International Journal of Advances in Intelligent Informatics
Vol. 9, No. 2, July 2023, pp. 286-300

Deep learning approaches for MIMO time-series analysis

Fachrul Kurniawan [a,1,*], Sarina Sulaiman [b,2], Siaka Konate [c,3], Modawy Adam Ali Abdalla [d,e,4]

[a] Informatics Engineering, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Kota Malang, Jawa Timur 65144, Indonesia
[b] UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research, Soft Computing Research Group, Universiti Teknologi Malaysia, Malaysia.
[c] Department of Electronic and Telecommunications, Normal School of Technical and Vocational Education, 91094 Bamako, Mali
[d] College of Energy and Electrical Engineering, Hohai University, Nanjing 211100, China
[e] Department of Electrical and Electronic Engineering, College of Engineering Science, Nyala University, Nyala 63311, Sudan
[1] fachrulk@ti.uin-malang.ac.id; [2] sarina@utm.my; [3] konatesiaka77@gmail.com; [4] brojacter88@yahoo.com
* corresponding author

ARTICLE INFO

Article history
Received April 18, 2023
Revised May 31, 2023
Accepted June 6, 2023
Available online June 13, 2023

Keywords
MIMO
Time series
Deep learning
Stock prices

ABSTRACT

This study presents a comparative analysis of various deep learning (DL) methods for multi-input and multi-output (MIMO) time-series forecasting of stock prices. The analysis is conducted on a dataset comprising the stock price of Bitcoin. The dataset consists of 2950 rows from December 2017 to December 2021. This study aims to evaluate the performance of multiple DL methods, including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Unit (GRU). The evaluation criteria for selecting the best-performing methods in this research are based on two performance metrics: Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). These metrics were chosen for specific reasons related to assessing the accuracy and reliability of the forecasting models. MAPE is used to assess accuracy, while RMSE helps detect outliers in the system. Results show that the LSTM method achieves the best performance, outperforming other methods with an average MAPE value of 8.73% and Bi-LSTM has the best average RMSE value of 0.02216. The findings of this study have practical implications for time-series forecasting in the field of stock trading. The superior performance of LSTM highlights its potential as a reliable method for accurately predicting stock prices. The Bi-LSTM model's ability to detect outliers can aid in identifying abnormal stock market behavior. In summary, this research provides insights into the performance of various DL models of MIMO for stock price forecasting. The results contribute to the field of time-series forecasting and offer valuable guidance for decision-making in stock trading by identifying the most effective methods for predicting stock prices accurately and detecting unusual market behavior.

1. Introduction

Time series analysis involves the examination and prediction of data that is collected sequentially over time. This field of study is crucial in various domains, including finance [1], economics [2], meteorology [3], and sales forecasting [4]. However, time series analysis poses several challenges that need to be addressed to ensure accurate and reliable predictions. These challenges can be categorized into two main areas: single-output forecasting and multi-output forecasting. In single-output forecasting, common problems include handling changing trends [5], identifying and modeling seasonal patterns [6], dealing with outliers [7], addressing stationarity assumptions [8], managing autocorrelation, [9] and working with limited data [10]. On the other hand, multi-output forecasting introduces additional complexities such as high dimensionality and the need to account for interactions between multiple output variables [11]. The multi-output forecasting problem has its own challenges because changes in one variable or output can affect other variables or output [12]. Therefore, it is important to consider the interactions and interrelationships between variables when developing an effective multi-output forecasting model.

Multi-Input and Multi-Output (MIMO) time-series forecasting is useful for real-world multivariate data applications. Financial markets use past prices [8], trade volumes [13], market sentiment [14], and macroeconomic data [15] to anticipate stock prices. MIMO time-series forecasting also has other uses. Temperature [16], humidity [17], wind speed [18], and precipitation interact [19] in the weather forecast. MIMO forecasting predicts energy consumption [20], resource availability [21], and production outputs [22] in resource management and optimization. Due to its complexity and real-world applications, the problem of MIMO time-series forecasting has gained attention in recent years [23][24][25][26]. The difficulty in this problem lies in modeling the dependencies between the input and output variables, especially when they have different time resolutions and levels of noise. Traditional statistical and machine learning (ML) approaches have limitations in capturing the dynamics and interactions between variables [27][28]. Therefore, developing a robust and efficient solution for MIMO time-series forecasting for developing robust time series models and obtaining accurate predictions, which have significant implications for decision-making and planning in various fields is crucial.

MIMO Time-series forecasting techniques struggle to represent long-term dependencies [29]. Trends, cycles, and seasonality are long-term dependencies. Various solutions have been proposed to address this problem, including vector autoregression moving average (ARIMA) models [30], multilayer perceptron (MLP) [31], and dynamic Bayesian networks (DBNs) [32][33]. These methods have shown promising results in capturing the nonlinear dependencies between variables and handling missing values and noisy data. However, they still have limitations in modeling long-term dependencies and dealing with high-dimensional data [34]. Therefore, there is a need for more advanced and flexible models that can overcome these limitations and provide accurate and interpretable predictions. Deep learning (DL) models, such as Convolutional Neural Networks (CNNs) [35][36][37], Long Short-Term Memory Networks (LSTMs) [38][39][40], and Gated Recurrent Units (GRU) [41], have been proposed as a promising solution for MIMO time-series forecasting. Therefore, this problem remains an active area of research, and further investigations are needed to develop more efficient and interpretable models.

This paper aims to reveal the performance of five different deep learning approaches: CNN, RNN, LSTM, GRU, and Bidirectional LSTM (Bi-LSTM). Recurrent Neural Networks (RNNs), LSTMs, and GRU can simulate long-term dependencies [42]. Recurrent connections and memory cells allow these models to store and learn from prior observations and dependencies. These models can better anticipate long-term trends by adding memory processes. MIMO time-series forecasting uses high-dimensional data to predict multiple outputs from multiple inputs [43]. Computational, model, and dimensionality issues arise with high-dimensional data. CNNs can handle high-dimensional data [44]. CNNs find data patterns by extracting local and global characteristics from input sequences using convolutional algorithms, making suitable for time-series forecasting. These models can capture the complex temporal patterns and interactions between variables and provide accurate and robust predictions. Moreover, they can handle missing values, noisy data, and high-dimensional input and output variables [45][46]. Developing a DL model for MIMO time-series forecasting requires careful consideration of the model architecture, data preprocessing, and hyperparameter tuning. MLP as baseline of deep learning is used to compare the performance of developed models for stock prediction.

2. Method

2.1. Data Collection

The dataset used is 2950 rows of data from https://www.cryptodatadownload.com/ from December 18, 2017, to December 31, 2021. The dataset consists of 7 attributes whose visualization can be seen in Fig. 1. The description of the dataset can be seen in Table 1.

Table 1. Dataset Describe

index	open	close	high	low	Volume BTC	Volume USDT	tradecount
count	2950	2950	2950	2950	2950	2950	2950
mean	18296.42	18315.04	18836.06	17672.97	54331.53	1126496122.00	817203.89
std	17653.74	17668.82	18172.63	17054.67	35003.00	1426290194.00	756509.22
min	3211.71	3211.72	3276.50	3156.26	1521.53	11770168.04	12417.00
25%	7099.74	7099.735	7295.18	6863.25	31580.60	246861465.70	255846.75
50%	9509.07	9507.64	9709.17	9236.61	45692.96	452651204.90	506273.00
75%	26440.43	26932.90	27477.50	25835.00	67269.47	1561412415.00	1200025.00
max	67525.82	67525.83	69000.00	66222.40	402201.67	13477694935.00	6331062.00

This study uses two attributes as target data: the open attribute and the close attribute. The open attribute is the stock price at the beginning of the trade opening, and the close attribute is the price at the end of the period. The open and close prices are beneficial for analyzing pattern trends in stock prices [47]. Both of these attributes are attributes that affect changing the pattern trend (pattern) that is generated for stock predictions. The difference between the open and close prices can provide insights into the intraday price movement, such as whether the stock experienced a positive or negative trend during the trading session [48]. These trends often indicate investor sentiment, market momentum, and trading strategies market participants employ.

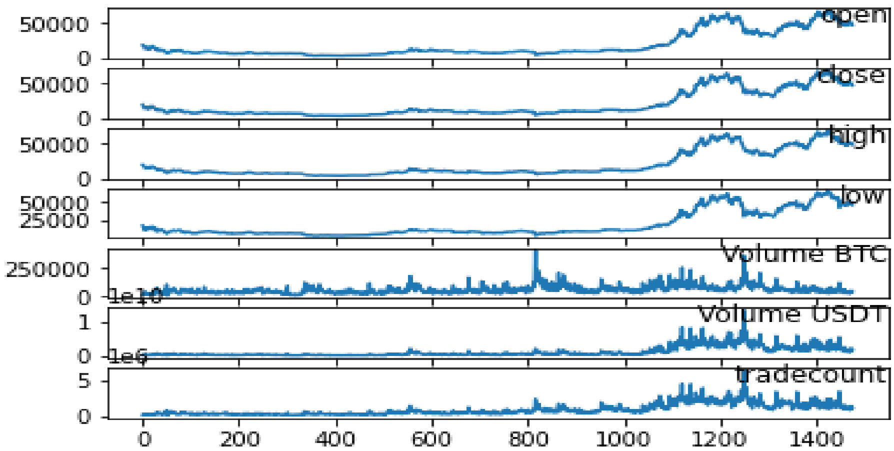

Fig. 1. Visualization of Dataset

Incorporating open and close prices as target variables enables the models to learn and capture these patterns, enhancing forecasting accuracy. Considering that factor, the selection of open and close prices as target variables for the forecasting task allows the models to leverage the intraday dynamics and patterns exhibited by these attributes. By incorporating this information into the training process, the models can learn to capture and utilize the trends and fluctuations in stock prices, leading to more accurate and reliable predictions. The comparison of the values between the close and open attributes can be seen in Fig. 2.

Fig. 2. Graphic Data Open and Close

2.2. Data Preprocessing

Min-max normalization is a common data preprocessing technique used in time-series analysis to scale the data values to a specific range, usually between 0 and 1,s is achieved by subtracting the minimum value from each data point and dividing it by the range between the minimum and maximum values. The resulting values are then scaled to the desired range. Min-max normalization is well-suited for time-series analysis as it addresses time-dependent data's specific scaling and range normalization requirements [49]. It preserves temporal relationships, handles seasonal and trend components, mitigates the influence of outliers, provides an interpretable scale, and is compatible with various modeling techniques.

The first step is to identify the dataset's minimum and maximum values and apply min-max normalization to time-series data, can be done by iterating through each time step and keeping track of the minimum and maximum values [50]. Once these values have been identified, the data can be normalized using the equation (1).

$$x\_norm = (x - min\_val) / (max\_val - min\_val) \quad (1)$$

where x is the original data point, min_val is the minimum value in the dataset, max_val is the maximum value in the dataset, and x_norm is the normalized data point.

It is important to note that min-max normalization should be applied separately to the training and test datasets. The minimum and maximum values used for normalization should be based on the training dataset only, then applied to the test dataset using the same formula, ensuring that the test dataset is not used to inform the normalization process and prevents data leakage.

2.3. MIMO Forecasting

The forecasting model framework in the study can be seen in Fig. 3. Fig. 3 shows that the number of inputs is seven, and the number of outputs is 2. The forecasting process uses various types of methods that have been selected, as shown in Table 2. The first method is Convolutional Neural Networks (CNN), CNN are artificial neural networks that can be used for time-series forecasting. While CNNs were initially developed for image recognition tasks, they have also been applied to sequential data, including time-series data [51]. The main idea behind CNNs is to use filters that convolve over the input data to extract relevant features. These filters are typically small and move across the input data, computing dot products at each location. The outputs from the dot products are then passed through a nonlinear activation function, such as ReLU, and pooled to reduce the dimensionality of the data. This process is repeated multiple times in multiple layers, with each layer capturing more complex features of the input data. In the context of time-series forecasting, CNNs can be used to extract temporal features from sequential data [52]. For example, the filters can convolve over the input time-series data to capture patterns such as trends, cycles, and spikes.

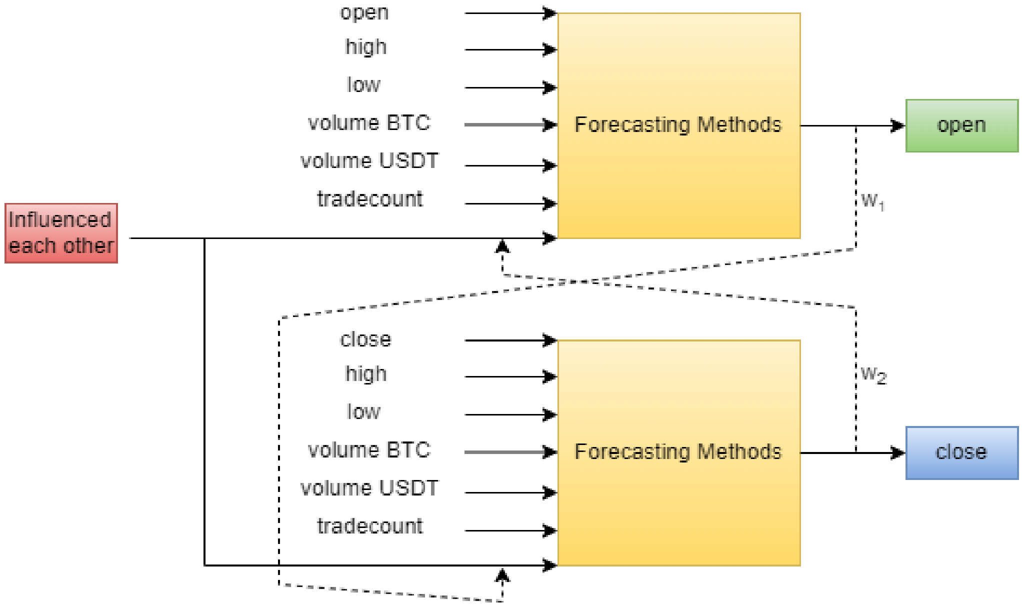

Fig. 3. MIMO Forecasting Scheme

The resulting feature maps can then be fed into fully connected layers to produce a forecast. One advantage of CNNs for time-series forecasting is their ability to capture local and global dependencies in the data. Additionally, CNNs can handle varying input lengths, making them useful for forecasting time-series data with irregular time intervals. The equation of CNN can be seen in equations (2) to (5).

$$H(t) = f(conv(X(t))) \tag{2}$$

$$F(t) = flatten(H(t)) \tag{3}$$

$$D(t) = g(W(f) * F(t) + b(f)) \tag{4}$$

$$Y(t) = D(t) \tag{5}$$

where X(t) represents the input at time step t, H(t) is the hidden state at time step t, F(t) is flattened, D(t) is a dense layer, and Y(t) is the output. In the case of MIMO, the output is Y(t)1 and Y(t)2, which are depend on each other as in (6) and (7).

$$Y(t)1 = D(t) + Y(t)2 \tag{6}$$

$$Y(t)2 = D(t) + Y(t)1 \tag{7}$$

The second is Recurrent Neural Networks (RNN). Unlike traditional feedforward neural networks, which have no memory and process each input independently, RNNs have a memory component that allows them to process sequential data [53]. The key idea behind RNNs is the use of recurrent connections between nodes, which allow information to persist over time. In this way, the network can capture temporal dependencies in the data. RNNs use a hidden state that is updated at each time step, and the output at each time step is a function of the current input and the hidden state. The hidden state is passed from one-time step to the next, allowing the network to learn a representation of the entire sequence. In time-series forecasting, RNNs can predict the next value in a time series based on the previous values. The network is trained using a supervised learning algorithm, such as backpropagation through time, where the loss is minimized between the predicted and actual values of the target variable [54]. One advantage of RNNs for time-series forecasting is their ability to capture long-term dependencies in the data, making them useful for predicting trends and cycles. The equation of RNN can be seen in equations (8) to (9).

$$H(t) = f(W(xh) * X(t) + W(hh) * H(t-1) + b(h)) \tag{8}$$

$$Y(t) = g(W(hy) * H(t) + b(y)) \tag{9}$$

The output in the case of MIMO is $Y(t)1$ and $Y(t)1$, which depend on one another as in (10) and (11).

$$Y(t)1 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)2 \tag{10}$$

$$Y(t)2 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)2 \tag{11}$$

Third, Long Short-Term Memory (LSTM) is a type of RNN commonly used for time-series forecasting. LSTMs were designed to address the limitations of traditional RNNs, such as difficulty in capturing long-term dependencies and vanishing gradients [55]. The key idea behind LSTMs is using a memory cell that can remember information for long periods. Three gates control the memory cell: the input gate, the forget gate, and the output gate. These gates allow the network to update and forget information from the memory cell selectively. In the context of time-series forecasting, LSTMs can predict the next value in a time series based on the previous values [56]. The network is trained using a supervised learning algorithm, such as backpropagation through time, where the loss is minimized between the predicted and actual values of the target variable. One advantage of LSTMs for time-series forecasting is their ability to capture long-term dependencies in the data, making them useful for predicting trends and cycles [57]. Additionally, LSTMs can handle variable-length sequences, making them helpful in forecasting time-series data with irregular time intervals. The equation of LSTM can be seen in equations (12) to (17).

$$\mathrm{C(t)\ =\ f(W(xc)\ *\ X(t)\ +\ W(hc)\ *\ H(t-1)\ +\ b(c))} \tag{12}$$

$$F(t) = \sigma(W(xf) * X(t) + W(hf) * H(t-1) + b(f)) \tag{13}$$

$$I(t) = \sigma(W(xi) * X(t) + Whi) * H(t-1) + b(i)) \tag{14}$$

$$O(t) = \sigma(W(xo) * X(t) + W(ho) * H(t-1) + b(o)) \tag{15}$$

$$H(t) = O(t) * \tanh(C(t)) \tag{16}$$

$$Y(t) = g(W(hy) * H(t) + b(y)) \tag{17}$$

In the MIMO situation, the outputs are the dependent variables Y(t)1 and Y(t)2 as in (18) and (19).

$$Y(t)1 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)2 \tag{18}$$

$$Y(t)2 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)1 \tag{19}$$

In these equations, C(t), F(t), I(t), O(t) sequentially represents the cell state, forget gate, input gate, and output gate of the LSTM. The variables W and b denote the learnable weights and biases of the model, respectively. The activation functions f and g represent the non-linear activation functions applied to the hidden state and output, respectively. σ represents the sigmoid activation function, and tanh represents the hyperbolic tangent activation function.

The fourth is Bidirectional Long Short-Term Memory (Bi-LSTM) is an extension of the LSTM architecture for time-series forecasting. As the name suggests, Bi-LSTMs involve processing the input sequence in both forward and backward directions [58]. In a standard LSTM, the output at each time step is a function of the current input and the hidden state from the previous time step. In contrast, in a Bi-LSTM, the output at each time step is a function of the current input and the hidden states from both the forward and backward directions. This allows the network to capture dependencies not only from the past, but also from the future. The key advantage of Bi-LSTMs is their ability to capture both past and future dependencies in the data, which can be especially useful for time-series forecasting tasks where future information may be useful for predicting the next value in the sequence [59]. In the context of time-series forecasting, Bi-LSTMs can be used to predict the next value in a time series based on the previous values in both the forward and backward directions. The network is trained using a supervised learning algorithm, such as backpropagation through time, where the loss is minimized between the predicted and actual values of the target variable. (20) to (23).

$$H_{f(t)} = LSTM_f(X)t)) \tag{20}$$

$$H_{b(t)} = LSTM_b(X(t)) \tag{21}$$

$$H(t) = [H_f(t); H_b(t)] \tag{22}$$

$$Y(t) = g(W(hy) * H(t) + b(y)) \tag{23}$$

In the case of MIMO, the results are the dependent variables $Y(t)1$ and $Y(t)1$ as in (24) and (25), respectively.

$$Y(t)1 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)2 \tag{24}$$

$$Y(t)2 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)1 \tag{25}$$

where $H_f(t)$ is forward LSTM, $H_b(t)$ is backward LSTM, and H(t) is concatenation of forward and backward LSTM.

The last is Gated Recurrent Unit (GRU). GRUs were designed to address the limitations of traditional RNNs, such as difficulty in capturing long-term dependencies and vanishing gradients [60]. GRUs are similar to LSTMs in that they use a memory cell to remember information for long periods of time. However, unlike LSTMs, GRUs use only two gates: the reset and update gates. The reset gate determines how much of the previous memory to forget, while the update gate determines how much of the current input to remember. GRUs can be used to predict the next value in a time series based on the previous values. The network is trained using a supervised learning algorithm, such as backpropagation through time, where the loss is minimized between the predicted and actual values of the target variable. One advantage of GRUs for time-series forecasting is their ability to capture long-term dependencies in the data while requiring fewer parameters than LSTMs [61]. Additionally, GRUs can handle variable-length sequences, making them useful for forecasting time-series data with irregular time intervals.

$$R(t) = \sigma(W(xr) * X(t) + W(hr) * H(t-1) + b(r)) \tag{26}$$

$$Z(t) = \sigma(W(xz) * X(t) + W(hz) * H(t-1) + b(z)) \tag{27}$$

$$H(t) = \big(1 - Z(t)\big) * H(t-1) + Z(t) * \tanh(W(xh) * X(t) + R(t) * \big(W(hh) * H(t-1) + b(h)\big) \tag{28}$$

$$Y(t) = g(W(hy) * H(t) + b(y)) \tag{29}$$

Both Y(t)1 and Y(t)2 are dependent on time in the case of MIMO, as shown in (30) and (31).

$$Y(t)1 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)2 \tag{30}$$

$$Y(t)2 = g\big(W(hy) * H(t) + b(y)\big) + Y(t)1 \tag{31}$$

where R(t) is reset gate and Z(t) is update gate.

These equations capture the MIMO deep learning models' forward pass for time-series forecasting. The models are trained by optimizing the weights and biases to minimize the forecasting error through backpropagation and gradient descent techniques.

According to the background, this paper establishes the method to create an effective deep learning-based for fundamental trading in Bitcoin stock price. Six forecasting methods, including MLP, CNN, RNN, LSTM, Bi-LSTM, and GRU were selected and analyzed to determine the best method based on accuracy in forecasting the future price. Hyperparameter tuning is used to determine the parameter setting values of various existing methods [62]. The hyperparameter tuning method used is random search. The setting parameter for all method can be seen in Table 2.

Table 2. Setting Parameter

Method	Hidden Layes	Neuron	Activation Function	Dropout	Optimizer	Loss	Epoch	Batch Size
MLP	2	32	Sigmoid	0.2	Adam	MSE	100	64
CNN	2	32	ReLU	0.2	Adam	MSE	100	64
RNN	2	32	Tanh	0.2	Adam	MSE	100	64
LSTM	2	32	Tanh	0.2	Adam	MSE	100	64
Bi-LSTM	2	32	Tanh	0.2	Adam	MSE	100	64
GRU	2	32	Tanh	0.2	Adam	MSE	100	64

Table 2 hyperparameter choices were chosen after thoroughly assessing their importance and projected influence on model performance. The following is a complete explanation of these parameters [63]. Hidden layers determine neural network depth and complexity. Although overfitting may grow, the model can capture more intricate interactions with hidden layers. The model's ability to learn and represent complicated patterns depends on hidden layer neuron counts. More neurons can grasp more complex data correlations, although overfitting may rise. The activation function gives the neural network non-linearity to describe complicated input-output interactions. Non-saturation, smoothness, and gradient propagation differ among activation functions. The batch size determines the number of samples processed before updating the model's weights during training. A smaller batch size updates weights more frequently but may be noisy, whereas a bigger batch size may take more memory but update more smoothly. Training epochs determine how many times the model processes the dataset. If not regularised, adding epochs lets the model learn more from the data but risks overfitting. Each of these hyperparameters plays a crucial role in determining the model's performance and behavior.

2.4. Evaluation

The evaluation process uses several forecasting methods to determine the performance of the Bitcoin stock price prediction. The evaluation performance uses Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). MAPE is used to show errors that can represent accuracy. MAPE has several ranges of value-meaning categories, as shown in Table 3 [64]. RMSE is used to detect outliers in the designed system. The equation of MAPE and RMSE can be seen in (32) to (33).

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\frac{|xt-\widehat{xt}|}{xt} \text{ x } 100\% \tag{32}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(xt-\widehat{xt})^2} \tag{33}$$

Table 3. MAPE Category

Range MAPE	Description
< 10%	The ability of the forecasting model is very good
10% – 20%	The ability of the forecasting model is good
20% – 50%	The ability of the forecasting model is feasible
> 50%	Poor forecasting model capability

3. Results and Discussion

The performance comparison of different methods for MIMO time-series forecasting can be seen in Table 4. The MLP method achieved a relatively low MAPE and RMSE for both open and close prices, with a MAPE of 9.56258% for open prices and 9.77263% for close prices, and an RMSE of 0.04473 for open prices and 0.04093 for close prices. This suggests that the MLP method performed well in predicting the open and close prices of the time-series data. Overall, the MLP method performed well in predicting the open and close prices of the time series, suggesting that MLPs can be a suitable choice for time-series forecasting tasks.

Table 4. Performance Evaluation Result

Methods	Open		Close	
	MAPE (%)	RMSE	MAPE (%)	RMSE
MLP	9.56258	0.04473	9.77263	0.04093
CNN	10.61802	0.09547	10.75626	0.06094
RNN	9.14382	0.04004	8.68090	0.04764
LSTM	8.79643	0.05272	8.67726	0.05097
Bi-LSTM	9.62744	0.02013	9.63486	0.02419
GRU	9.20339	0.02520	8.73113	0.03806

The CNN method achieved a MAPE of 10.61802%, an RMSE of 0.09547 for the Open, and a MAPE of 10.75626% and an RMSE of 0.06094 for the Close, indicating that the CNN method performed better than the ARIMA model but outperformed the MLP, RNN, LSTM, and GRU methods. CNN may not perform well on time-series data with long-term dependencies, as the filters may not be able to capture the full range of temporal patterns in the data. However, CNN also requires more training data to produce good MAPE values. In this study, the 1475 data processed using CNN could still not produce optimal values. The performance of the CNN method suggests that it may be useful for some time-series forecasting tasks but may not be optimal for all applications.

Based on Table 4, the RNN method achieved a MAPE of 9.14382%, an RMSE of 0.04004 for the open price prediction, a MAPE of 8.68090%, and an RMSE of 0.04764 for the close price prediction, this indicates that the RNN method performed relatively well compared to the other methods regarding MAPE and RMSE values, especially for the open price prediction. A lower MAPE and RMSE values indicate that the RNN model was able to make more accurate predictions of the stock prices based on the previous values in the time series. It is important to note that the performance of the RNN method may vary depending on the specific data and problem being addressed. However, RNNs may have difficulty with vanishing gradients, where the gradients become very small, and the network cannot learn long-term dependencies.

The LSTM model achieved a MAPE of 8.79643% for the Open and 8.67726% for the Close. The corresponding RMSE values were 0.05272 for the Open and 0.05097 for the Close. LSTM shows the lowest MAPE for both opening and closing prices. A lower MAPE indicates better performance of the model. In this case, the LSTM model achieved a lower MAPE than all other models except for RNN, indicating its superior performance in predicting stock prices. Overall, the LSTM model demonstrated strong performance in forecasting stock prices based on historical data. Its ability to capture long-term dependencies in the time-series data and remember information for long periods likely contributed to its superior performance.

Table 4 shows the Bi-LSTM method achieved a MAPE of 9.62744%, an RMSE of 0.02013 for the Open, and a MAPE of 9.63486% and an RMSE of 0.02419 for the Close. The results suggest that Bi-LSTM performed well compared to the ARIMA method, which had the highest MAPE and RMSE values among all the methods. However, Bi-LSTM's performance was slightly lower than the RNN and

LSTM methods, which achieved lower MAPE and RMSE values. The Bi-LSTM shows the lowest RMSE for both open and close prices, indicating its ability to detect outliers in the result predictions. Bi-LSTM can capture a sequence's past and future context at each time step. Overall, the Bi-LSTM method showed promise for time-series forecasting tasks, but its performance may vary depending on the specific data.

The GRU model achieved a MAPE of 9.20339% for open prices and 8.73113% for close prices. Additionally, the model achieved an RMSE of 0.02520 for available prices and 0.03806 for close prices. The GRU model showed good performance in predicting both open and close prices, with a MAPE lower than ARIMA and Bi-LSTM for open prices and lower than ARIMA for close prices. The GRU model is known for capturing long-term dependencies in time-series data, while requiring fewer parameters than LSTMs. The results in the table suggest that the GRU model is effective for predicting stock prices and could be a useful tool for traders and investors. GRU can handle the vanishing gradient problem better than traditional RNNs.

MIMO and ensemble are two different approaches in time-series forecasting. In MIMO, multiple time series are used as inputs to the model to predict the values of all the time series simultaneously. This can be useful in situations where multiple related variables influence each other. In this research, the overall attributes could be used to predict open and closed prices. On the other hand, an ensemble involves using multiple models to make predictions and then combining the results of those models to make a final prediction; this can be useful in uncertain situations about the best model to use or where different models have different strengths and weaknesses. For example, an ensemble could include a combination MLP, LSTM, and CNN models, with the final prediction being a weighted average of the predictions made by each model [31][45][65].

Overall, the result provides insights into the performance of different DL methods for time-series forecasting and highlights the trade-offs between accuracy and complexity of the methods. The MAPE values of all methods at open and close fall into the category of the ability of the forecasting model is very good (<10%). Depending on the specific problem and methods characteristics, one may choose an appropriate method that balances the trade-offs and meets the desired performance criteria.

The results obtained in this study have significant practical implications, particularly in the context of stock trading and investment. Accurate stock price predictions can greatly benefit investors, traders, and financial institutions by providing valuable insights for decision-making. By leveraging MIMO time-series forecasting models, market participants can gain a competitive edge by identifying potential price movements, trends, and patterns in stock markets, this can help optimize trading strategies, improve risk management, and enhance portfolio performance. Moreover, the ability to accurately forecast stock prices can aid in identifying opportunities for arbitrage, hedging, and market timing, leading to increased profitability and reduced financial risks. Additionally, the performance of different methods in the study may have revealed specific patterns or trends. For example, certain deep learning models with specific architectures or hyperparameters might have better captured complex market dynamics, long-term dependencies, or nonlinear relationships. Identifying such patterns can guide practitioners in selecting appropriate models and techniques for stock price prediction tasks.

In the context of real-time or large-scale applications, it is essential to evaluate the efficiency of the models in processing and forecasting stock price data, includes assessing their ability to handle high-dimensional data, adapt to evolving market conditions, and provide timely predictions. Discussing the computational aspects of the models can help stakeholders assess the trade-offs between accuracy and computational efficiency, enabling them to make informed decisions when selecting models for real-world deployment. Future research should consider evaluating the performance of MIMO time-series forecasting models on multiple datasets from different stocks, various market conditions, and diverse periods, this will help establish the robustness and reliability of the proposed methods and provide a more comprehensive understanding of their performance across different contexts. Ultimately, choosing a suitable method for time-series forecasting tasks depends on the specific problem, the desired performance criteria, and the trade-offs between accuracy and complexity. This study provides valuable insights into the performance of different deep learning methods for time-series forecasting and can help guide the selection of appropriate methods for various applications.

4. Conclusion

In conclusion, this study compared the performance of various deep learning methods, including MLP, CNN, RNN, LSTM, Bi-LSTM, and GRU, in predicting open and close prices for time-series data. Overall, the LSTM method demonstrated the best performance in terms of MAPE, indicating its superior ability to predict stock prices. The Bi-LSTM method achieved the lowest RMSE values, highlighting its effectiveness in detecting prediction outliers. The GRU model also performed well and is known for handling the vanishing gradient problem better than traditional RNNs. While the CNN method performed better than the ARIMA model, it was outperformed by other deep learning methods, suggesting that it may not be optimal for all time-series forecasting tasks. The MIMO and ensemble approaches provide alternative ways to improve forecasting performance by leveraging multiple time series or combining the strengths of different models. The study focused on evaluating deep learning models for MIMO time-series forecasting, but a comprehensive benchmarking and comparative analysis with conventional approaches may have been limited. Future research should evaluate a more extensive range of conventional forecasting methodologies to understand their strengths and drawbacks better. Experimentation is recommended to confirm findings and test these models on different data sets and periods.

Declarations

Author contribution. The contribution or credit of the author must be stated in this section.
Funding statement. The unding agency should be written in full, followed by the grant number in square brackets and year.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning : A systematic literature review: 2005–2019," *Appl. Soft Comput.*, vol. 90, p. 106181, May 2020, doi: 10.1016/j.asoc.2020.106181.

[2] M. S. Gorus and M. Aydin, "The relationship between energy consumption, economic growth, and CO2 emission in MENA countries: Causality analysis in the frequency domain," *Energy*, vol. 168, pp. 815–822, Feb. 2019, doi: 10.1016/j.energy.2018.11.139.

[3] H. Lan, C. Zhang, Y.-Y. Hong, Y. He, and S. Wen, "Day-ahead spatiotemporal solar irradiation forecasting using frequency-based hybrid principal component analysis and neural network," *Appl. Energy*, vol. 247, pp. 389–402, Aug. 2019, doi: 10.1016/j.apenergy.2019.04.056.

[4] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, and B. Seaman, "Sales Demand Forecast in E-commerce Using a Long Short-Term Memory Neural Network Methodology," in *Lecture Notes in Computer Science*, 2019, pp. 462–474, doi: 10.1007/978-3-030-36718-3_39.

[5] T. Bikku, "Multi-layered deep learning perceptron approach for health risk prediction," *J. Big Data*, vol. 7, no. 1, p. 50, Dec. 2020, doi: 10.1186/s40537-020-00316-7.

[6] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 343–366, Feb. 2021, doi: 10.1007/s10044-020-00898-1.

[7] X. Yang, X. Liu, and Z. Li, "Multimodel Approach to Robust Identification of Multiple-Input Single-Output Nonlinear Time-Delay Systems," *IEEE Trans. Ind. Informatics*, vol. 16, no. 4, pp. 2413–2422, Apr. 2020, doi: 10.1109/TII.2019.2933030.

[8] C. Deng, Y. Huang, N. Hasan, and Y. Bao, “Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition,” *Inf. Sci. (Ny).*, vol. 607, pp. 297–321, Aug. 2022, doi: 10.1016/j.ins.2022.05.088.

[9] K. Li, G. Huang, S. Wang, B. Baetz, and W. Xu, “A Stepwise Clustered Hydrological Model for Addressing the Temporal Autocorrelation of Daily Streamflows in Irrigated Watersheds,” *Water Resour. Res.*, vol. 58, no. 2, pp. 1–31, Feb. 2022, doi: 10.1029/2021WR031065.

[10] F. Amato, F. Guignard, S. Robert, and M. Kanevski, “A novel framework for spatio-temporal prediction of environmental data using deep learning,” *Sci. Rep.*, vol. 10, no. 1, p. 22243, Dec. 2020, doi: 10.1038/s41598-020-79148-7.

[11] Z. Qu *et al.*, “Temperature forecasting of grain in storage: A multi-output and spatiotemporal approach based on deep learning,” *Comput. Electron. Agric.*, vol. 208, p. 107785, May 2023, doi: 10.1016/j.compag.2023.107785.

[12] R. Rakholia, Q. Le, B. Quoc Ho, K. Vu, and R. Simon Carbajo, “Multi-output machine learning model for regional air pollution forecasting in Ho Chi Minh City, Vietnam,” *Environ. Int.*, vol. 173, p. 107848, Mar. 2023, doi: 10.1016/j.envint.2023.107848.

[13] A. Thakkar and K. Chaudhari, “Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks,” *Appl. Soft Comput.*, vol. 96, p. 106684, Nov. 2020, doi: 10.1016/j.asoc.2020.106684.

[14] G. Ding and L. Qin, “Study on the prediction of stock price based on the associated network model of LSTM,” *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 6, pp. 1307–1317, Jun. 2020, doi: 10.1007/s13042-019-01041-1.

[15] J. Silva *et al.*, “An Early Warning Method for Agricultural Products Price Spike Based on Artificial Neural Networks Prediction,” in *Lecture Notes in Computer Science*, 2019, pp. 622–632, doi: 10.1007/978-3-030-22741-8_44.

[16] S. Mishra, C. Bordin, K. Taharaguchi, and I. Palu, “Comparison of deep learning models for multivariate prediction of time series wind power generation and temperature,” *Energy Reports*, vol. 6, pp. 273–286, Feb. 2020, doi: 10.1016/j.egyr.2019.11.009.

[17] C. V. Hudiyanti, J. L. Buliali, and A. Saikhu, “Modelling MIMO Transfer Functions for Analysis of The Relationship Between Temperature and Air Humidity with the Number of Confirmation, Suspect and Probable COVID-19 in Surabaya,” in *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*, Jun. 2021, pp. 246–251, doi: 10.1109/ICAICST53116.2021.9497836.

[18] H. Rodriguez, M. Medrano, L. M. Rosales, G. P. Penunuri, and J. J. Flores, “Multi-step forecasting strategies for wind speed time series,” in *2020 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Nov. 2020, pp. 1–6, doi: 10.1109/ROPEC50909.2020.9258743.

[19] P. Hewage *et al.*, “Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station,” *Soft Comput.*, vol. 24, no. 21, pp. 16453–16482, Nov. 2020, doi: 10.1007/s00500-020-04954-0.

[20] A. Alzaghir, A. R. Abdellah, and A. Koucheryavy, “Predicting Energy Consumption for UAV-Enabled MEC Using Machine Learning Algorithm,” in *Lecture Notes in Computer Science*, 2022, pp. 297–309, doi: 10.1007/978-3-030-97777-1_25.

[21] D. Saxena, I. Gupta, A. K. Singh, and C.-N. Lee, “A Fault Tolerant Elastic Resource Management Framework Toward High Availability of Cloud Services,” *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 3, pp. 3048–3061, Sep. 2022, doi: 10.1109/TNSM.2022.3170379.

[22] J.-J. Zhang and H.-S. Yan, “MTN optimal control of MIMO non-affine nonlinear time-varying discrete systems for tracking only by output feedback,” *J. Franklin Inst.*, vol. 356, no. 8, pp. 4304–4334, May 2019, doi: 10.1016/j.jfranklin.2019.03.008.

[23] T. Niu, J. Wang, H. Lu, and P. Du, “Uncertainty modeling for chaotic time series based on optimal multi-input multi-output architecture: Application to offshore wind speed,” *Energy Convers. Manag.*, vol. 156, pp. 597–617, Jan. 2018, doi: 10.1016/j.enconman.2017.11.071.

[24] P. Lara-Benítez, L. Gallego-Ledesma, M. Carranza-García, and J. M. Luna-Romera, "Evaluation of the Transformer Architecture for Univariate Time Series Forecasting," in *Lecture Notes in Computer Science*, 2021, pp. 106–115, doi: 10.1007/978-3-030-85713-4_11.

[25] C. Yin and Q. Dai, "A deep multivariate time series multistep forecasting network," *Appl. Intell.*, vol. 52, no. 8, pp. 8956–8974, Jun. 2022, doi: 10.1007/s10489-021-02899-x.

[26] Z. Wu, G. Luo, Z. Yang, Y. Guo, K. Li, and Y. Xue, "A comprehensive review on deep learning approaches in wind forecasting applications," *CAAI Trans. Intell. Technol.*, vol. 7, no. 2, pp. 129–143, Jun. 2022, doi: 10.1049/cit2.12076.

[27] H. Huang, Y. Wang, Y. Li, Y. Zhou, and Z. Zeng, "Debris-Flow Susceptibility Assessment in China: A Comparison between Traditional Statistical and Machine Learning Methods," *Remote Sens.*, vol. 14, no. 18, p. 4475, Sep. 2022, doi: 10.3390/rs14184475.

[28] A. Truchot *et al.*, "Machine learning does not outperform traditional statistical modelling for kidney allograft failure prediction," *Kidney Int.*, vol. 103, no. 5, pp. 936–948, May 2023, doi: 10.1016/j.kint.2022.12.011.

[29] M. Sousa, A. M. Tomé, and J. Moreira, "Long-term forecasting of hourly retail customer flow on intermittent time series with multiple seasonality," *Data Sci. Manag.*, vol. 5, no. 3, pp. 137–148, Sep. 2022, doi: 10.1016/j.dsm.2022.07.002.

[30] A. L. Schaffer, T. A. Dobbins, and S.-A. Pearson, "Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions," *BMC Med. Res. Methodol.*, vol. 21, no. 1, p. 58, Dec. 2021, doi: 10.1186/s12874-021-01235-8.

[31] H. Taud and J. F. Mas, "Multilayer Perceptron (MLP)," in *Lecture Notes in Geoinformation and Cartography*, 2018, pp. 451–455, doi: 10.1007/978-3-319-60801-3_27.

[32] B. Cai *et al.*, "Resilience evaluation methodology of engineering systems with dynamic-Bayesian-network-based degradation and maintenance," *Reliab. Eng. Syst. Saf.*, vol. 209, p. 107464, May 2021, doi: 10.1016/j.ress.2021.107464.

[33] R. Reichenberg, "Dynamic Bayesian Networks in Educational Measurement: Reviewing and Advancing the State of the Field," *Appl. Meas. Educ.*, vol. 31, no. 4, pp. 335–350, Oct. 2018, doi: 10.1080/08957347.2018.1495217.

[34] A. Safari and M. Davallou, "Oil price forecasting using a hybrid model," *Energy*, vol. 148, pp. 49–58, Apr. 2018, doi: 10.1016/j.energy.2018.01.007.

[35] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 24–49, Mar. 2021, doi: 10.1016/j.isprsjprs.2020.12.010.

[36] J. Lu, L. Tan, and H. Jiang, "Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification," *Agriculture*, vol. 11, no. 8, p. 707, Jul. 2021, doi: 10.3390/agriculture11080707.

[37] R. Chauhan, K. K. Ghanshala, and R. . Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Dec. 2018, pp. 278–282, doi: 10.1109/ICSCCC.2018.8703316.

[38] I. E. Livieris, N. Kiriakidou, S. Stavroyiannis, and P. Pintelas, "An Advanced CNN-LSTM Model for Cryptocurrency Forecasting," *Electronics*, vol. 10, no. 3, p. 287, Jan. 2021, doi: 10.3390/electronics10030287.

[39] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018, doi: 10.1016/j.ejor.2017.11.054.

[40] S. Ghimire, Z. M. Yaseen, A. A. Farooque, R. C. Deo, J. Zhang, and X. Tao, "Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks," *Sci. Rep.*, vol. 11, no. 1, p. 17497, Sep. 2021, doi: 10.1038/s41598-021-96751-4.

[41] N. Q. K. Le, E. K. Y. Yapp, and H.-Y. Yeh, "ET-GRU: using multi-layer gated recurrent units to identify electron transport proteins," *BMC Bioinformatics*, vol. 20, no. 1, p. 377, Dec. 2019, doi: 10.1186/s12859-019-2972-5.

[42] A.-N. Buturache and S. Stancu, "Solar Energy Production Forecast Using Standard Recurrent Neural Networks, Long Short-Term Memory, and Gated Recurrent Unit," *Eng. Econ.*, vol. 32, no. 4, pp. 313–324, Oct. 2021, doi: 10.5755/j01.ee.32.4.28459.

[43] H. V. Bitencourt, O. Orang, L. A. F. de Souza, P. C. L. Silva, and F. G. Guimarães, "An embedding-based non-stationary fuzzy time series method for multiple output high-dimensional multivariate time series forecasting in IoT applications," *Neural Comput. Appl.*, vol. 35, no. 13, pp. 9407–9420, May 2023, doi: 10.1007/s00521-022-08120-5.

[44] U. Kamath, J. Liu, and J. Whitaker, "Convolutional Neural Networks," in *Deep Learning for NLP and Speech Recognition*, Cham: Springer International Publishing, 2019, pp. 263–314, doi: 10.1007/978-3-030-14596-5_6.

[45] K. Sekaran, P. Chandana, N. M. Krishna, and S. Kadry, "Deep learning convolutional neural network (CNN) With Gaussian mixture model for predicting pancreatic cancer," *Multimed. Tools Appl.*, vol. 79, no. 15–16, pp. 10233–10247, Apr. 2020, doi: 10.1007/s11042-019-7419-5.

[46] S. Fan, N. Xiao, and S. Dong, "A novel model to predict significant wave height based on long short-term memory network," *Ocean Eng.*, vol. 205, p. 107298, Jun. 2020, doi: 10.1016/j.oceaneng.2020.107298.

[47] G. Kumar, S. Jain, and U. P. Singh, "Stock Market Forecasting Using Computational Intelligence: A Survey," *Arch. Comput. Methods Eng.*, vol. 28, no. 3, pp. 1069–1101, May 2021, doi: 10.1007/s11831-020-09413-5.

[48] S. Carta, A. Ferreira, A. S. Podda, D. Reforgiato Recupero, and A. Sanna, "Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting," *Expert Syst. Appl.*, vol. 164, p. 113820, Feb. 2021, doi: 10.1016/j.eswa.2020.113820.

[49] Y. Kumar, A. Koul, S. Kaur, and Y.-C. Hu, "Machine Learning and Deep Learning Based Time Series Prediction and Forecasting of Ten Nations' COVID-19 Pandemic," *SN Comput. Sci.*, vol. 4, no. 1, p. 91, Dec. 2022, doi: 10.1007/s42979-022-01493-3.

[50] A. P. Wibawa, I. T. Saputra, A. B. P. Utama, W. Lestari, and Z. N. Izdihar, "Long Short-Term Memory to Predict Unique Visitors of an Electronic Journal," in *2020 6th International Conference on Science in Information Technology (ICSITech)*, Oct. 2020, pp. 176–179, doi: 10.1109/ICSITech49800.2020.9392031.

[51] A. P. Wibawa, A. B. P. Utama, H. Elmunsyah, U. Pujianto, F. A. Dwiyanto, and L. Hernandez, "Time-series analysis with smoothed Convolutional Neural Network," *J. Big Data*, vol. 9, no. 1, p. 44, Dec. 2022, doi: 10.1186/s40537-022-00599-y.

[52] A. R. F. Dewandra, A. P. Wibawa, U. Pujianto, A. B. P. Utama, and A. Nafalski, "Journal Unique Visitors Forecasting Based on Multivariate Attributes Using CNN," *Int. J. Artif. Intell. Res.*, vol. 6, no. 2, pp. 1-8, Jul. 2022, doi: 10.29099/ijair.v6i1.274.

[53] P. Dhruv and S. Naskar, "Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review," in *Advances in Intelligent Systems and Computing*, 2020, pp. 367–381, doi: 10.1007/978-981-15-1884-3_34.

[54] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Phys. D Nonlinear Phenom.*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[55] A. W. Saputra, A. P. Wibawa, U. Pujianto, A. B. P. Utama, and A. Nafalski, "LSTM-based Multivariate Time-Series Analysis : A Case of Journal Visitors Forecasting," *Ilk. J. Ilm.*, vol. 14, no. 1, pp. 57–62, 2022, doi: 10.33096/ilkom.v14i1.1106.57-62.

[56] A. P. Wibawa, R. R. Ula, A. B. P. Utama, M. Y. Chuttur, A. Pranolo, and Haviluddin, "Forecasting e-Journal Unique Visitors using Smoothed Long Short-Term Memory," in *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Oct. 2021, pp. 609–613, doi: 10.1109/ICEEIE52663.2021.9616628.

[57] A. Pranolo, Y. Mao, A. P. Wibawa, A. B. P. Utama, and F. A. Dwiyanto, "Robust LSTM With Tuned-PSO and Bifold-Attention Mechanism for Analyzing Multivariate Time-Series," *IEEE Access*, vol. 10, pp. 78423–78434, 2022, doi: 10.1109/ACCESS.2022.3193643.

[58] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, "Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach," *Fractal Fract.*, vol. 7, no. 2, p. 203, Feb. 2023, doi: 10.3390/fractalfract7020203.

[59] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, "Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study," *Chaos, Solitons & Fractals*, vol. 140, p. 110121, Nov. 2020, doi: 10.1016/j.chaos.2020.110121.

[60] G. Yigit and M. F. Amasyali, "Simple But Effective GRU Variants," in *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug. 2021, pp. 1–6, doi: 10.1109/INISTA52262.2021.9548535.

[61] J. Zhao, H. Qu, J. Zhao, H. Dai, and D. Jiang, "Spatiotemporal graph convolutional recurrent networks for traffic matrix prediction," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 11, pp. 1-14, Nov. 2020, doi: 10.1002/ett.4056.

[62] A. B. P. Utama, A. P. Wibawa, Muladi, and A. Nafalski, "PSO based Hyperparameter tuning of CNN Multivariate Time-Series Analysis," *J. Online Inform.*, vol. 7, no. 2, pp. 193–202, 2022, doi: 10.15575/join.v7i2.858.

[63] Y. Mao, A. Pranolo, A. P. Wibawa, A. B. Putra Utama, F. A. Dwiyanto, and S. Saifullah, "Selection of Precise Long Short Term Memory (LSTM) Hyperparameters based on Particle Swarm Optimization," in *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, May 2022, pp. 1114–1121, doi: 10.1109/ICAAIC53929.2022.9792708.

[64] A. P. Wibawa, Z. N. Izdihar, A. B. P. Utama, L. Hernandez, and Haviluddin, "Min-Max Backpropagation Neural Network to Forecast e-Journal Visitors," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Apr. 2021, pp. 052–058, doi: 10.1109/ICAIIC51459.2021.9415197.

[65] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," *IEEE Access*, vol. 8, pp. 180544–180557, 2020, doi: 10.1109/ACCESS.2020.3028281.