Journal of Computational Intelligence in Control** ***13***(2):

[15] M. Biswas, A. Shome, M. A. Islam, A. J. Nova, and S. Ahmed. "Predicting Stock Market Price: A Logical Strategy using Deep Learning". In: *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. Penang, Malaysia: IEEE, 2021, 218–223. DOI: 10.1109/ISCAIE51753.2021.9431817.

[16] J. Zheng, X. Chen, K. Yu, L. Gan, Y. Wang, and K. Wang. "Short-term Power Load Forecasting of Residential Community Based on GRU Neural Network". In: *2018 International Conference on Power System Technology (POWERCON)*. Guangzhou, China: IEEE, 2018, 4862–4868. DOI: 10.1109/POWERCON.2018.8601718.

[17] J. Becerra-Rico, M. A. Aceves-Fernández, K. Esquivel-Escalante, and J. C. Pedraza-Ortega, (2020) *"Airborne particle pollution predictive model using Gated Recurrent Unit (GRU) deep neural networks"* **Earth Science Informatics** ***13***(3): 821–834. DOI: 10.1007/s12145-020-00462-9.

[18] M. Xia, H. Shao, X. Ma, and C. W. de Silva, (2021) *"A Stacked GRU-RNN-Based Approach for Predicting Renewable Energy and Electricity Load for Smart Grid Operation"* **IEEE Transactions on Industrial Informatics** ***17***(10): 7050–7059. DOI: 10.1109/TII.2021.3056867.

[19] G. Memarzadeh and F. Keynia, (2021) *"Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm"* **Electric Power Systems Research** ***192***: 106995. DOI: 10.1016/j.epsr.2020.106995.

[20] Z. Chang, Y. Zhang, and W. Chen, (2019) *"Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform"* **Energy** ***187***: 115804. DOI: 10.1016/j.energy.2019.07.134.

[21] N. Buslim, I. L. Rahmatullah, B. A. Setyawan, and A. Alamsyah. "Comparing Bitcoin's Prediction Model Using GRU, RNN, and LSTM by Hyperparameter Optimization Grid Search and Random Search". In: *2021 9th International Conference on Cyber and IT Service Management (CITSM)*. Bengkulu, Indonesia: IEEE, 2021, 1–6. DOI: 10.1109/CITSM52892.2021.9588947.

[22] F. Ferdiansyah, S. H. Othman, R. Zahilah Raja Md Radzi, D. Stiawan, Y. Sazaki, and U. Ependi. "A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market". In: *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*. Batam, Indonesia: IEEE, 2019, 206–210. DOI: 10.1109/ICECOS47637.2019.8984499.

[23] M. Pavithra, K. Saruladha, and K. Sathyabama. "GRU Based Deep Learning Model for Prognosis Prediction of Disease Progression". In: *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. Erode, India: IEEE, 2019, 840–844. DOI: 10.1109/ICCMC.2019.8819830.

[24] M. Djerioui, Y. Brik, M. Ladjal, and B. Attallah. "Heart Disease prediction using MLP and LSTM models". In: *2020 International Conference on Electrical Engineering (ICEE)*. Istanbul, Turkey: IEEE, 2020, 1–5. DOI: 10.1109/ICEE49691.2020.9249935.

[25] S. Hochreiter and J. Schmidhuber, (1997) *"Long Short-Term Memory"* **Neural Computation** ***9***(8): 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[26] A. Sherstinsky, (2020) *"Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network"* **Physica D: Nonlinear Phenomena** ***404***: 132306. DOI: 10.1016/j.physd.2019.132306.

[27] H. M. Lynn, S. B. Pan, and P. Kim, (2019) *"A Deep Bidirectional GRU Network Model for Biometric Electrocardiogram Classification Based on Recurrent Neural Networks"* **IEEE Access 7**: 145395–145405. DOI: 10.1109/ACCESS.2019.2939947.

[28] S. Li and X. Zhao, (2019) *"Image-Based Concrete Crack Detection Using Convolutional Neural Network and Exhaustive Search Technique"* **Advances in Civil Engineering *2019***: 1–12. DOI: 10.1155/2019/6520620.

[29] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning phrase representations using RNN encoder–decoder for statistical machine translation". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. 2014, 1–15.

[30] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho. "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks". In: *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. Montreal, QC, Canada: IEEE, 2018, 202–206. DOI: 10.1109/NETSOFT.2018.8460090.

[31] Q. Li and Y. Xu, (2019) *"VS-GRU: A Variable Sensitive Gated Recurrent Neural Network for Multivariate Time Series with Massive Missing Values"* **Applied Sciences *9***(15): 3041. DOI: 10.3390/app9153041.

[32] S. Hansun, V. Charles, T. Gherman, and V. Varadarajan, (2022) *"Hull-WEMA: a novel zero-lag approach in the moving average family, with an application to COVID-19"* **International Journal of Management and Decision Making *21***(1): 92–112. DOI: 10.1504/IJMDM.2022.119582.

[33] N. F. Omran, S. F. Abd-el Ghany, H. Saleh, A. A. Ali, A. Gumaei, and M. Al-Rakhami, (2021) *"Applying Deep Learning Methods on Time-Series Data for Forecasting COVID-19 in Egypt, Kuwait, and Saudi Arabia"* **Complexity *2021***: 1–13. DOI: 10.1155/2021/6686745.

[34] BPS - Statistics Indonesia. *Value of Import Oil&Gas - Non Oil&Gas (Million US)*. 2022.

[35] A. Raudys, V. Leniauskas, and E. Malius. "Moving Averages for Financial Data Smoothing". In: *Communications in Computer and Information Science, vol 403*. Ed. by T. Skersys, R. Butleris, and R. Butkiene. Berlin, Heidelberg: Springer, 2013. Chap. Informatio, 34–45. DOI: 10.1007/978-3-642-41947-8_4.

[36] S. Hansun, A. Wicaksana, and A. Q. M. Khaliq, (2022) *"Multivariate cryptocurrency prediction: comparative analysis of three recurrent neural networks approaches"* **Journal of Big Data *9***(50): 1–15. DOI: 10.1186/s40537-022-00601-7.

[37] S. Smyl, (2020) *"A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting"* **International Journal of Forecasting *36***(1): 75–85. DOI: 10.1016/j.ijforecast.2019.03.017.

[38] Y. Su, C. Cui, and H. Qu, (2022) *"Self-Attentive Moving Average for Time Series Prediction"* **Applied Sciences *12***(7): 3602. DOI: 10.3390/app12073602.