Tri Andi
Magister of Information Technology,
Universitas Teknologi Digital Indonesia,
Yogyakarta, Indonesia
email: triandi@ieee.org

Widyastuti Andriyani[1]
Magister of Information Technology,
Universitas Teknologi Digital Indonesia,
Yogyakarta, Indonesia
email: widya@utdi.ac.id

Bambang Purnomosidi Dwi Putranto
Magister of Information Technology,
Universitas Teknologi Digital Indonesia,
Yogyakarta, Indonesia
email: bpdp@utdi.ac.id

Deep Learning Architecture for Stock Price Prediction

In the world of stock investment, the ability to accurately predict stock price movements is essential. The two main issues that are the focus of this research are, how is the modeling of N-BEATS compared to LSTM and ARIMA on Bank BCA stock prices, and what are the forecasting results of the N-BEATS, LSTM, and ARIMA models on Bank BCA stock data. To answer this, this study discusses the development and evaluation of the N-BEATS time series forecasting model. However, the analysis results show that the ARIMA model shows superior performance, with a MAPE achievement of 0.001% in minute data, 0.006% in hourly data, and 0.018% in day data. This advantage is significant compared to the N-BEATS and LSTM models. Therefore, ARIMA models show great potential for use in financial time series forecasting, risk assessment, and modeling by financial analysts.

***KeyWords**: Stocks, N-BEATS, LSTM, ARIMA*

This Article was:
submited: 14-05-24
accepted: 19-06-24
publish on: 20-07-24

How to Cite:
T. Andi, et al, "Deep Learning Architecture for Stock Price Prediction", Journal of Intelligent Software Systems, Vol.3, No.1, 2024, pp.7–12, 10.26798/jiss.v3i1.1343

1 Introduction

In the current digital era, investment in the stock market has emerged as a popular choice among several individuals and institutions. One critical aspect of investment is the ability to predict future stock price movements, thereby assisting investors in making informed decisions. In the last few decades, prediction methods have made significant progress along with developments in information technology and computer science[1]. In this context, there are several technological approaches that stand out in predicting stock prices, namely N-BEATS, ARIMA, and LSTM.
The ARIMA (Autoregressive Integrated Moving Average) model is a well-established statistical model that has been widely used for time series analysis[2]. Metode ini memanfaatkan gabungan model autoregressive dan moving average dalam rangka memproyeksikan data deret waktu. Although ARIMA has been shown effective in many situations, it has limitations, particularly when the data exhibits non-linear characteristics or is influenced by complex external factors[3].
The LSTM (Long Short-Term Memory) is a variant of artificial neural networks known as Recurrent Neural Networks (RNN)[4]. The advantage of LSTM lies in its ability to retain long-term and short-term information, making it highly suitable for time series prediction, including stock prices. LSTM has demonstrated excellent performance in various stock price prediction applications, especially when applied to data that has a non-linear nature[5].
N-BEATS, an abbreviation for Neural Basis Expansion Analysis, is a relatively recent time series prediction model that has demonstrated promising results in several applications[6]. In contrast to LSTM, N-BEATS is designed with a more flexible architecture that can accommodate various types of inputs, including non-stationary data[7].

[1]Corresponding Author.

In the context of the development of sophisticated prediction methods such as N-BEATS, ARIMA, and LSTM, a deep understanding of the strengths and weaknesses of each method becomes important for investors and researchers. By gaining this understanding, we can increase the chances of success in predicting stock prices and improving investment strategies.

2 Material and Methods

2.1 Data Collection. BCA (Bank Central Asia) stock data was obtained from Yahoo Finance[8], from November 1 2021 to January 6 2023. The data were in real time data based on days, hours and minutes. After the data were collected, the next step were to create a windowing dataset. It should be noted that windowing is a method for converting time series datasets into supervised learning problems.

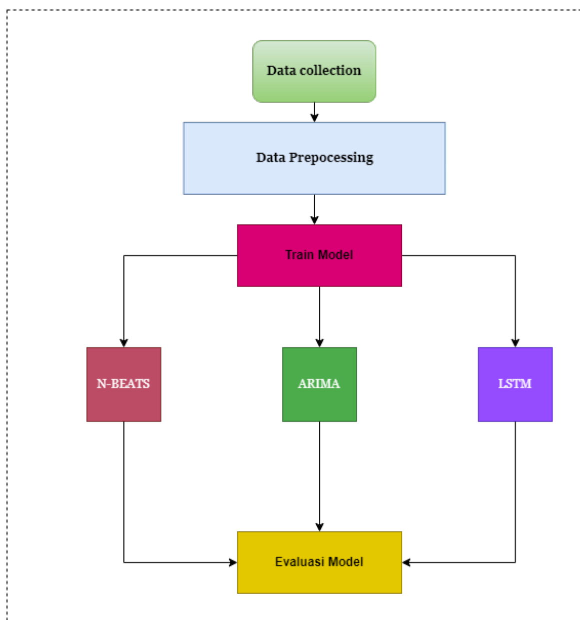

Fig. 1 Research Process Flow

2.2 Research Process Flow.

For a univariate time series, windowing over a week (window=7) to predict the next single value (horizon=1) might look something like:
Windowing for one week (univariate time series)

$$[0, 1, 2, 3, 4, 5, 6] \rightarrow [7]$$

$$[1, 2, 3, 4, 5, 6, 7] \rightarrow [8]$$

$$[2, 3, 4, 5, 6, 7, 8] \rightarrow [9]$$

or for BCA shares, it will look like this:

$$[7450747574757475745074257400] \rightarrow Label : [7400]$$

$$[7475747574757450742574007400] \rightarrow Label : [7350]$$

$$[7475747574507425740074007350] \rightarrow Label : [7350]$$

$$[7475745074257400740073507350] \rightarrow Label : [7350]$$

$$[7450742574007400735073507350] \rightarrow Label : [7350]$$

After windowing the dataset, the next step is to divide the results of the windowing dataset into training sets and test sets. The percentage of the distribution is 80% of the data were used for training data and the remaining 20% were as test data. The data werw divided sequentially, the test data must be data from the future when compared with the training data Figure **??**.

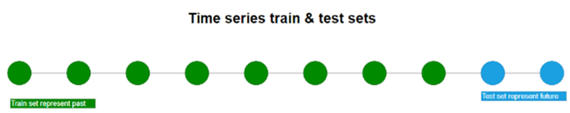

Fig. 2 Distribution of Training dan Test Data

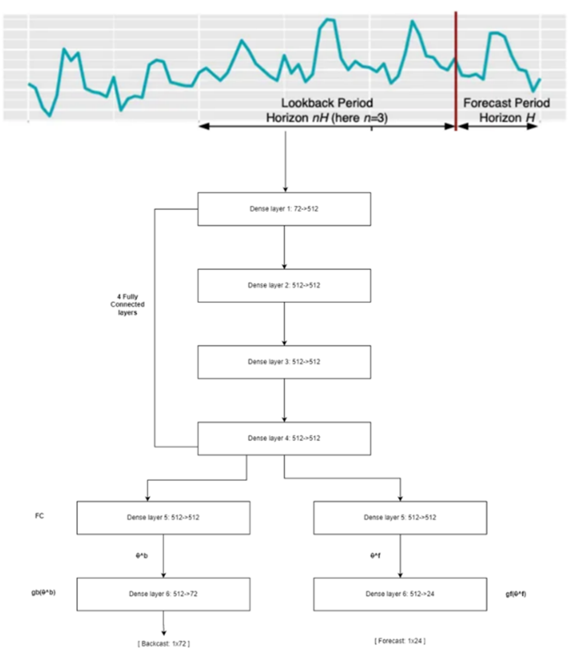

Fig. 3 All Operations Are Inside Basic Block

2.3 N-BEATS.

N-Beats (Neural Basis Expansion Analysis For Interpretable Time Series Forecasting) is a pure deep learning architecture based on an ensemble feed forward network stack that is also stacked by connecting backcast and forecast links[6], [9]. Each successive block only models the residual error due to backcast reconstruction of the previous block and then updates the estimates based on that error. This process mimics the Box-Jenkins method when installing an ARIMA model[10].
Explanation regarding the image above:

- The model looks back 3 days= 72 hours= 3 horizons to predict power usage 24 hours into the future.
- Blocks receive lookback window input..
- The input is then passed through a 4-layer neural network.
- The results of this calculation are directed to 2 outputs. Here, the dense layer 5 estimates the theta parameters (θ^b and θ^f), which are called expansion coefficients.
- These parameters are then projected linearly into the new space using base layer transformations g^b and g^f to generate backcast and forecast signals. This process is called "nerve base expansion".

It should be noted that the backcast signal is an approximate vector that can optimally predict the forecast signal, given the g^b and g^f transformations. When g^b and g^f take a certain form, the backcast and forecast vectors become interpretable (more on that later).

2.4 ARIMA.

Autoregressive Integrated Moving Average (ARIMA) model is a time series model that combines the Autoregressive (AR) and Moving Average (MA) models. In the ARIMA model, all input data must be stationary, so we must first check whether the data is stationary or not. A data can be said to be stationary if the data pattern is in equilibrium around a constant average value and the variance around the average is constant for a certain time[1]. In this case, the stationarity of the data can be checked by analyzing the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF)[7], [11], [12].
Autocorrelation is a correlation or relationship between observational data of a time series data, which is one of the indicators to determine the order of q in the ARIMA model (p, d, q). simple correlation coefficient Y_t with Y_{t-1} can be found by the following formula[13]:

$$r_k = \frac{\sum_t^n (Y_t - \bar{Y})(Y_{t-1} - \bar{Y})}{\sum_t^n (Y_t - \bar{Y})^2} \tag{1}$$

where:
r_k = *lag* auto correlation coefficient to k
k = 0, 1, 2, 3, ...,k
n = amount of data
Y_t = data values from stationary time series
$\bar{(Y)}$ = random variable mean Y.

2.5 LSTM.

Long short term memory is an architecture developed by[14]. This architecture was developed to overcome the weaknesses of the recurrent neural network (RNN) architecture to study problems that require long-term memory[4],[5],[7].
Long short term memory has four core units as shown in Figure 4, cell state, forget gate, input gate and output gate. The units are interconnected so that the input will be selectively stored, determined, and forgotten at each gate until it finally gets the final output.

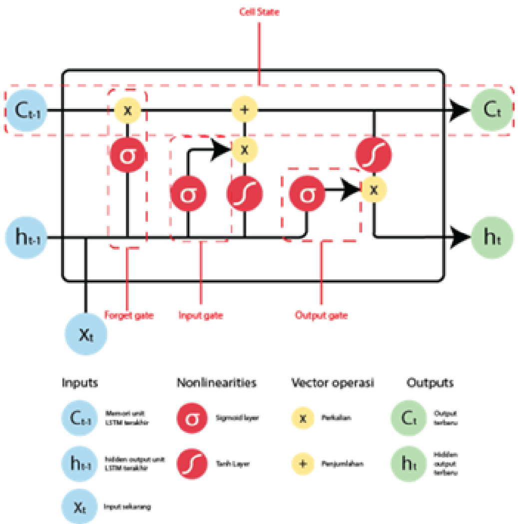

Fig. 4 LSTM Architecture

2.6 Hyperparameter Tuning Process. Detailed hyperparameter tuning process for each model used in the study: ARIMA, LSTM, and N-BEATS.

(1) ARIMA (Autoregressive Integrated Moving Average), ARIMA is a statistical model used for time series analysis and forecasting. Hyperparameter alignment in ARIMA involves determining three main parameters: p, d, and q.

 (a) Initial Model Selection

  i. Start with the raw data and ensure the data is stationary using the ADF (Augmented Dickey-Fuller) test.

  ii. If the data is not stationary, perform differencing until the data becomes stationary.

 (b) Parameter Determination d

  i. ddd is the order of differencing required to make the data stationary.

  ii. Test various differencing values (e.g., 0, 1, or 2) and select the ddd value that makes the data stationary without too much differencing.

 (c) Determination of p and q Parameters

  i. Use the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to determine the initial values of p and q.

  ii. p is an autoregressive sequence, which can be identified by looking at the significant PACF points.

  iii. q is the moving average sequence, which can be identified by looking at the significant ACF points.

 (d) Model Evaluation: Use information criteria such as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) to compare different ARIMA models and select the model with the lowest AIC/BIC value.

 (e) Cross-validation: Perform cross-validation by dividing the data into a training set and a testing set. Test the model performance on the test set to ensure the model is not overfitting.

 (f) Parameter Optimization: Use grid search to try different combinations of ppp, ddd, and qqq values and select the combination that gives the best performance on the validation data.

(2) LSTM (Long Short-Term Memory) LSTM is a type of artificial neural network that is specifically used to process time series data and handle long-term dependencies.

 (a) Determination of Network Architecture

  i. Start by determining the number of layers of the LSTM and the number of units in each layer

  ii. Test several configurations (e.g., one layer with 50 units, two layers with 100 units, etc.) and choose the one that gives the best results on the training data.

 (b) Main Hyperparameters

  i. Number of LSTM Units: The number of neurons in each layer. More units can capture more patterns, but also require more data and computation.

  ii. Learning Rate: The learning rate that determines how much the weights are updated at each iteration. Try different values (e.g., 0.01, 0.001, 0.0001) and choose the most suitable one.

  iii. Batch Size: The number of samples processed in one weight update. Larger batch sizes usually speed up training but require more memory.

  iv. Dropout Rate: The dropout rate to prevent overfitting. Common values are 0.2 or 0.5.

 (c) Hyperparameter Optimization:

  i. Use random search or Bayesian optimization to find the best combination of parameters.

  ii. Conduct multiple experiments with different parameters and evaluate the results using metrics such as MAPE (Mean Absolute Percentage Error) and RMSE (Root Mean Square Error).

 (d) Model Validation

  i. Use cross-validation to verify that the model is not overfitting the training data

  ii. Split the data into a training set and a test set, and test the model's performance on the test set.

(3) N-BEATS (Neural Basis Expansion Analysis) N-BEATS is a deep learning model used for time series forecasting with a flexible architecture.

 (a) Architecture Selection

  i. Determine the number of backcast and forecast blocks. Start with a simple configuration and add complexity gradually.

  ii. Test multiple configurations (e.g., 2-5 blocks with 32-64 units per block).

 (b) Key Hyperparameters

  i. Number of Blocks: The number of backcast and forecast blocks. More blocks can capture more patterns from the data.

  ii. Block Size: The number of neurons in each block. A larger size can capture more complex patterns.

  iii. Learning Rate: The step size for weight updates. Start with a standard value such as 0.001 and make adjustments based on the initial results.

  iv. Epochs: The number of training cycles to run. Use early stopping to avoid overfitting if the number of epochs is too large.

 (c) Hyperparameter Optimization

  i. Use grid search or Bayesian optimization to explore various combinations of parameters and select the one that gives the best results.

ii. Evaluate the model performance using MAPE and RMSE metrics.

(d) Model Validation

i. Use cross-validation to verify that the model does not only fit the training data.

ii. Evaluate the model results on data not seen during training to ensure generalization of the model.

3 Result and Discussion

3.1 Mean Absolute Percentage Error (MAPE). It is a measure of relative accuracy that is used to determine the percentage of deviation from prediction results. The Mean Absolute Percentage Error (MAPE) is calculated using the absolute error for each period divided by the actual observed value for that period[15]. Then, average the absolute percentage errors. This approach is useful when the size or magnitude of the predictive variable is important in evaluating the accuracy of the prediction. MAPE indicates how big the error is in predicting compared to the real value[16][17]. The equation is as follows:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|\hat{Y}_t - Y_t\right|}{Y_t} \times 100 \quad (2)$$

where:
n = amount of data
$\hat{Y}_t$ = Actual Value of the request Y_t = forecasting value

3.2 Root Mean Square Error (RSME). Root Mean Square Error (RMSE), is the sum of the squared errors or the difference between the actual value and the predetermined predicted value][15][18]. The RMSE formula is as follows:

$$RSME = \sqrt{\sum \frac{(Y' - Y)^2}{n}} \quad (3)$$

where:
Y'= actual value of the request
Y = prediction result value n = amount of data

3.3 Model Evaluation. Evaluation of the model that will be used in this study is to use MAPE and RMSE[19]. MAPE is a measure of relative accuracy used to determine the percentage of deviation from prediction results. Mean Absolute Percentage Error (MAPE) is calculated using the absolute error in each period divided by the real observed value for that period[1][2].
This section presents the results of statistical analysis to test the N-BEATS model. effectiveness compared to LSTM and ARIMA. RMSE and MAPE values are used to evaluate forecasting accuracy for the three models. The Arima model on minute, hour and day data gets better results compared to the N-BEATS and LSTM models[20][21][22].

Table 1 Minutely MAPE and RMSE Data

Model	MAPE	RMSE
N-BEATS	0.0798%	12.48
LSTM	0.0740%	11.75
ARIMA	0.00079%	10.1

In table 1, regarding minute data, the ARIMA model has better results than the N-BEATS and LSTM models.

Table 3 Daily MAPE and RMSE Data

Model	MAPE	RMSE
N-BEATS	1,098%	131,61
LSTM	1,352%	168,29
ARIMA	0.018%	210.7

Table 2 Hourly MAPE and RMSE Data

Model	MAPE	RMSE
N-BEATS	0.459%	54.16
LSTM	0.473%	55.82
ARIMA	0.006%	76.0

In Table 2 and Table 3, the test results using Hour and Day data for the arima model are better than the N-BEATS and LSTM models. The findings indicate that all models demonstrate a high level of accuracy in predicting prices. Figure 5 through Figure 10 displays individual graphs representing predictions on minute, hour, and day data for each model.

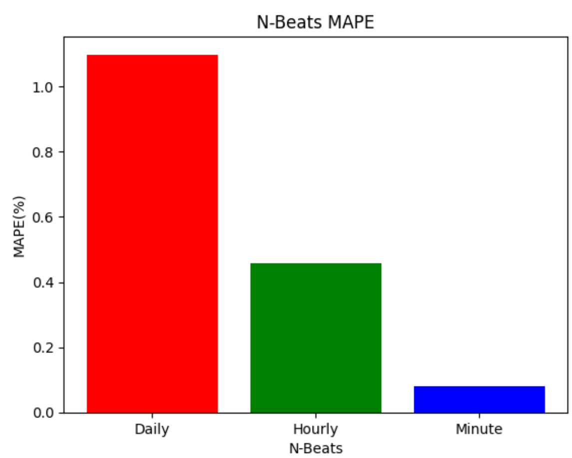

Fig. 5 MAPE Results: N-BEATS

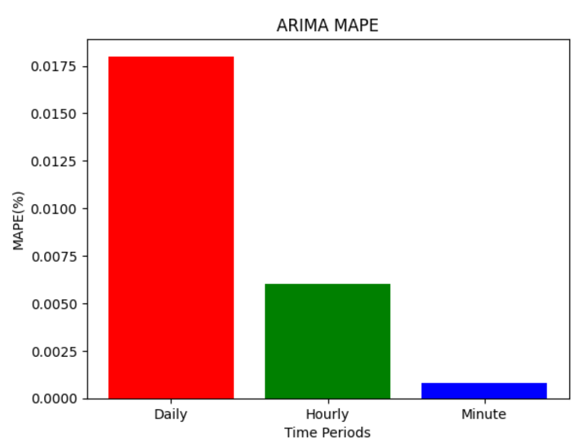

Fig. 6 MAPE Results: ARIMA

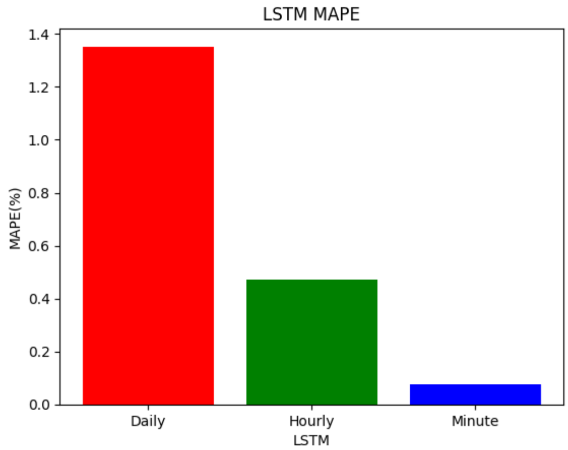

Fig. 7 MAPE Results: LSTM

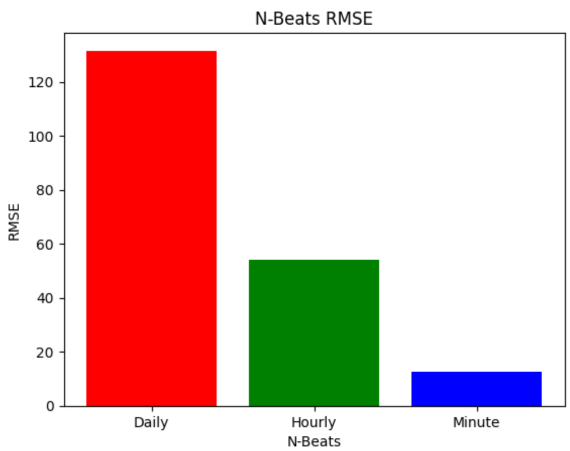

Fig. 8 MAPE Results: N-BEATS

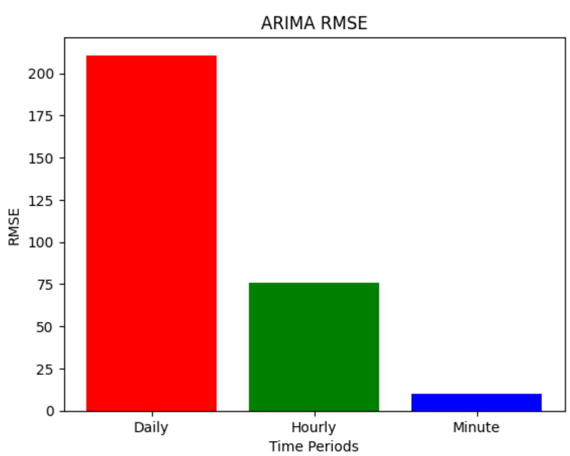

Fig. 9 MAPE Results: ARIMA

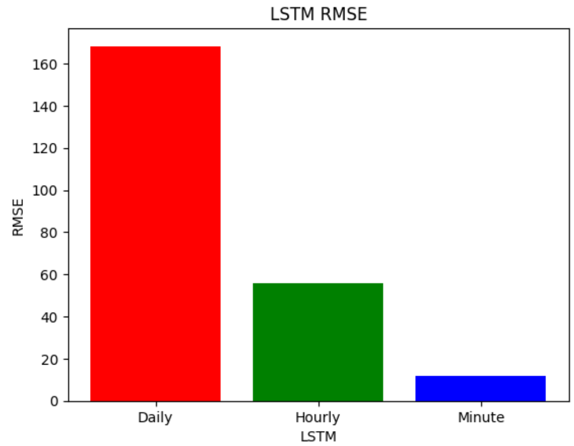

Fig. 10 MAPE Results: LSTM

4 Conclusions

The objective of this study is to present an introduction to the application of deep learning models for time series forecasting, specifically focusing on the utilization of bank stock data for training purposes. The model that has been created demonstrates notable outcomes, with ARIMA exhibiting exceptional performance in the prediction process. Specifically, ARIMA achieves a Mean Absolute Percentage Error (MAPE) of 0.00079% for minute-level data, 0.006% for hourly-level data, and 0.018% for daily-level data. The obtained results demonstrate a significant improvement over the performance of both the N-BETAS and LSTM models. The model that has been constructed has the potential to be utilized by financial analysts for the purposes of forecasting financial time series, conducting risk assessment, and engaging in modeling activities.

References

[1] Thakkar, A. and Chaudhari, K., 2021, "Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions," Information Fusion, **65**, pp. 95–107.

[2] Ma, Q., 2020, "Comparison of ARIMA, ANN and LSTM for Stock Price Prediction," E3S Web of Conferences, **218**, p. 1026.

[3] Garlapati, A., Krishna, D. R., Garlapati, K., Srikara Yaswanth, N. M., Rahul, U., and Narayanan, G., 2021, "Stock Price Prediction Using Facebook Prophet and Arima Models," 2021 6th International Conference for Convergence in Technology, I2CT 2021.

[4] Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., and Khatri, R. K., 2022, "Predicting stock market index using LSTM," Machine Learning with Applications, **9**, p. 100320.

[5] Jin, Z., Yang, Y., and Liu, Y., 2020, "Stock closing price prediction based on sentiment analysis and LSTM," Neural Computing and Applications, **32**(13), pp. 9713–9729.

[6] Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y., 2020, "N-Beats: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting," 8th International Conference on Learning Representations, ICLR 2020.

[7] Liu, H., Mi, X., Li, Y., Duan, Z., and Xu, Y., 2019, "Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional Gated Recurrent Unit network and Support Vector Regression," Renewable Energy, **143**, pp. 842–854.

[8] Y.Finance, 2023, "Yahoo! Finance," doi: 10.5860/choice.43sup-0514, accessed 2023-07-21, https://finance.yahoo.com

[9] Wang, X., Li, C., Yi, C., Xu, X., Wang, J., and Zhang, Y., 2022, "EcoForecast: An interpretable data-driven approach for short-term macroeconomic forecasting using N-BEATS neural network," Engineering Applications of Artificial Intelligence, **114**.

[10] Debnath, K. B. and Mourshed, M., 2018, "Forecasting methods in energy planning models," Renewable and Sustainable Energy Reviews, **88**, pp. 297–325.

[11] Selvaraj, J. J., Arunachalam, V., Coronado-Franco, K. V., Romero-Orjuela, L. V., and Ramírez-Yara, Y. N., 2020, "Time-series modeling of fishery landings in the Colombian Pacific Ocean using an ARIMA model," Regional Studies in Marine Science, **39**.

[12] Xiao, R., Feng, Y., Yan, L., and Ma, Y., 2022, "Predict stock prices with ARIMA and LSTM," .

[13] Makridakis, Spyros, Wheelwright, C, S., McGee, and E., V., 1999, *Metode Dan Aplikasi Peramalan*.

[14] Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y., 2020, "N-Beats: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting," 8th International Conference on Learning Representations, ICLR 2020.

[15] Deng, L., Ruan, K., Chen, X., Huang, X., Zhu, Y., and Yu, W., 2022, "An IP Network Traffic Prediction Method based on ARIMA and N-BEATS," 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems, ICPICS 2022, pp. 336–341.

[16] Singhal, V. and Mathew, J., 2022, "Fusion of Wavelet Decomposition and N-BEATS for improved Stock Market Forecasting," Research Square, pp. 0–21.

[17] El Majzoub, A., Rabhi, F. A., and Hussain, W., 2023, "Evaluating interpretable machine learning predictions for cryptocurrencies," Intelligent Systems in Accounting, Finance and Management, **30**(3), pp. 137–149.

[18] Dhawan, M. and Aggrawal, T., "State-of-the-art vs prominent models: An empirical analysis of various neural networks on stock market prediction," 2022.

[19] Nils Widmark, 2022, "Short-term electricity consumption forecasting using deep learning and external variables," , pp. 8–51.

[20] Ye, J., Gu, J., Dash, A., Deek, F. P., and Wang, G. G., 2023, "Prediction with Time-Series Mixer for the SP500 Index," Proceedings - 2023 IEEE 39th International Conference on Data Engineering Workshops, ICDEW 2023, pp. 20–27.

[21] Gajamannage, K., Park, Y., and Jayathilake, D. I., 2023, "Real-time forecasting of time series in financial markets using sequentially trained dual-LSTMs," Expert Systems with Applications, **223**.

[22] Van Zyl, T. L., 2023, "Late Meta-learning Fusion Using Representation Learning for Time Series Forecasting," 2023 26th International Conference on Information Fusion, FUSION 2023.

