ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Comparative Study on Stock Movement Prediction Using Hybrid Deep Learning Model

Neny Sulistianingsih[1] and Galih Hendro Martono[2]

ABSTRACT

Applying machine learning techniques in stock market prediction has evolved significantly, with deep learning methodologies gaining prominence. Conventional algorithms such as Linear Regression and Neural Networks initially dominated but struggled to capture complex temporal dependencies in financial data. Recent research has explored deep learning architectures like LSTM and CNN and hybrids such as CNN-LSTM and LSTM-CNN, showcasing promising results. However, there's a gap in research comparing these models across different datasets, particularly in predicting stock movements. This study addresses this gap by conducting a comparative analysis of deep learning and hybrid models for stock movement prediction in the Indonesian banking sector. The evaluation based on RMSE and MAE reveals that the LSTM-CNN hybrid consistently outperforms other models, showcasing its versatility and accuracy across different data characteristics. Then, exploration through hyperparameter tuning demonstrates the criticality of parameter selection in optimizing model performance. These findings contribute to advancing predictive modeling in financial markets, offering valuable insights for investors, analysts, and policymakers. Further research in hyperparameter tuning and model optimization holds promise for enhancing accuracy and reliability in stock price prediction.

Article information:

Keywords: Stock Market Prediction, Deep Learning, Hybrid Models, LSTM-CNN, Hyperparameter Tuning

Article history:
Received: April 3, 2024
Revised: August 15, 2024
Accepted: September 26, 2024
Published: October 5, 2024
(Online)

DOI: 10.37936/ecti-cit.2024184.256303

1. INTRODUCTION

Applying machine learning techniques in stock market prediction has significantly evolved, particularly with deep learning methodologies. Initially, conventional machine learning algorithms such as Linear Regression, Decision Trees, Support Vector Machines (SVM), and Neural Networks were predominantly utilized. These algorithms demonstrated promising results by capturing linear and non-linear relationships within financial data. For instance, [1]–[5] evaluated various machine learning algorithms, including linear regression, decision trees, SVM, and ensemble methods, highlighting their effectiveness in predicting stock prices. Few researchers specifically used machine learning, such as Random Forest [6] and vector machine [7]. However, despite their success, these conventional machine learning methods often struggled to capture complex temporal dependencies and patterns inherent in financial time series data.

In recent years, there has been a growing interest in leveraging deep learning architectures for stock market prediction, which can automatically learn hierarchical representations of data. Research conducted by [8], [9] used ANN [8], DNN and RNN models [4], [9], [10]. Research by [11] proposed a novel approach for stock market prediction based on Twitter sentiment extraction using a BiLSTM-Attention model, which outperformed traditional methods in terms of Root Mean Squared Error (RMSE). Similarly, [12] explored the use of real-time Twitter data and sentiment analysis (SA) in predicting stock market movements using machine learning techniques, underscoring the potential of social media data in enhancing prediction accuracy. Furthermore, [13] conducted a comparative analysis of various machine learning algorithms, including neural networks, SVM, Naive Bayes, CNN, and deep learning, shedding light on the effectiveness and drawbacks of these methods in forecasting stock

[1,2]The authors are with the Departement of Computer Science, Faculty of Engineering, Universitas Bumigora, Mataram, Indonesia 83127, E-mail: neny.sulistianingsih@universitasbumigora.ac.id and galih.hendro@universitasbumigora.ac.id
[1]Corresponding author: neny.sulistianingsih@universitasbumigora.ac.id

patterns.

Furthermore, Hybrid deep learning models, such as Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) [14]–[16] and Long Short-Term Memory-Convolutional Neural Network (LSTM-CNN) [17], [18], have emerged as powerful alternatives. These models combine the strengths of convolutional neural networks (CNNs) for feature extraction and long short-term memory (LSTM) networks for sequence modeling.

The application of deep learning and hybrid deep learning methodologies in various domains has demonstrated significant success in tasks such as fault [14] and phishing [17] detection, medical area [15], [16], and power flow [18] prediction. However, there is a noticeable gap in the literature concerning their application in stock movement prediction, particularly in financial markets. While there have been studies focusing on conventional machine learning techniques for stock prediction, such as linear regression and neural networks, there is a lack of comprehensive research comparing the efficacy of deep learning and hybrid deep learning approaches.

Furthermore, the existing literature primarily focuses on the performance evaluation of individual models, such as LSTM, CNN, and their hybrids, in specific domains. Studies that systematically compare these models' performance across different datasets and application scenarios, particularly in predicting stock movements, are scarce. Additionally, studies address these models' practical implementation and scalability, hindering their adoption in real-world financial applications.

This research seeks to bridge this gap by conducting a comparative study on stock movement prediction utilizing deep learning and hybrid deep learning approaches. Specifically, the research aims to achieve the following objectives:

1. Compare the performance of deep learning models, including LSTM, CNN, MLP, and their hybrids in predicting stock movements across multiple banking stocks in Indonesia.
2. The research searched for the optimal parameters for the best-performing hybrid method in prediction.
3. Identify the limitations and challenges associated with each model, such as overfitting, dataset size, and external factors, and explore opportunities for improvement and optimization. Additionally, it provides insights into the strengths and weaknesses of different deep learning and hybrid deep learning approaches for stock movement prediction, aiming to enhance understanding among researchers, practitioners, and stakeholders in the financial sector.

By fulfilling these objectives, this research aims to contribute to advancing predictive modeling in financial markets and provide valuable insights for investors, financial analysts, and policymakers.

The remaining part of this paper is organized, such as the related work (Chapter 2), which reviews prior studies, emphasizing their relevance to the identified gap. The proposed research methodology (Chapter 3) is outlined, which involves evaluating various deep learning models, including LSTM, CNN, and hybrids like CNN-LSTM and LSTM-CNN, against conventional machine learning techniques like MLP. Subsequently, Chapter 4 presents the results and discussions, analyzing the performance of the proposed models across multiple banking stocks. Finally, Chapter 5 encapsulates the findings in conclusion, emphasizing the significance of the research in advancing predictive modeling in financial markets and providing actionable insights for investors, financial analysts, and policymakers.

2. RELATED WORK

The literature review within the publication offers a comprehensive examination of the methodologies employed in stock market prediction using machine learning and sentiment analysis techniques. Research conducted by [19] explored the integration of machine learning and sentiment analysis, mainly through analyzing tweets related to stocks, identifying the ARIMA model as the most accurate for stock price prediction. This study underscores the historical evolution of stock markets and the transition to Intelligent Trading Systems driven by technological advancements. Building upon this, [11] proposed a model incorporating sentiment analysis from financial tweets for stock price prediction, utilizing a Bidirectional Long Short-Term Memory (BiLSTM) architecture. They emphasized integrating social media data into stock movement prediction, especially from platforms like Twitter. However, despite these advancements, there remains a gap in understanding these models' practical implementation considerations and scalability assessments.

In parallel, [1] introduced an AI-based stock market prediction system utilizing various machine learning algorithms, challenging the Efficient Market Hypothesis and highlighting the diversity of methodologies employed for stock market prediction. Similarly, [13] extended sentiment analysis to stock market-related lexicons with the StockSentiWordNet (SSWN) model, emphasizing the need for more robust prediction models to capture market dynamics. These studies collectively underline the growing importance of incorporating diverse data sources and advanced algorithms in enhancing predictive accuracy. However, gaps persist in addressing the ethical implications and unintended consequences of deploying such models in financial markets.

Transitioning to deep learning techniques, [16] focused on corona fault detection in metal-clad switchgear, highlighting the need for preventing elec-

Table 1: *State-of-the-art research using Hybrid Deep Learning.*

No.	Publication	Data	Method	Result	Limitation
1	[16]	Time series data from switchgear sensors	1D-CNN, LSTM, 1D-CNN-LSTM hybrid models	High accuracy in corona fault detection	Lack of discussion on real-world implementation challenges
2	[14]	Time series data from electrical machine sensors	Hybrid CNN-LSTM attention-based model	Superior performance over traditional models	Limited discussion on scalability and practical implementation
3	[15]	MIT-Covid-19 dataset	CNN-LSTM approach	Outperformed other techniques	Limited discussion on real-world deployment challenges
4	[17]	Phishing website data	CNN, LSTM, LSTM-CNN-based approach	High accuracy in phishing detection	Lack of evaluation on model generalization
5	[18]	Power flow measurement data, weather data	CNN-LSTM, LSTM-CNN hybrid models	Achieved minor errors under certain conditions	Challenges in finding optimal model configuration
6	Proposed Work	Stock movement data from Indonesia banks	CNN-LSTM, LSTM-CNN, CNN-MLP and LSTM-MLP	-	-

trical stress buildup with the hybrid model demonstrating superior accuracy. Research conducted by [14] shifted the focus to predicting electrical machine failures using a hybrid CNN-LSTM attention-based model, while [15] concentrated on COVID-19 detection using acoustic sound data with the CNN-LSTM approach. Research by [17] proposed a phishing detection system utilizing CNN, LSTM, and LSTM-CNN, showcasing the efficacy of deep learning across various domains. Lastly, [18] uses LSTM-CNN and CNN-LSTM to predict the high-voltage submet of Northeast Germany. The hybrid method has higher accuracy than single model algorithms, but it still has the challenge of finding the optimal configuration for the structure of the models.

However, despite the advancements in deep learning methodologies, gaps remain in practical implementation considerations, scalability assessments, and ethical implications, necessitating further research to address these aspects comprehensively. Hence, this comparative study aims to bridge these gaps by evaluating deep learning and hybrid deep learning approaches in stock movement prediction, thereby contributing to a deeper understanding of their effectiveness and applicability in financial markets. Table 1 shows state-of-the-art research using hybrid deep learning.

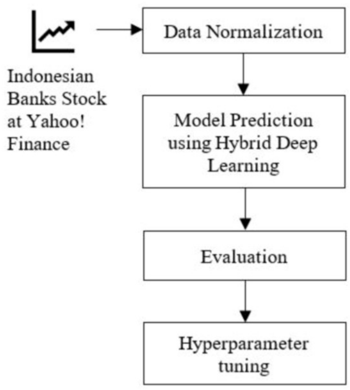

Fig.1: *Research Methodology.*

3. METHODOLOGY

The research process in this study consists of several stages, including data collection of stock data from multiple banks in Indonesia from Yahoo! Finance, data preprocessing, and predictive model development. The predictive models used in this study include various hybrid models of deep learning models such as CNN, LSTM, and Multi-Layer Perceptron (MLP). Additionally, these deep learning models will also serve as benchmarks. Evaluation is conducted on the deep learning and hybrid deep learning models. Finally, for the best-performing predictive model based on the evaluation results, a hyperparameter tuning process will be conducted to find the optimal parameters for each dataset. Figure 1 illustrates the research methodology employed in this study.

3.1 Data Collection

The research utilizes a dataset encompassing the stock movements of several banks in Indonesia sourced from Yahoo! Finance. It spans from January 1, 2015, to April 1, 2024. The dataset comprises various variables: Open, High, Low, Close, Adj Close, and Volume. These variables depict different aspects of stock performance. Open refers to the opening price of a stock on a particular day; High and Low represent the highest and lowest prices reached during the trading day, respectively. Close signifies the closing price of the stock for the day. Adj Close adjusts the closing price for factors such as dividends and stock splits, providing a more accurate reflection of the stock's value. Lastly, Volume denotes the total number of shares traded on a given day. These variables collectively offer a comprehensive view of the stock market dynamics, enabling researchers to analyze trends, volatility, and other pertinent aspects of stock behavior over the specified period.

3.2 Data Normalization

In data normalization, several steps are typically undertaken to prepare the data for modeling. Firstly, the dataset is often filtered to include only the relevant features or columns. In this research, the Close variable is selected, as it presumably represents the closing prices of the stocks. Then, the data undergoes conversion into a format commonly used for data manipulation and analysis. This transformation is integral to preparing the data for further processing and analysis tasks. Subsequently, the dataset is scaled using a technique called Min-Max scaling. It transforms the data to a specific range, usually between 0 and 1, making it suitable for various machine-learning algorithms. This scaling ensures that all features contribute equally to the analysis and prevents features with larger scales from dominating the model. The formula for Min-Max scaling is as follows [20].

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

In the Eq. 1 variable, X is the original value, while X_{min} is the minimum value of X in the dataset, and X_{max} is the maximum value of X in the dataset. After scaling, the data is split into training and testing sets. Finally, the input data for the model is reshaped into a 3D array, as many machine learning models, particularly neural networks, require input data to be in this format, with dimensions representing samples, time steps, and features, respectively. This preprocessing ensures the data is appropriately formatted and scaled to train the machine learning model effectively.

3.3 Hybrid Deep Learning Model

A. LSTM Model

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture type. LSTMs are particularly suited for sequential data modeling due to their ability to retain information over long periods and the highest prediction result compared to other models [4], [5].

LSTMs are designed to address the vanishing gradient problem in traditional RNNs [10]. It occurs when the gradients diminish as they propagate backward, challenging learning long-term dependencies. The critical innovation of LSTMs lies in their gating mechanisms, including input, forget, and output gates, which regulate the flow of information within the network. These gates selectively update and erase data from the cell state, enabling LSTMs to remember or forget information over time. Figure 2 shows the architecture of the LSTM model.

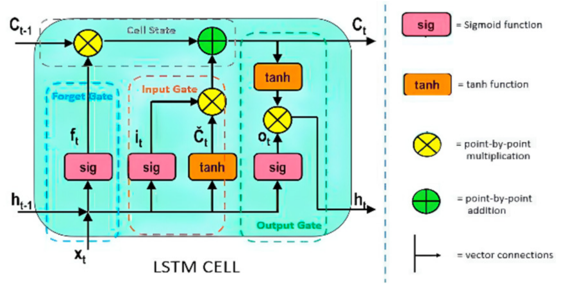

Fig.2: *LSTM Architecture [18], [21].*

The operations within an LSTM cell can be expressed as follows: [10], [16], [21].

$$f_t = \sigma_g(W_f.[h_{t-1}, x_t] + b_f \tag{2}$$
$$i_t = \sigma_g(W_i.[h_{t-1}, x_t] + b_i \tag{3}$$
$$\tilde{C}_t = \tanh(W_c.[h_{t-1}, x_t] + b_C \tag{4}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$
$$o_t = \sigma_g(W_o.[h_{t-1}, x_t] + b_o \tag{6}$$
$$h_t = o_t * \tanh(C_T) \tag{7}$$

Where f_t, i_t, and o_t are the forget, input, and output gate vectors, respectively, $\tilde{C}_t$ is the candidate cell state, C_t is the cell state, h_t is the hidden state/output, σ_g is the sigmoid activation function, W_f, W_i, W_C, W_O, are weight matrices, b_f, b_i, b_C, b_O are bias vectors and $[h_{t-1}, x_t]$ denotes the concatenation of the previous hidden state and the current input.

LSTMs are powerful tools for modeling sequential data like stock prices because they capture long-term dependencies and mitigate the vanishing gradient problem inherent in traditional RNNs. Through the intricate interplay of their gating mechanisms, LSTMs can effectively learn and leverage temporal patterns in the data, making them well-suited for time series forecasting tasks like stock price prediction.

B. CNN Model

Convolutional Neural Networks (CNNs) are renowned for their effectiveness in processing two-dimensional grid-based data like images and videos [18], [22], surpassing traditional neural networks in handling time-delayed data due to shared temporal weights, reducing computational complexity. The workflow involves steps such as fetching labeled training data, splitting it randomly into train and test sets, constructing the CNN architecture, incorporating max-pooling layers after each convolution to distill essential features, applying dropout regularization to prevent overfitting, and using a sigmoid function for classification [17]. CNNs are versatile, excelling in image analysis and learning abstract features from sequence data with multiple variables, making them suitable for various prediction tasks.

In a typical CNN model, layers like convolutional, pooling, flattening, and fully connected layers collectively serve crucial functions in processing input data. The convolutional layer extracts features through sliding windows and weight sharing, capturing spatial hierarchies of features. Following this, the pooling layer reduces dimensionality and selects salient features from the feature maps produced. The flattening layer transforms these multidimensional feature maps into one-dimensional vectors, maintaining spatial information for further processing. Finally, the fully connected layers facilitate inter-layer neuron connections, integrating extracted features to make final predictions or classifications. These layers collaborate to extract meaningful features, reduce dimensionality, and enable adequate information flow within the CNN model. Figure 3 shows the architecture of the CNN model.

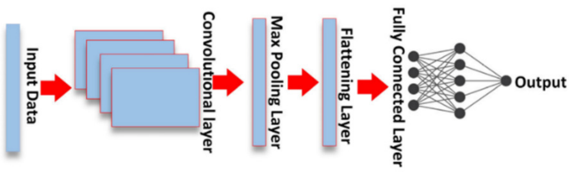

Fig.3: *CNN Architecture [18].*

C. CNN-LSTM Hybrid Model

The concept of a hybrid CNN-LSTM model involves combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to leverage the strengths of both architectures for sequential data analysis. In the provided code, the CNN layers are utilized initially to extract spatial features from the input data. The first Conv1D layer applies 1-dimensional convolution with specific filters and kernel size, followed by an activation function to capture essential patterns in the sequential data. Subsequently, another Conv1D layer with filters and a similar kernel size further refines the learned features. After the convolutional layers, an LSTM layer is introduced to capture the data's temporal dependencies and long-term patterns. The LSTM layer operates sequence-to-one, which generates output based on the entire sequence input. This layer enhances the model's ability to understand sequential dependencies and predict future values. Finally, a Dense layer with one unit is added for regression tasks, predicting a single output value. The model is compiled for training. During training, the model is fitted to the training data, allowing the model to learn the optimal parameters to minimize the loss function and make accurate predictions. Overall, the hybrid CNN-LSTM model integrates spatial and temporal information effectively, making it suitable for sequential data analysis tasks such as time series forecasting or natural language processing. The Pseudo code of the Hybrid CNN-LSTM model is seen below.

Pseudo code of Hybrid CNN-LSTM Model

Input : x_train, y_train, filters, kernel size, activation, input_shape
Output : Trained hybrid CNN-LSTM model
Process :

1. Initialize a sequential model.
2. Add a 1D convolutional layer with a specified number of filters, kernel size, activation function, and input shape.
3. Add another 1D convolutional layer with a specified number of filters and kernel size.
4. Add an LSTM layer with specified units and return_sequences set to False to return output only at the last step.
5. Add a dense layer with 1 unit for regression output.
6. Compile the model using the Adam optimizer and mean squared error loss function.
7. Train the model using input and target training data with specified batch size and number of epochs.

D. LSTM-CNN Hybrid Model

The research demonstrates the construction of a hybrid LSTM-CNN model. The hybrid LSTM-CNN model combines the strengths of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks for sequence data processing. LSTM layers are strategically positioned in this architecture before and after the convolutional layers. The LSTM layer preceding the convolutional layers enables the model to capture temporal dependencies within the input sequence data. Subsequently, the Convolutional layers extract spatial features from the temporal data representation produced by the LSTM layer. By integrating LSTM and CNN layers, the model can effectively learn temporal and spatial features from the input sequence data. Finally, a Dense layer is employed to perform the output prediction task. The model is compiled using the Adam optimizer and the mean squared error loss function. During training, the model is fitted to the training data with a specified batch size and number of epochs, allowing it to learn the underlying patterns and relationships within the input sequence data. Through this hybrid architecture, the model can leverage the complementary strengths of LSTM and CNN layers to achieve improved performance in tasks involving sequential data analysis and prediction—The Pseudo code of the hybrid LSTM-CNN model is seen below.

Pseudo code of Hybrid LSTM-CNN Model

Input : x_train, y_train
Output : Trained LSTM-CNN model
Process :

1. Initialize a Sequential model instance.
2. Add an LSTM layer, return sequences, and specify the input shape.
3. Use the activation function to add a 1D convolutional layer and a kernel size.
4. Add another 1D convolutional layer and kernel size using the same activation function.
5. Add another LSTM layer, but do not return sequences.
6. Add a Dense layer.
7. Compile the model
8. Train the model on the input training data (x_train) and target training data (y_train) with the designed epoch.

3.4 Evaluation

The evaluation metrics utilized in this study are Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to assess the performance model in this research. RMSE measures the average magnitude of the errors between predicted and actual values, providing insight into the model's accuracy [19]. It is calculated by taking the square root of the mean of the squared differences between predicted and actual values [1], [2]—the formula for RMSE is below [2], [8], [10].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (8)$$

Where n is the number of observations, y_i is the actual value, and $\hat{y}_i$ is the predicted value.

Conversely, MAE measures the average magnitude of the errors without considering their direction, providing a more robust indication of model performance. It is calculated by taking the mean of the absolute differences between predicted and actual values [23]. The formula for MAE is as bellow [10], [18].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (9)$$

A good model is characterized by low values of RMSE and MAE, indicating minimal error between predicted and actual values. Lower values of RMSE and MAE signify higher predictive accuracy and better model performance. Therefore, in this study, the model with the lowest RMSE and MAE values would be considered the most effective in accurately predicting the target variable.

3.5 Hyperparameter tuning

Hyperparameter tuning is crucial to optimizing machine learning models to find the best set of hyperparameters that maximize performance. Hyperparameter tuning aims to explore various combinations of hyperparameters to enhance the model's predictive capability. The hyperparameters considered for tuning include the number of filters in the CNN layers, kernel size, number of LSTM units, optimizer, and batch size.

A random search approach is employed to search for the optimal hyperparameters systematically. Random search randomly samples from the predefined hyperparameter space, consisting of different values for each hyperparameter. Each iteration randomly chooses a set of hyperparameters from the parameter space. Subsequently, the model is built using the selected hyperparameters, trained on the training data, and evaluated using a predefined evaluation metric, which in this case is MAE. MAE measures the average absolute difference between predicted and actual values, providing a quantitative measure of prediction accuracy.

The random search continues for a specified number of iterations, during which the model's performance is evaluated for each set of randomly selected hyperparameters. The process aims to identify the hyperparameters that yield the lowest MAE, indicative of the best model performance. Once the specified number of iterations is completed, the best-performing set of hyperparameters and the corresponding MAE value are reported. This approach enables efficient exploration of the hyperparameter space and facilitates the selection of optimal hyperparameters for the best model, ultimately enhancing its predictive accuracy.

4. RESULT AND DISCUSSION

Analyzing the stock data from eight Indonesian banks further demonstrates the relevance of applying predictive models like CNN, LSTM, and their hybrid combinations to forecast stock prices. The price trend analysis shows varied patterns across different banks. For instance, the adjusted closing price exhibits consistent upward movement for some banks, with minor fluctuations, as seen in the first few charts. In these cases, the Moving Averages (MA) of 10 and 50 days provide clear bullish signals, indicating steady market confidence in those banks. In some cases, there are observable crossovers between MA 10 and MA 50, signifying potential shifts in momentum either upwards or sideways. The less volatile stock prices correlate with banks with a relatively stable business environment and may exhibit more predictable growth patterns.

In contrast, certain banks show more volatility, as the later charts show. In these cases, the price fluctuations and the frequent crossovers between MA 10 and MA 50 days indicate more unstable market conditions. These stocks have experienced sharp declines and rapid recoveries, reflecting more complex economic factors affecting these banks. Such volatility would require more sophisticated models to capture the nuances of stock price movement, which might explain the efficacy of hybrid models like CNN-LSTM or LSTM-CNN in predicting such data. Figure 4 shows stock prices (Adjusted Close) and moving averages (MA) for the dataset used over different periods.

The data used in this analysis consists of stock data from 8 banks in Indonesia. For each company, 95% of the data is allocated for training, while the remaining 5% is used for testing. Although the percentage of testing data is relatively low, this decision was made to maximize the data available for training, considering the importance of training the model with a larger dataset to improve prediction accuracy. However, it is acknowledged that the smaller percentage of testing data might not be sufficient to evaluate the model's robust performance. Therefore, the model evaluation is conducted by calculating RMSE and MAE metrics to assess the performance of different prediction models. The models utilized include LSTM, CNN, and MLP, as well as hybrid combina-

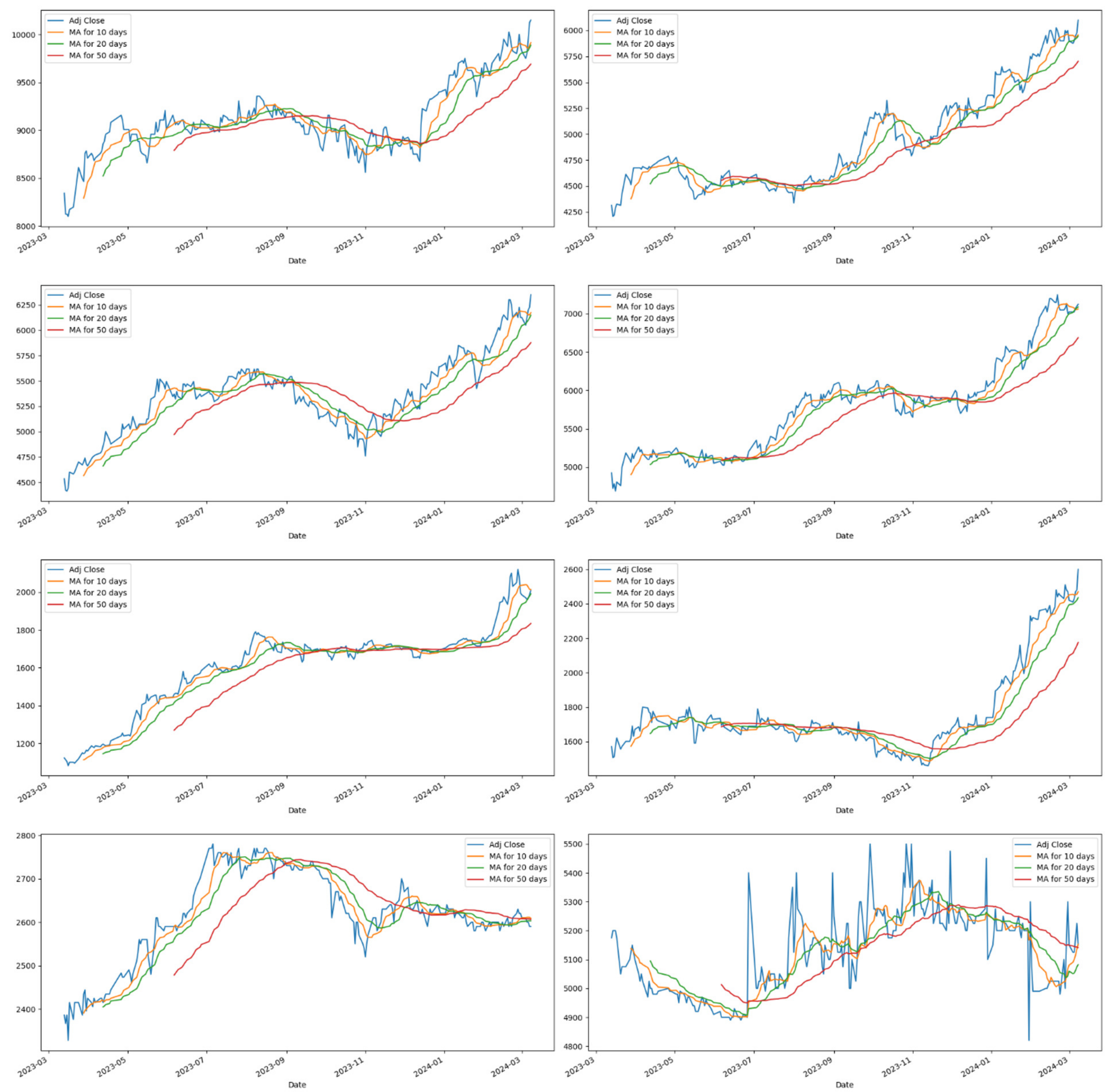

Fig.4: Stock Prices (Adjusted Close) along with Moving Averages (MA) over Different Time Periods.

tions such as CNN-LSTM and LSTM-CNN.

Additionally, other hybrid deep learning models like CNN-MLP and LSTM-MLP are employed as prediction models to analyze their impact on the prediction process and compare them with other hybrid models. The RMSE and MAE values are presented for each model configuration, including the standalone LSTM, CNN, and MLP models and the hybrid models that combine these different architectures. The layer structure of the model used in this research can be seen in Table 3.

This analysis aims to identify the model configuration that yields the lowest RMSE and MAE values, indicating superior predictive accuracy. Lower RMSE and MAE values reflect better model performance by demonstrating more minor errors between predicted and actual values. Consequently, the model with the lowest RMSE and MAE values is deemed most effective for accurately forecasting stock prices. The evaluation of various model configurations facilitates the selection of the optimal prediction model for each company, thereby improving decision-making and forecasting capabilities in financial markets.

Additionally, it is essential to note that the forecasting horizon of the model is designed for single-day predictions. The statement is evidenced by the fact that the RMSE and MAE metrics are computed based on daily stock data, highlighting that the model is intended for day-to-day forecasting rather than longer-term projections.

One of the outputs from the prediction model is the prediction result graph. An example of the prediction result output using the hybrid CNN-LSTM can be seen in Figure 5. The graph analysis illustrates predictions for stock prices over the past ten years (2015-2024) and forecasts for the next year, 2024. The blue line depicts Historical stock prices, representing the average stock price over the last decade. From 2015 to 2020, the stock price exhibited a steady upward trend. However, in 2020, there was a significant decline in stock prices. Subsequently, from 2021 to 2022, the stock prices resumed their upward trajectory. In 2023, there was a slight downturn in stock prices. The green line represents predictions for future stock prices, showing an upward trend over the next five years. By 2024, the predicted stock price is expected to reach USD 7,000. This analysis provides insights into the historical trends and future projections of stock prices, aiding investors in making informed decisions regarding their investments.

In Table 3, the columns represent different types of hybrid deep learning models used for stock prediction, including LSTM, CNN, MLP (Multi-layer Perceptron), CNN-LSTM, LSTM-CNN, CNN-MLP, and LSTM-MLP. Each cell in the table corresponds to the Root Mean Squared Error (RMSE) value obtained for the respective model in predicting the stock prices of various banks. In the analysis of experiments 1 through 8, represented by the number of lines (No.), different hybrid deep learning models were evaluated for stock price prediction, including LSTM, CNN, and MLP, and their hybrid combinations in various bank datasets.

In Table 4, similar to Table 3, the columns represent different hybrid deep learning models. At the same time, each cell contains the Mean Absolute Error (MAE) value obtained for the corresponding model in predicting the stock prices of various banks. The RMSE and MAE values presented in Tables 4 and 5 are calculated based on the testing period rather than the training period. The models were evaluated using the test dataset to provide a realistic assessment of their predictive performance on unseen data. The calculations for RMSE and MAE involved reversing the scaling on the predictions and actual test data, ensuring that the error metrics reflect the true differences between predicted and actual stock prices.

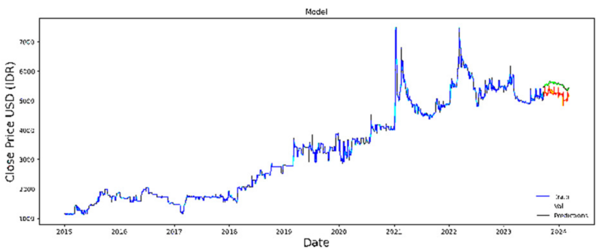

Fig.5: *Example of Hybrid CNN-LSTM Prediction Graph.*

The evaluation based on RMSE, detailed in Table 3, highlights the performance of various models across different banks. For instance, in experiment 1, characterized by high volatility, the LSTM-CNN model outperformed others by capturing intricate patterns effectively. The result demonstrates the model's adaptability to data complexities, which is particularly suited for banks facing similar market conditions. When considering the characteristics of the data for each bank, it becomes evident that specific models exhibit more consistent performance than others. For instance, banks such as scenarios 1, 2, and 3, which likely possess more complex and volatile stock data, consistently benefit from the LSTM-CNN hybrid model, showcasing its ability to effectively capture the intricate patterns inherent in such data sets.

Conversely, experiments 5 and 8, which may have relatively more predictable stock data, also demonstrate favorable outcomes with the LSTM-CNN hybrid model, indicating its versatility across different data characteristics. Notably, experiment 7, characterized by minimal variability in its data, shows consistent results across models, with the CNN model standing out in terms of RMSE. The result suggests simpler models suffice for banks with less volatile stock data. Overall, the analysis underscores the importance of tailoring model selection to the specific characteristics of each bank's data, ensuring optimal performance in stock prediction tasks.

Similarly, the evaluation based on MAE presented in Table 4 offers additional insights into model performance, particularly concerning the absolute errors in predictions across different banks. When considering the data characteristics of each bank, a similar pattern emerges, with the LSTM-CNN hybrid model consistently outperforming other models in minimizing absolute errors for banks such as experiments 1, 2, and 3. The result reaffirms the versatility and effectiveness of the LSTM-CNN architecture in handling diverse data sets with varying levels of complexity. Moreover, for banks like experiments 5 and 8, the LSTM-CNN hybrid model again stands out, indicating its ability to accurately predict stock prices while minimizing absolute errors, even in the case of relatively more predictable data. Conversely, simpler models like the CNN model may suffice for banks with minimal data variability, such as experiment 7, as reflected in their consistent performance across models. Overall, the analysis emphasizes the importance of considering the unique characteristics of each bank's data when selecting the most suitable model for stock prediction, ensuring optimal accuracy and reliability in real-world applications.

Furthermore, after the best hybrid model, LSTM-CNN was identified in the previous experiment, the next step to determine the best parameter using hyperparameter tuning was conducted. The provided results, as shown in Table 6, depict the performance of the LSTM-CNN hybrid model across a range of hyperparameter configurations, encompassing the number of filters, kernel size, LSTM units, optimizer, and batch size, evaluated through MAE and RMSE. Upon scrutiny and comparison, several vital insights emerge.

Based on the analysis of the performance evaluation results measured using RMSE and MAE, both metrics demonstrate good consistency in evaluating the performance of various hybrid deep learning models. In the RMSE table, the LSTM-CNN model consistently yields lower RMSE values than other models, especially in experiments 1 through 4 and 7. The results indicate that the models have stable predictive performance and can predict with minimal error. Conversely, models like CNN, MLP, and other hybrids such as CNN-LSTM, CNN-MLP, and LSTM-MLP tend to have high RMSE values (around 1.000) in almost all experiments. The result also suggests that these models are less capable of capturing data patterns as effectively as the LSTM-CNN hybrid model.

Table 2: *The Prediction Model's Layer Structure.*

Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5 (Output)
LSTM	LSTM (128 units, input)	LSTM (64 units)	Dense (25 units)	Dense (1 unit)	
CNN	Conv1D (64 filters, kernel=3, relu)	MaxPooling1D (pool_size=2)	Conv1D (32 filters, kernel=3, relu)	MaxPooling1D (pool_size=2)	Flatten
MLP	Dense (64 units, relu)	Dense (32 units, relu)	Dense (16 units, relu)	Dense (8 units, relu)	Dense (1 unit)
CNN-LSTM	Conv1D (64 filters, kernel=3, relu)	Conv1D (32 filters, kernel=3, relu)	LSTM (64 units)	Dense (1 unit)	
LSTM-CNN	LSTM (64 units, input)	Conv1D (64 filters, kernel=3, relu)	Conv1D (32 filters, kernel=3, relu)	LSTM (64 units)	Dense (1 unit)
CNN-MLP	Conv1D (64 filters, kernel=3, relu)	Conv1D (32 filters, kernel=3, relu)	Flatten	Dense (64 units, relu)	Dense (1 unit)
LSTM-MLP	LSTM (64 units, input)	Dense (32 units, relu)	Dense (1 unit)		

Table 3: *Performance evaluation based on RMSE using a Hybrid Deep Learning Model.*

No.	RMSE						
	LSTM	CNN	MLP	CNN-LSTM	LSTM-CNN	CNN-MLP	LSTM-MLP
1	0.004	1.000	1.000	1.000	0.002	0.002	0.002
2	0.002	1.000	1.000	1.000	0.001	0.001	0.001
3	0.002	1.000	1.000	1.000	0.001	0.001	0.001
4	0.001	1.000	1.000	1.000	0.000	0.001	0.000
5	0.023	1.000	1.000	1.000	0.023	0.018	0.011
6	0.008	1.000	1.000	1.000	0.018	0.023	0.011
7	0.005	1.000	1.000	1.000	0.003	0.011	0.003
8	0.000	1.000	1.000	1.000	0.010	0.045	0.003

Table 4: *Performance Evaluation based on MAE using Hybrid Deep Learning Model.*

No.	MAE						
	LSTM	CNN	MLP	CNN-LSTM	LSTM-CNN	CNN-MLP	LSTM-MLP
1	0.002	0.999	1.000	0.999	0.000	0.015	0.001
2	0.000	1.000	0.999	1.000	0.004	0.000	0.002
3	0.000	1.000	0.999	0.999	0.005	0.005	1.000
4	0.000	1.000	1.000	1.000	0.003	0.004	0.000
5	0.021	1.000	0.999	0.999	0.018	0.014	0.008
6	0.000	1.000	0.999	0.999	0.002	0.003	0.000
7	0.000	1.000	1.000	1.000	0.000	0.010	0.000
8	0.000	1.000	1.000	1.000	0.027	0.127	0.008

RMSE values vary in experiments 5, 6, and 8, where models such as CNN-MLP and LSTM-MLP show slightly lower RMSE values than CNN and MLP models but still do not perform as reliably as LSTM-CNN. The result reflects the possibility that the data characteristics used in these experiments may be more volatile or difficult to predict.

In the MAE table, a similar pattern is observed. The LSTM-CNN model again shows lower and consistent MAE values, particularly in experiments 1 through 4 and 7. The result confirms that the model has a reliable prediction capability with minimal error. Conversely, CNN, MLP, and other hybrid variants frequently produce high MAE values (around 1.000), indicating that these models have poor predictive performance.

The consistency of the LSTM-CNN model in producing low RMSE and MAE values suggests that both metrics are reliable tools for assessing the performance of predictive models, especially for more complex models like LSTM-CNN. However, simpler models like CNN and MLP, RMSE and MAE reveal their limitations in predicting stock price data, as reflected in consistently high error values. Overall, RMSE and MAE can be considered reliable metrics for identifying the best-performing model and highlighting the weaknesses of less complex models in predicting stock prices across various financial data sets.

Following identifying the optimal hybrid LSTM-CNN model in the prior analysis, the next phase involved hyperparameter tuning to pinpoint the most effective settings. The results in Table 6 illustrate the model's performance across various hyperparameter configurations, including the number of filters, kernel size, LSTM units, optimizer type, and batch size, evaluated using MAE and RMSE metrics.

The analysis reveals that the choice of hyperparameters, particularly the number of filters and kernel size, significantly impacts model performance. Notably, variations in MAE are observed with different kernel sizes while keeping the number of filters constant at 128, highlighting the model's sensitivity to these parameters. Additionally, the number of LSTM units proves to be a crucial determinant of performance. For instance, using 128 LSTM units in one experiment results in a higher MAE than a scenario with only 64 LSTM units, suggesting that excessive LSTM units may lead to overfitting rather than improved performance.

Furthermore, the selection of an optimizer plays a critical role in the outcomes. An RMSprop experiment yields substantially lower MAE than one using Adam despite similar LSTM units and filter configurations. The result indicates that optimizer choice is vital for fine-tuning model performance. Lastly, the impact of batch size on training and convergence is evident, with larger batch sizes (32) showing significantly lower MAE values than smaller batch sizes (16). The result suggests that a larger batch size can enhance convergence and overall model performance.

In conclusion, hyperparameter tuning plays a vital role in optimizing the performance of the LSTM-CNN hybrid model. Critical parameters such as the number of filters, kernel size, LSTM units, optimizer selection, and batch size greatly influence the model's accuracy. Adjusting the number of filters and kernel size leads to significant variations in MAE, while increasing LSTM units does not constantly improve performance and may cause overfitting. The RMSprop optimizer outperforms Adam in achieving lower MAE, and larger batch sizes improve convergence and overall performance. The LSTM-CNN architecture effectively combines temporal and spatial features, enhancing prediction accuracy by leveraging LSTM and CNN layers.

In contrast, the CNN-LSTM architecture processes data differently by initially applying CNN layers to extract spatial features before using LSTM layers to capture temporal dependencies. While this method can be helpful in some contexts, it might not fully harness the temporal dynamics from the start. This sequential approach could limit its ability to capture the sequential patterns significantly influencing stock price movements, potentially impacting its overall effectiveness in prediction tasks.

The analysis of stock price predictions highlights significant historical and future trends. Stock prices exhibited consistent growth from 2015 to 2020, fol-

Table 5: *Hyperparameter tuning for Hybrid LSTM-CNN Model.*

<table>
<tr><th>No</th><th>Number of Filters</th><th>Kernel Size</th><th>Number LSTM Units</th><th>Optimizer</th><th>Batch Size</th><th>MAE</th><th>RMSE</th></tr>
<tr><td>1</td><td>32</td><td>5</td><td>128</td><td>rmsprop</td><td>16</td><td>0.050</td><td>0.069</td></tr>
<tr><td>2</td><td>32</td><td>7</td><td>128</td><td>adam</td><td>32</td><td>0.046</td><td>0.052</td></tr>
<tr><td>3</td><td>32</td><td>3</td><td>128</td><td>rmsprop</td><td>32</td><td>0.053</td><td>0.065</td></tr>
<tr><td>4</td><td>64</td><td>3</td><td>128</td><td>adam</td><td>32</td><td>0.058</td><td>0.071</td></tr>
<tr><td>5</td><td>32</td><td>5</td><td>32</td><td>adam</td><td>32</td><td>0.003</td><td>0.003</td></tr>
<tr><td>6</td><td>64</td><td>3</td><td>32</td><td>rmsprop</td><td>32</td><td>0.037</td><td>0.044</td></tr>
<tr><td>7</td><td>64</td><td>3</td><td>32</td><td>adam</td><td>32</td><td>0.000</td><td>0.000</td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">64</td><td rowspan="2">7</td><td>32</td><td>rmsprop</td><td rowspan="2">32</td><td>0.014</td><td>0.018</td></tr>
<tr><td>128</td><td>adam</td><td>0.013</td><td>0.018</td></tr>
</table>

lowed by a sharp decline in 2020, likely due to the COVID-19 pandemic. The subsequent recovery through 2022 reflects market resilience. The forecast for 2024, predicting a rise to USD 7,000, suggests optimism about future market conditions and potential growth.

For investors, selecting an appropriate model is crucial. The LSTM-CNN hybrid model has proven effective for volatile and stable datasets, emphasizing the importance of choosing models tailored to specific data characteristics to improve prediction accuracy. While the model's focus on single-day forecasts supports short-term trading strategies, long-term investors should consider additional factors, such as broader economic conditions and market indicators, to make well-informed investment decisions. The LSTM-CNN model's superior performance, demonstrated through rigorous hyperparameter tuning and evaluation, highlights its ability to handle complex stock price data effectively, making it a valuable tool for short-term and long-term investment strategies.

5. CONCLUSION

From the evaluation based on RMSE and MAE, it is evident that the LSTM-CNN hybrid model consistently outperforms other models across various banks, demonstrating versatility in capturing complex patterns and providing accurate predictions. Tailoring model selection to the specific characteristics of each bank's data is crucial for optimal performance in stock prediction tasks.

Further research could focus on hyperparameter tuning to optimize the LSTM-CNN hybrid model's performance, considering parameters such as the number of filters, kernel size, LSTM units, optimizer, and batch size. By refining parameter configurations and leveraging insights from experimental results, practitioners can develop more robust and practical models for real-world applications in stock prediction. Continued research in hyperparameter tuning and model optimization holds promise for achieving enhanced accuracy and reliability, ensuring better stock price prediction outcomes.

AUTHOR CONTRIBUTIONS

Conceptualization, N.S. and G.H.M.; methodology, N.S.; software, N.S.; validation, N.S. and G.H.M.; formal analysis, N.S.; investigation, N.S.; data curation, N.S.; writing—original draft preparation, N.S.; writing—review and editing, N.S. and G.H.M.; visualization, N.S. and G.H.M.; supervision, G.H.M.; funding acquisition, G.H.M. All authors have read and agreed to the published version of the manuscript.

References

[1] M. Iyyappan, S. Ahmad, S. Jha, A. Alam, M. Yaseen and H. A. M. Abdeljaber, "A Novel AI-Based Stock Market Prediction Using Machine Learning Algorithm," *Scientific Programming*, vol. 2022, 2022.

[2] C. C. Emioma and S. O. Edeki, "Stock price prediction using machine learning on least-squares linear regression basis," *Journal of Physics: Conference Series*, vol. 1734, no. 1, pp. 1-9, 2021.

[3] A. Singh, P. Gupta and N. Thakur, "An empirical research and comprehensive analysis of stock market prediction using machine learning and deep learning techniques," *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, pp. 1-12, 2021.

[4] A. Subasi, F. Amir, K. Bagedo, A. Shams and A. Sarirete, "Stock Market Prediction Using Machine Learning," *Procedia Computer Science*, vol. 194, pp. 173–179, 2021.

[5] H. Kabir, A. Sobur and R. Amin, "Stock Price Prediction Using The Machine Learning," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 11, no. 7, pp. 946–950, 2023.

[6] M. Pinelis and D. Ruppert, "ScienceDirect," *The Journal of Finance and Data Science*, vol. 8, pp. 35–54, 2022.

[7] M. M. Akhtar, A. S. Zamani, S. Khan, A. S. A. Shatat, S. Dilshad and F. Samdani, "Stock

market prediction based on statistical data using machine learning algorithms," *Journal of King Saud University - Science*, vol. 34, no. 4, 2022.

[8] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock Closing Price Prediction using Machine Learning Techniques," *Procedia Computer Science*, vol. 167, pp. 599–606, 2020.

[9] P. Jaquart, S. Köpke and C. Weinhardt, "Machine learning for cryptocurrency market prediction and trading," *The Journal of Finance and Data Science*, vol. 8, pp. 331–352, 2022.

[10] U. Singh, D. Vasa, L. R. Mitta, K. Saurabh, R. Vyas and O. Vyas, "Use of Heterogeneous Data for Forecasting of Stock Market Movement using Social Media and Financial News Headlines," *ECTI-CIT Transactions*, vol. 17, no. 04, pp. 488–500, 2023.

[11] D. H. Fudholi, R. A. N. Nayoan and S. Rani, "Stock Prediction Based on Twitter Sentiment Extraction Using BiLSTM-Attention," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 10, no. 1, pp. 187–198, 2022.

[12] K. R. Baskaran and B. Kaviya, "Stock Market Prediction Using Machine Learning and Deep Learning Algorithms,"in*Sustainable Digital Technologies for Smart Cities Healthcare, Communication, and Transportation*, 1st Edition, Boca Raton, CRC press, 2023, pp. 127–138.

[13] S. Albahli, A. Irtaza, T. Nazir, A. Mehmood, A. Alkhalifah and W. Albattah, "A Machine Learning Method for Prediction of Stock Market Using Real-Time Twitter Data," *Electron*, vol. 11, no. 20, 2022.

[14] A. Borré, L. O. Seman, E. Camponogara, S. F. Stefenon, V. C. Mariani, and L. dos S. Coelho, "Machine Fault Detection Using a Hybrid CNN-LSTM Attention-Based Model," *Sensors*, vol. 23, no. 9, pp. 1–21, 2023.

[15] A. Rayan *et al.*, "Utilizing CNN-LSTM techniques for the enhancement of medical systems," *Alexandria Engineering Journal*, vol. 72, pp. 323–338, 2023.

[16] Y. A. Mohammed Alsumaidaee *et al.*, "Detection of Corona Faults in Switchgear by Using 1D-CNN, LSTM, and 1D-CNN-LSTM Methods," *Sensors*, vol. 23, no. 6, pp. 1–19, 2023.

[17] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," *Electron*, vol. 12, no. 1, pp. 1–18, 2023.

[18] F. Aksan, Y. Li, and V. Suresh, "CNN-LSTM vs LSTM-CNN to Predict Power Flow Direction: A Case Study of The High-Voltage Submet of Northeast Germany," *Sensors*, vol. 23, pp. 1–20, 2023.

[19] Y. Mehta, A. Malhar, and R. Shankarmani, "Stock Price Prediction using Machine Learning and Sentiment Analysis," in *2021 2nd International Conference for Emerging Technology (INCET)*, pp. 189–193, 2021.

[20] M. Awad and S. Fraihat, "Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for Machine Learning-Based Intrusion Detection Systems," Journal of Sensor Actuator Networks, vol. 12, no. 5, 2023.

[21] M. Bansal, A. Goyal and A. Choudhary, "Stock Market Prediction with High Accuracy using Machine Learning Techniques," *Procedia Computer Science*, vol. 215, pp. 247–265, 2022.

[22] S. Aslam, H. Herodotou, S. M. Mohsin, N. Javaid, N. Ashraf and S. Aslam, "A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids," *Renewable and Sustainable Energy Reviews*, vol. 144, no. 110992, 2021.

[23] D. Kumar, P. K. Sarangi, and R. Verma, "A systematic review of stock market prediction using machine learning and statistical techniques," *Materials Today Proceedings*, vol. 49, pp. 3187–3191, 2020.

Neny Sulistianingsih obtained a bachelor's degree in informatics engineering (S.Kom) and a master's degree in Informatics Engineering (M.Kom) from the Universitas Islam Indonesia, Yogyakarta. She has just completed his doctoral education at Universitas Gadjah Mada, Yogyakarta. Her research interests include data mining, sentiment analysis, big data, text mining, graph mining, and machine learning. She is currently a lecturer who teaches at Universitas Bumigora, Mataram. She can be contacted via email at neny.sulistianingsih@universitasbumigora.ac.id

Galih Hendro Martono obtained a bachelor's degree in informatics engineering (S.Kom) from the Universitas Islam Indonesia, Yogyakarta. While he obtained his master of engineering degree and doctoral education at Universitas Gadjah Mada, Yogyakarta. Her research interests include Computer Science, Big Data, Expert System, and Data Mining. He is currently a lecturer who teaches at Universitas Bumigora, Mataram. He can be contacted via email at galih.hendro@universitasbumigora.ac.id