International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 1, February 2024, pp. 861~869
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i1.pp861-869

Stock price forecasting in Indonesia stock exchange using deep learning: a comparative study

Agus Tri Haryono, Riyanarto Sarno, Kelly Rossa Sungkono
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received May 31, 2023
Revised Jul 12, 2023
Accepted Jul 17, 2023

Keywords:

Benchmark deep learning
Gated recurrent units
Indonesia stock exchange
Stock price forecasting
Trend evaluation

ABSTRACT

In 2022, the Indonesia stock exchange (IDX) listed 825 companies, making it challenging to identify low-risk companies. Stock price forecasting and price movement prediction are vital issues in financial works. Deep learning has previously been implemented for stock market analysis, with promising results. Because of the differences in architecture and stock issuers in each study report, a consensus on the best stock price forecasting model has yet to be reached. We present a methodology for comparing the performance of convolutional neural networks (CNN), gated recurrent units (GRU), long short-term memory (LSTM), and graph convolutional networks (GCN) layers. The four layers types combination yields 11 architectures with two layers stacked maximum, and the architectures are performance compared in stock price predicting. The dataset consists of open, highest, lowest, closed price, and volume transactions and has 2,588,451 rows from 727 companies in IDX. The best performance architecture was chosen by a vote based on the coefficient of determination (R2), mean squared error (MSE), root mean square error (RMSE), mean absolute percent error (MAPE), and f1-score. TFGRU is the best architecture, producing the finest results on 315 companies with an average score of RMSE is 553.327, MAPE is 0.858, and f1-score is 0.456.

Corresponding Author:

Riyanarto Sarno
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
Email: riyanarto@if.its.ac.id

1. INTRODUCTION

Electing stock issuers for investing short-range had a high-risk failure because the stock market influences by many factors, which created an unstable stock market environment [1]. The task of electing best-fit model forecasting on each stock issuer is urgently needed to reduce the failure effect [2]. The Indonesia stock exchange (IDX) had attractive growth for investors. In 2022, the market share grew to 34.27%, and market capitalization reached 15.06%. Stock issuer listed on IDX is 825 companies. The stock issuer is separated into 11 sectors: raw goods, non-primary consumer goods, primary consumer goods, energy, infrastructure, health, finance, industry, property, technology, and transportation [3]. The best architecture model for each sector is essential to maximizing stock returns on growth IDX.

The IDX is already used in previous research but only uses a few stock issuers. The LQ45 index is used in previous research [4], [5], and [6] because the LQ45 index builds up from 45 companies with excellent company performance and can affect composite stock price index. Weaken of the composite stock price index occurred when the price of the LQ45 index dropped and vice versa. In other research, fewer companies are used to forecast stock prices, especially those with significant market capitalization, such as PT Bank Central Asia Tbk (IDX:BBCA), PT Bank Mandiri (Persero) Tbk (IDX:BMRI), and PT Bank Rakyat Indonesia (Persero) Tbk (IDX:BBRI) [7], [8]. The large market capitalization is chosen because stock price movements tend to climb since the initial public offering (IPO) [9]. The LQ45 only has 45 companies, and the large market capitalization usually only uses the top ten companies, which indicates that the other companies in 825 stock issuers have not been tested yet. We propose using 88% of the stock issuer population to discover that the issuer has the lowest risk of failure.

Stock price forecasting already uses several models, such as statistical [10], pattern recognition [11], machine learning [12]–[14], and deep learning [15]–[18]. The deep learning model using convolutional neural networks (CNN) and recurrent neural networks (RNN) outperformed statistical and machine learning models [15], [18], [19]. In previous research, the combination of the CNN-long short-term memory (LSTM) layer performs better than the standalone CNN or long LSTM layer. The CNN layer successfully extracts features from the input data and passes them to the LSTM layer to predict the output [16]. The LSTM layer surpasses nine machine learning models. The cell of LSTM can recognize the pattern of random time interval data [19]. The GRU layer performance slightly surpasses the LSTM layer [17]. Graph convolutional networks (GCN) perform better than gated recurrent units (GRU) on stock price forecasting. The GCN had a correlation matrix as input data to recognize market signals [20]. Although previous studies have presented the comparative performance of the deep learning model with other models, the consensus of the best architecture is not fulfilled yet because the model architecture is not tested with a large dataset, and the alternative stacked layer model has not been implemented.

In stock price forecasting, no model or parameter can be best-fit for all stock issuers. The model and parameters may be the best option for some issuers but the worst for others. Varying architecture models are required and must be compared in performance before being used in problem-solving. The best architecture was determined by comparison using particular statistical criteria. This study tests the CNN, LSTM, GRU, GCN, and mixed layer models on each stock issuer. The best architecture is determined by voting consisting of five evaluation metrics. Each stock issuer will have a best-fit architecture to reduce stock forecasting failure.

The main contribution of this study is a comprehensive benchmark that compares the performance between CNN, LSTM, GRU, GCN, and mixed layers to fulfill the consensus best architecture on stock price forecasting in IDX. The forecasting models used 727 of 825 stock issuers in the IDX and then measured performance using mean squared error (MSE), mean absolute percentage error (MAPE), coefficient of determination (R^2), root mean square error (RMSE), and f1-score. The impact of this study is not only for the scientific literature but also for investors to reduce risk failure on stock price forecasting in terms of the ability to forecast the IDX. The rest of the article is organized as follows: section 2 presents the research method, section 3 results and discussion, followed by a conclusion in section 4.

2. METHOD

Our comparative deep learning model in stock forecasting uses four layers: CNN, LSTM, GRU, and GCN. Each layer will be stacked with a maximum of two layers. The dataset and experiment setup are presented in this section.

2.1. Dataset

The stock issuer listed on the IDX amount to 825 in 2022. Table 1 shows 727 companies spread across 11 sectors, and 2,588,451 data rows are used in this study. The selected issuer is based on criteria with a minimum of 200 days of the stock transaction to meet the needs of time-lag variations, and the variance on the data test is not zero. The dataset stock transaction consists of open price, highest price, lowest price, close price, and volume of transactions.

2.2. Model forecasting

The combination of CNN, LSTM, GRU, and GCN is used to get the best architecture on stock price forecasting for each issuer. The explanation of each layer is presented in this sub-section. The characteristics of each layer are explained to understand their contributions to the forecasting process.

2.2.1. Convolutional neural networks

Convolutional neural networks (CNN) are feed-forward neural networks that excel in computer vision and text processing. CNN architecture is already used to forecast time series data, especially extracting features from raw input [18]. The CNN architecture uses local perception and weight sharing to enhance the efficiency of the learning model. The fully connected, pooling, and convolutional layers are stacked to build

CNN architecture [21]. The critical layer of CNN architecture is the convolutional layer, which contains the kernel to extract data features. The convolution process calculation in the convolutional layer is shown in (1). The pooling layer had a task to reduce the dimensionality of data which impacts the training cost networks. The last fully connected layer received flattened input from the convolutional and pooling layer to produce the final output [16].

$$c_t = tanh(x_t * w_t + b_t) \tag{1}$$

The value of c_t represents the result of the convolutional process, which is formed using the $tanh$ activation function. The input vector x_t, the convolutional kernel weight w_t, and the convolutional kernel bias b_t are used in $tanh$ activation function. These components collectively contribute to calculate c_t, which reflects the converted and filtered information derived from the input.

Table 1. Dataset of stock issuer grouped by sector

Sector	Issuer	Data Row
Raw goods	89	308,750
Non-primary consumer goods	121	362,480
Primary consumer goods	97	283,436
Energy	67	195,815
Infrastructure	52	133,130
Health	24	61,862
Finance	103	364,665
Industry	51	171,737
Property	69	209,202
Technology	29	43,319
Transportation	25	60,955
Total	727	2,588,451

2.2.2. Long short-term memory

Long short-term memory (LSTM) is designed to solve vanishing gradient limitations in standard recurrent neural networks (RNN). Three gates link memory cells in LSTM networks: forget, input, and output. All gates had the task of controlling the flow information-passed of the network [7]. LSTM implementation is shown in (2) to (7).

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{2}$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3}$$

$$\tilde{c}_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{5}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{6}$$

$$h_t = o_t * tanh\ (c_t) \tag{7}$$

The forget gate f_t is calculated using the sigmoid activation function σ, the input value of the current state x_t, and the output value from the previous cell h_{t-1}. The forget gate weight is W_{xf} and W_{hf}. The forget bias represented by b_f. The input gate i_t is determined using the sigmoid activation function σ, the current state input value x_t, and the preceding cell output value h_{t-1}. W_{xi} and W_{hi} are the input gate weights. The input gate bias is represented by b_f. The output gate o_t is calculated by combining the sigmoid activation function σ, the current state input value x_t, and the preceding cell output value h_{t-1}. W_{xo} and W_{ho} are the output gate weights. The output gate bias is represented by b_o. The candidate current state $\tilde{c}_t$ is calculated using the activation function $tanh$, the current state input value x_t, and the output value h_{t-1} of the preceding cell. The candidate current state weights are W_{xc} and W_{hc}. b_c represents the forget gate bias. The current state c_t is updated using forget gate f_t, previous state c_{t-1}, input gate i_t, and the candidate current state $\tilde{c}_t$. The current state c_t and output gate o_t are used to produce the final output h_t.

2.2.3. Gated recurrent unit

Another variant of RNN is GRU, which is simple than LSTM architecture. GRU architecture only has two gates to control information on neural networks. The reset gate had the task of forgetting information from the previous state. The update gate had to control how much information was added to the current state [22]. GRU Implementation is shown in (8) to (11).

$$u_t = \sigma(W_u * [h_{t-1}, x_t]) \tag{8}$$

$$f_t = \sigma\left(W_f * [h_{t-1}, x_t]\right) \tag{9}$$

$$\tilde{h}_t = tanh(W_{\tilde{h}} * [f_t * h_{t-1}, x_t]) \tag{10}$$

$$h_t = (1 - u_t) * h_{t-1} + u_t * \tilde{h}_t \tag{11}$$

The forget gate u_t and update gate f_t are produced using the sigmoid activation function σ, input the current state x_t, and the previous output state h_{t-1}. The candidate current state $\tilde{h}_t$ calculated using activation function $tanh$, forget gate f_t, the previous output state h_{t-1} and input the current state x_t. Final output GRU h_t is used update gate u_t, the previous output state h_{t-1} and the candidate current state $\tilde{h}_t$. The weight on each gate is represented by W_u, W_f, and $W_{\tilde{h}}$.

2.2.4. Graph convolutional networks

GCN address limitations in prior semi-supervised node classification, in which the edges encode only node similarity and optimize a multi-step pipeline [23]. GCN used graph-structured data as input, which contains node and edge, and then used a matrix of the undirected graph in the convolution process [20]. The implementation of GCN layer-wise propagation is shown in (12).

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{12}$$

$H^{(l)}$ is the matrix of activations in the l^{th}, with $H^{(0)} = X$. The activations matrix produced using activation function $\sigma = ReLU$. The adjacency matrix of the undirected graph $\tilde{A}$, and layer-specific trainable weight matrix $\widetilde{D}$ and $W^{(l)}$.

2.3. Experimental setup

As shown in Figure 1, each stock issuer from Table 1 past pre-processing process to identify missing data was removed from the dataset. The dataset is split into 80% training data and 20% testing data, and the amount of data depends on the date IPO of each stock issuer. The model consistency was tested with 5, 10, 15, 20, and 25 time-lag data. The various time-lag represents the unstable environment of the stock market. The pre-processing process results in an input matrix time series consisting of number samples, time lag, and features, or an input matrix undirected graph with dimensions consisting of number samples, nodes, and nodes. This research produced the nodes representation of time-lag features and edges by correlation score.

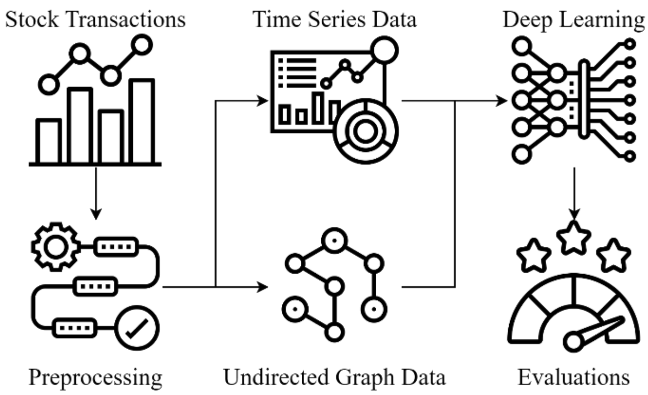

Figure 1. Proposed comparison framework on stock forecasting model in IDX

The CNN, LSTM, GRU, and GCN layers from the TensorFlow (TF) framework are used to learn the input matrix and resolve stock price forecasting. Standalone layer or two layers combination is used to identify which layer significantly affected stock forecasting. As shown in Table 2, the only model using the GCN layer receives an input matrix undirected graph and input matrix time-series; the other only uses input matrix time-series.

Table 2. Model layer parameter

Architecture	Layer Parameter
TFCNN	Input(x) → Conv1D (32 filters, 3 kernels, ReLU activation) → MaxPooling1D (2 pool) → Flatten → Dense → Output(y)
TFCNNLSTM	Input(x) → Conv1D (32 filters, 3 kernels, ReLU activation) → MaxPooling1D (2 pool) → Flatten → Dense → Reshape → TimeDistributed → LSTM (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
TFCNNGRU	Input(x) → Conv1D (32 filters, 3 kernels, ReLU activation) → MaxPooling1D (2 pool) → Flatten → Dense → Reshape → TimeDistributed → LSTM (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
TFLSTM	Input(x) → LSTM (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
TFLSTMCNN	Input(x) → LSTM (128 units, tanh activation) → Dropout (0.25) → Conv1D (32 filters, 3 kernels, ReLU activation) → MaxPooling1D (2 pool) → Flatten → Dense → Output(y)
TFLSTMGRU	Input(x) → LSTM (128 units, tanh activation) → GRU (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
TFGRU	Input(x) → GRU (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
TFGRUCNN	Input(x) → GRU (128 units, tanh activation) → Dropout (0.25) → Conv1D (32 filters, 3 kernels, ReLU activation) → MaxPooling1D (2 pool) → Flatten → Dense → Output(y)
TFGRULSTM	Input(x) → GRU (128 units, tanh activation) → LSTM (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
TFGCNLSTM	Input(x, g) → GCNConv (128 units, ReLU activation) → LSTM (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
TFGCNGRU	Input(x, g) → GCNConv (128 units, ReLU activation) → GRU (128 units, tanh activation) → Dropout (0.25) → Dense → Output(y)
Adam (0.001 learning rate); MSE loss; Early Stopping (validation loss monitoring, 50 patience, 0.01 minimum delta); 150 epochs	

2.4. Evaluations

This study proposes voting using five evaluation metrics to determine the best architecture stock price forecasting on the IDX. The voting has a rule of 2 of 5 best metrics to determine the best architecture on each stock issuer. The performance comparison used five evaluation metrics: the R^2 score and RMSE to determine architecture fitness with the data and the MSE and MAPE to measure error values from the forecasting model [24]. The last metric used is the $F1$-score to measure the performance model on predicting price movement direction. The actual d_i and predicted $\hat{d}_i$ direction calculation shown in (13) and (14). The up direction is represented by 1, and the down direction is represented by 0.

$$d_i = \begin{cases} 1 & if \;\; y_i - y_{i-1} > 0 \\ 0 & else \; y_i - y_{i-1} \leq 0 \end{cases} \tag{13}$$

$$\hat{d}_i = \begin{cases} 1 & if \;\; \hat{y}_i - y_{i-1} > 0 \\ 0 & else \; \hat{y}_i - y_{i-1} \leq 0 \end{cases} \tag{14}$$

The confusion matrix can be utilized to produce the $F1$-score. The true positive (TP), true negative (TN), false positive (FP), and false negative (FN) calculate using d_i and $\hat{d}_i$ value. The $F1$-score calculation shown in (15) utilized precision (P) and recall (R) based on TP, TN, FP, and FN [25].

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{15}$$

3. RESULTS AND DISCUSSION

The code and the experiment results are publicly available on a GitHub repository to make the research reproducible [26]. This study uses the layer CNN, GRU, LSTM, and GCN to produce 11 architectures and forecast stock prices the next day. The voting with minimum rule best 2 of 5 evaluation metrics determines outstanding architecture. The TFGRU is the best architecture that produced the best performance on 315 of 727 stock issuers. TFGRULSTM architecture is the second-best architecture, producing the best performance on 148 companies.

As shown in Figure 2, the rank four lowest achieved by models use two layers, and the first layers use GCN or CNN. The convolutional layer cannot extract essential features from input data and then pass them into GRU or LSTM layer to learn sequence patterns. The GCN or CNN is the best architecture for some stock issuers but not good enough for the rest companies. The top three best performances are produced by GRU or LSTM layers, meaning GRU and LSTM are the best layers to build stock price forecasting on the IDX.

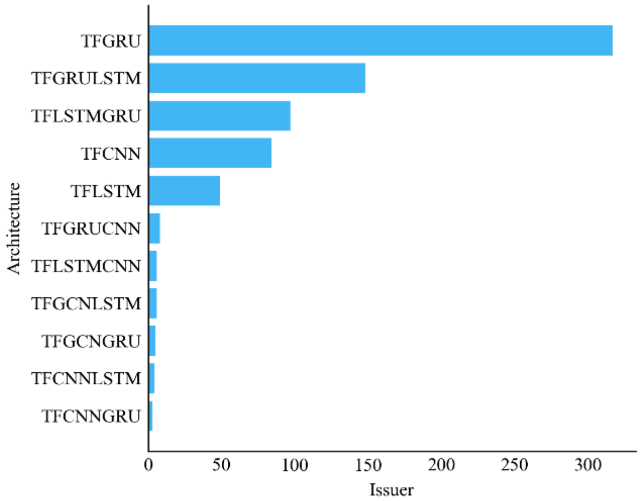

Figure 2. Architecture performance comparison on stock price forecasting

The performance comparison of architecture forecasting is breakdown into four evaluation metric regressions using average scores from 727 stock issuers, presented in Table 3. Confirmed the TFGRU as outstanding architecture with MSE, RMSE, and MAPE had the lowest error. The TFGCNLSTM outperforms the others model with the highest R^2 score, but the average R^2 score on all architectures is negative, meaning all architectures do not fit almost stock issuers. The TFGCNGRU is the worst architecture using MSE, RMSE, and MAPE evaluation. TFCNNGRU achieved the lowest fitted model in the IDX.

Table 3. Architecture score evaluation comparison on stock price forecasting

Architecture	R^2	MSE	RMSE	MAPE	F1	ACC
TFCNN	-75.4127	267,375,025.76	719.9594	1.2676	0.4647	0,4968
TFCNNGRU	-89.6234	300,788,274.08	798.2403	1.5921	**0.4658**	0,5060
TFCNNLSTM	-64.5708	606,576,469.96	1,084.4894	2.8006	0.4590	0,4999
TFGCNGRU	-42.6948	126,577,901,386.31	13,219.6636	42.9366	0.4570	0,5067
TFGCNLSTM	**-29.2006**	28,398,994,058.88	6,362.0688	19.8911	0.4494	0,5033
TFGRU	-40.1303	**127,978,064.19**	**553.3277**	**0.8590**	0.4567	0,5169
TFGRUCNN	-44.0986	493,147,958.88	977.2448	2.3392	0.4430	0,4998
TFGRULSTM	-36.0197	179,564,424.55	651.5528	1.3272	0.4601	0,5120
TFLSTM	-33.0115	177,654,498.63	638.0830	1.2922	0.4631	**0,5174**
TFLSTMCNN	-57.0220	532,755,647.04	995.0271	2.3751	0.4452	0,5000
TFLSTMGRU	-38.0457	192,836,396.32	662.0842	1.3458	0.4622	0,5150

The prediction evaluation of the stock price movement direction is essential because the prediction error can inflict financial loss on investors. The actual movement direction is down, but the model predicted is up. Then the investor decides to sell the stock, and the investor will lose the deviation between the actual value and predicted value multiplied by the shares sold.

The highest f1-score in Table 3 is TFCNNGRU architecture, with a score is 0.4658. All architecture had performance under 0.5, meaning the investor has under 50% chance to take profit from selling shares if using one architecture to all stock issuers. The TFLSTM outperformed the other architecture based on accuracy performance with an accuracy is 0.5174. Based on accuracy performance, all architectures had high risk if used in the real-time stock market, and deep analysis is needed to pick the best architecture for each stock issuer.

The top ten issuers in Table 4 are elected by sorting the highest f1-score, highest accuracy, lowest MAPE, lowest RMSE, lowest MSE, and highest R^2. The TFGRU architecture still dominates to produce the best performance on issuer SRIL, ARKA, PURA, LCKM, and SOFA. The highest f1-score achieved by SRIL with a score is 0.81 and accuracy is 0.99. The SRIL issuer used a time lag of 10 and TFGRU architecture to produce low risk for investors because it has high accuracy and f1-score on prediction stock price movement and a low MAPE is 0.03. The second-best issuer is NANO, with a time lag feature of 5, and TFCNNGRU architecture produced an f1-score of 0.80, an accuracy score of 0.90, and an MSE score of 0.92.

In Table 3, TFGCNLSTM is the second worst architecture by MSE, RMSE, and MAPE evaluation, but in Table 4, TFGCNLSTM is used by the issuer JKSW and PURE from the raw goods sector to produce the best performance. The issuer JKSW has a MAPE of 0.0003 with an accuracy of 0.97 and an f1-score of 0.78. The issuer PURE has an accuracy of 0.99 and an f1-score of 0.75 with a MAPE of 0.01. The TFGCNLSTM is the second worst on overall performance comparison, but it is still needed on some stock issuers to get the best forecasting performance. As shown in Table 4, the various architecture is used to produce the highest performance on each stock issuer, meaning each issuer needed fitted parameters and architecture to achieve the best performance.

Based on votes, performance comparison, and top ten company analysis, TFGRU architecture was determined to be the best architecture for stock forecasting in IDX. The TFGRU yielded the best results on 315 companies, with an average R2 is -40.1303, MSE is 127,978,064.19, RMSE is 553.327, MAPE is 0.858, and f1-score is 0.456. On five of the top ten stock issuers with the best model performance, the TFGRU is the best architecture. The TFGRU evolved into the ideal architecture due to its ability to extract data sequences from time series data.

Table 4. Best-fit parameter on top ten issuers with best model performance

Code	Sector	Lag	Architecture	R^2	MSE	RMSE	MAPE	ACC	F1
SRIL	Non-Primary Consumer Goods	10	TFGRU	0.51	17.81	4.22	0.03	0.99	0.81
NANO	Primary Consumer Goods	5	TFCNNGRU	0.13	0.92	0.96	0.03	0.90	0.80
JKSW	Raw Goods	5	TFGCNLSTM	0.17	0.38	0.62	0.00	0.97	0.78
BOBA	Primary Consumer Goods	10	TFGRUCNN	0.26	20.30	4.51	0.02	0.79	0.78
ARKA	Industry	15	TFGRU	-24.17	55.85	7.47	0.05	0.99	0.75
PURE	Raw Goods	25	TFGCNLSTM	-10.57	0.20	0.45	0.01	0.99	0.75
PURA	Transportation	10	TFGRU	0.21	0.45	0.67	0.01	0.93	0.73
LCKM	Infrastructure	5	TFGRU	0.03	94.61	9.73	0.03	0.77	0.73
SOFA	Non-Primary Consumer Goods	5	TFGRU	0.70	1.76	1.33	0.04	0.90	0.73
BTEK	Primary Consumer Goods	5	TFCNNLSTM	0.70	32.67	5.72	0.10	0.93	0.73

4. CONCLUSION

This research proposed benchmarking the performance CNN, GRU, LSTM, and GCN layer to stock price forecasting on 727 companies in the IDX. Consensus on the best architecture has been fulfilled with TFGRU as the best model forecasting in the IDX. The architecture built from LSTM or GRU layer produces the finest results on performance comparison and confirms best result on previous work is influenced by LSTM or GRU layer. The various time lags and different architectures are used on the top ten stock issuers with the best forecast performance, meaning fitted parameters and architecture are needed on each stock issuer to produce optimum model performance. This comparative research can still be refined to boost existing performance. The hybrid forecasting and classification model can be developed to increase the performance of predicting stock price movement direction without lowering the performance of regression forecasting.

ACKNOWLEDGEMENTS

This research was funded by Asian Development Bank under HETI ADB Project; The Indonesian Ministry of Education and Culture under *Penelitian Terapan Unggulan Perguruan Tinggi* (PTUPT) Program; Institut Teknologi Sepuluh Nopember (ITS) under *Penelitian Dana Departemen Program*; and Institut Teknologi Sepuluh Nopember (ITS) under project scheme of the Publication Writing and IPR Incentive Program (PPHKI).

REFERENCES

[1] W. Buszko Michałand Orzeszko and M. Stawarz, “COVID-19 pandemic and stability of stock market—a sectoral approach,” *PLOS ONE*, vol. 16, no. 5, May 2021, doi: 10.1371/journal.pone.0250938.

[2] M. Mallikarjuna and R. P. Rao, “Evaluation of forecasting methods from selected stock market returns,” *Financial Innovation*, vol. 5, no. 1, Dec. 2019, doi: 10.1186/s40854-019-0157-x.

[3] Indonesia Stock Exchange, “IDX yearly statistics 2022.” Indonesia Stock Exchange, Jakarta, 2023, Accessed: May 15, 2023. [Online]. Available: https://www.idx.co.id/Media/5pfnt4gg/idx_yearly_statistics_2022.pdf.

[4] A. Syukur and D. Istiawan, "Prediction of LQ45 index in indonesia stock exchange: a comparative study of machine learning techniques," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 1, pp. 453–463, Feb. 2021, doi: 10.22266/ijies2021.0228.42.

[5] A. Y. Wiiava, C. Fatichah, and A. Saikhu, "Stock price prediction with golden cross and death cross on technical analysis indicators using long short term memory," in *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, Aug. 2022, pp. 278–283, doi: 10.1109/ICOIACT55506.2022.9971844.

[6] H. Subekti and D. Saepudin, "Cross-sectional machine learning approach on predicting stock return of LQ45 index," in *2022 1st International Conference on Software Engineering and Information Technology (ICoSEIT)*, Nov. 2022, pp. 192–197, doi: 10.1109/ICoSEIT55604.2022.10030044.

[7] S. Hansun and J. C. Young, "Predicting LQ45 financial sector indices using RNN-LSTM," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00495-x.

[8] M. Prastuti, L. Aridinanti, O. Trisnawati, and V. S. Zullah, "Hybrid time series and artificial neural network models for forecasting of the banking stock prices during COVID-19 pandemic," in *AIP Conference Proceedings*, 2022, vol. 2641, no. 1, doi: 10.1063/5.0115035.

[9] W. Budiharto, "Data science approach to stock prices forecasting in Indonesia during Covid-19 using long short-term memory (LSTM)," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00430-0.

[10] U. Farouk, I. Abdulrahman, N. Ussiph, and B. Hayfron-Acquah3, "A hybrid arima-LSTM model for stock price prediction," *International Journal of Computer Engineering and Information Technology*, vol. 12, no. 8, pp. 48–51, Oct. 2020, [Online]. Available: www.ijceit.org.

[11] L. Cagliero, J. Fior, and P. Garza, "Shortlisting machine learning-based stock trading recommendations using candlestick pattern recognition," *Expert Systems with Applications*, vol. 216, 2023, doi: https://doi.org/10.1016/j.eswa.2022.119493.

[12] C. Primasiwi, R. Sarno, K. R. Sungkono, and C. S. Wahyuni, "Stock composite prediction using nonlinear autoregression with exogenous input (NARX)," in *2019 12th International Conference on Information and Communication Technology and System (ICTS)*, Jul. 2019, pp. 43–48, doi: 10.1109/ICTS.2019.8850956.

[13] B. T. Khoa and T. T. Huynh, "Forecasting stock price movement direction by machine learning algorithm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 6, pp. 6625–6634, 2022, doi: 10.11591/ijece.v12i6.pp6625-6634.

[14] C. Manjunath, B. Marimuthu, and B. Ghosh, "Analysis of Nifty 50 index stock market trends using hybrid machine learning model in quantum finance," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 3549–3560, Jun. 2023, doi: 10.11591/ijece.v13i3.pp3549-3560.

[15] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLOS ONE*, vol. 14, no. 2, Feb. 2019, doi: 10.1371/journal.pone.0212320.

[16] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Computing and Applications*, vol. 33, no. 10, pp. 4741–4753, May 2021, doi: 10.1007/s00521-020-05532-z.

[17] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, "Prediction model for stock price trend based on recurrent neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 745–753, 2021, doi: 10.1007/s12652-020-02057-0.

[18] Z. Shi, Y. Hu, G. Mo, and J. Wu, "Attention-based CNN-LSTM and XGBoost hybrid model for stock prediction," *arXiv preprint*, vol. 14, no. 8, pp. 1–7, 2022, doi: 10.48550/arXiv.2204.02623.

[19] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.

[20] X. Yin, D. Yan, A. Almudaifer, S. Yan, and Y. Zhou, "Forecasting stock prices using stock correlation graph: a graph convolutional network approach," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1–8, doi: 10.1109/IJCNN52387.2021.9533510.

[21] Y. Hao and Q. Gao, "Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning," *Applied Sciences*, vol. 10, no. 11, Jun. 2020, doi: 10.3390/app10113961.

[22] C. Manjunath, B. Marimuthu, and B. Ghosh, "Deep learning for stock market index price movement forecasting using improved technical analysis," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 129–141, Oct. 2021, doi: 10.22266/ijies2021.1031.13.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017, pp. 1–14, doi: 10.48550/arXiv.1609.02907.

[24] D. R. Wijaya, R. Sarno, and E. Zulaika, "DWTLSTM for electronic nose signal processing in beef quality monitoring," *Sensors and Actuators B: Chemical*, vol. 326, Jan. 2021, doi: 10.1016/j.snb.2020.128931.

[25] A. T. Haryono, R. Sarno, and R. Abdullah, "Aspect-based sentiment analysis of financial headlines and microblogs using semantic similarity and bidirectional long short-term memory," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 3, pp. 233–241, 2022, doi: 10.22266/ijies2022.0630.20.

[26] A. T. Haryono, "Indonesia stock forecasting," *GitHub*. 2023, Accessed: Jul. 08, 2023. [Online]. Available: https://github.com/agusth24/indonesia-stock-forecast.

BIOGRAPHIES OF AUTHORS

Agus Tri Haryono was born in Balikpapan in 1988. He received a bachelor's degree in computer science from Universitas Mulawarman in 2011. He is currently pursuing the master's degree in informatics engineering from the Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember (ITS). His research interests include text mining and time series. He can be contacted at email: agustharyono24@gmail.com.

Riyanarto Sarno is a Professor, Informatics Department, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. He received the bachelor's degree in electrical engineering from Institut Teknologi Bandung, Bandung, Indonesia in 1987. He received M.Sc. and Ph.D. in computer science from the University of Brunswick Canada in 1988 and 1992, respectively. In 2003 he was promoted to a full professor. His teaching and research interest include internet of things, process aware information systems, intelligent systems and smart grids. He can be contacted at email: riyanarto@if.its.ac.id.

Kelly Rossa Sungkono was born in Surabaya, in June 1994. She received the master's degree in computer engineering, in 2016. She is currently a lecturer at the Institut Teknologi Sepuluh Nopember (ITS). Her research interests include databases, information systems, and machine learning. She can be contacted at email: kelsungkono@gmail.com.