International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

Hybrid ARIMA-ANN Model for Cryptocurrency Price Forecasting

Dian Kurniasari[1], Farrel Rizkiansyah[1], Notiragayu[1], Warsono[1]
[1]Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Lampung, Indonesia
Email address: dian.kurniasari@fmipa.unila.ac.id

Abstract— *Investments in digital currencies or other financial instruments during the COVID-19 pandemic were primarily driven by the pursuit of profits. Price forecasting for future periods using time series analysis is a key strategy in achieving this goal. ARIMA, a widely used method in time series analysis, is effective for modeling linear patterns but struggles with non-linear components often present in time series data. To address this limitation, this study introduces a Hybrid ARIMA-ANN approach, designed to capture both linear and non-linear patterns effectively. The proposed method involves two main models and evaluates performance under three data-splitting schemes: 60% training and 40% testing, 70% training and 30% testing, and 80% training and 20% testing. The ARIMA model is applied to forecast the primary data, while the ANN model predicts ARIMA residuals to refine overall accuracy. The optimal results were achieved using the 80%-20% data split, with the ANN model yielding an accuracy of 99.89%, an MSE of 0.0625, an RMSE of 0.2501, and a MAPE of 0.1032. These metrics indicate a highly precise model capable of reliable cryptocurrency price prediction. Based on this model, cryptocurrency prices are forecasted to decline from February 2022 to April 2023. This study demonstrates the potential of Hybrid ARIMA-ANN as a robust forecasting tool, particularly for applications where non-linear data dynamics are prominent. The findings contribute to advancing predictive analytics in financial markets.*

Keywords— *Hybrid ARIMA–ANN, Forecasting, Time series, Cryptocurrencies*.

I. INTRODUCTION

Cryptocurrency, a form of digital currency, facilitates online transactions and operates outside the framework of legal tender in Indonesia. Despite its unrecognized legal status, individuals can utilize cryptocurrencies at their own risk, assuming full responsibility for potential outcomes [1]. The cryptocurrency market presents significant profit opportunities for investors seeking capital gains. However, these opportunities are accompanied by substantial risks, primarily due to the inherent volatility and frequent price fluctuations characteristic of this market. Accurate forecasting tools or applications are thus essential for informed investment decision-making in the cryptocurrency sector [2].

Time series forecasting is a widely utilized technique for predicting future data points based on historical trends. Traditional time series methods, including Auto-regressive (AR), Moving Average (MA), Auto-regressive Integrated Moving Average (ARIMA), and Seasonal Auto-regressive Integrated Moving Average (SARIMA), rely on classical statistical models to achieve accurate results [3] [4]. While these models are known for their simplicity and flexibility, their performance diminishes when dealing with time series data characterized by non-linear components. This limitation arises from their inherent assumption of linearity, which often fails to capture the non-linear patterns present in the data and residuals, thereby reducing overall predictive accuracy [5].

Artificial Neural Networks (ANN) have demonstrated remarkable efficiency in addressing complex and non-linear problems, as well as in modeling univariate time series data with distinct patterns [6], [7]. The strengths of ANN include its ability to learn autonomously, its self-organizing functionality, and its exceptional computational speed [8]. These characteristics make ANN particularly suitable for tackling intricate problems that cannot be resolved using traditional mathematical approaches or conventional modeling techniques [9].

The ANN model is inspired by the structure and function of biological neural networks [10], mimicking the human brain's approach to data processing [11]. ANN is versatile and applicable across various domains, encompassing tasks such as regression, classification, prediction, and forecasting [12-14]. This study introduces a hybrid forecasting model that integrates classical statistical methods, represented by ARIMA, with machine learning techniques, specifically ANN. By combining these two approaches, the model effectively addresses both linear and non-linear patterns in datasets, resulting in more accurate and efficient forecasting outcomes compared to using either model independently [15]. Additionally, hyperparameter tuning is employed to optimize the model, ensuring enhanced performance and improved predictive accuracy.

Numerous studies have explored hybrid models similar to the one proposed in this research. For instance, Lopes et al. [16] utilized the ARIMA-ANN model to predict the spread of electromagnetic waves in densely forested urban areas, demonstrating its effectiveness as a forecasting tool. Similarly, Unnikkrishnan and Jothiprakash [17] introduced a hybrid model integrating Singular Spectrum Analysis (SSA), ARIMA, and ANN (SSA-ARIMA-ANN) to estimate daily rainfall in Koyna, Maharashtra, India. Statistical evaluations revealed that the SSA-ARIMA-ANN model achieved high accuracy in rainfall prediction.

In 2021, Nontapa et al. [15] integrated SARIMA and ANN models to analyze monthly electricity consumption data from Thai provinces. Their findings indicated that the SARIMA-ANN hybrid model demonstrated strong performance, achieving an accuracy rate of 93%. Priya et al. [18] employed a novel combination of ARIMA and ANN models for both linear and nonlinear modeling to enhance forecasting accuracy for the Covid-19 outbreak in India. Their results revealed that the hybrid model outperformed traditional methods, providing improved accuracy.

Chahal et al. [19] proposed a traffic congestion prediction model aimed at meeting the demands of an efficient and sustainable smart city using the Intelligent Transportation System (ITS). The model combines SARIMA for addressing linear components and Bi-LSTM for non-linear components. When compared to ARIMA and LSTM, the hybrid model demonstrated the lowest error rate of 0.03.

II. Material and Method

A. Data Description

The dataset employed consists of daily cryptocurrency prices sourced from the Kaggle Dataset Link, including five variables: Open (opening price), High (highest price), Low (lowest price), Close (closing price), and Volume (transaction volume). For the purpose of this study, only the Close variable is utilized, covering the period from November 2020 to January 2022, with a total of 457 data points. This selection aligns with the univariate nature of the ARIMA method.

TABLE I. Matic cryptocurrency data.

Date	Open	High	Low	Close	Volume
01/11/2020	0.107	0.115	0.101	0.109	496433632
02/11/2020	0.109	0.119	0.096	0.112	544802304
…					
…					
30/01/2020	0.112	0.134	0.112	0.122	613168000
31/01/2022	0.122	0.165	0.119	0.154	776462464

B. Data Preprocessing

Preprocessing represents a fundamental phase in data science and machine learning. The collected data undergoes preprocessing prior to being applied to machine learning algorithms or data mining techniques [20]. Three key preprocessing methods are commonly employed: data cleaning, data transformation, and data reduction [21].

Data cleaning addresses issues such as missing data, noise, and outliers by removing them. Data transformation involves processes like normalization, feature selection, and domain transformation. Data reduction focuses on dimensionality reduction and selecting relevant attribute subsets. While data transformation and reduction techniques are tailored to specific problems, data cleaning is essential for all machine learning tasks.

Furthermore, as the data used in this study is time series, it requires conversion into a supervised learning format, as shown in Equation (1).

$$y_t = f\left(y_{t-1}, y_{t-2}, \dots, y_{t-p}\right) \tag{1}$$

C. Autoregressive Integrated Moving Average (ARIMA)

The ARIMA model, developed by Box and Jenkins in the 1970s [22], combines Autoregressive (AR), Integration (I) or differencing, and Moving Average (MA) components [23]. Therefore, the ARIMA model (p, d, q) can be described as a combination of Autoregressive models with Moving Average (p, q) after differencing (d) [24].

Box-Jenkins models encompass both non-seasonal and seasonal variations. The ARIMA model, when seasonal and non-stationary, is represented by Equation (2) [5]:

$$y_t' = 1 + \alpha_1 y_{t-1}' + \alpha_2 y_{t-2}' + \cdots + \alpha_p y_{t-p}' + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \tag{2}$$

where,

y_t' represents the actual value, while e_t denotes the random error at time t. The coefficients α_1 and θ_1, with *p* and *q* as integers, are commonly referred to as the Autoregression and Moving Average coefficients, respectively [25].

D. Artificial Neural Network (ANN)

The ANN is an information processing system within the field of Artificial Intelligence (AI), designed to mimic the functionality of the human biological nervous system [26]. The architecture of an ANN comprises three primary layers: an input layer, a hidden layer, and an output layer [27].

1. Nodes at Hidden Layer

Determining the optimal number of nodes in the hidden layer is crucial for the success of ANN training [28], as it helps prevent both underfitting and overfitting. However, no definitive method exists for determining the ideal number of nodes without the need for training [29]. The most effective approach involves conducting hyperparameter tuning.

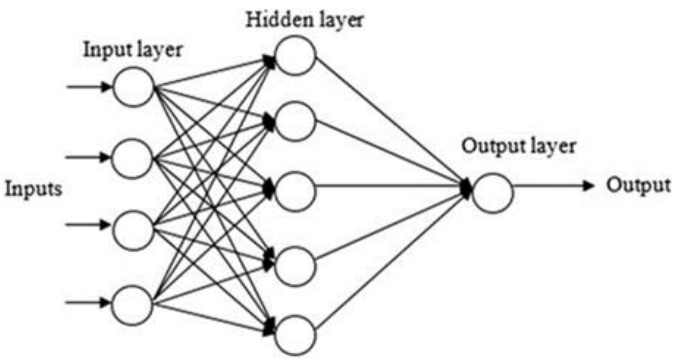


Fig. 1. ANN architecture.

2. Activation Function

The activation function is responsible for activating the input layer nodes connected to the hidden layer nodes, as well as the hidden layer nodes connected to the output layer nodes [30]. In practice, various types of activation functions are utilized, with their selection depending on the specific task and architecture employed [31]–[33].

3. Hyperparameter Tuning

Hyperparameter tuning plays a pivotal role in determining the optimal combination of input parameters [34]. In the context of ANN, these hyperparameters may include the number of layers, the number of nodes in each layer, dropout rates, batch size, and other related factors. A commonly employed technique for hyperparameter tuning is Grid Search [35], which systematically evaluates all possible combinations of parameters within a given configuration [36].

E. Hybrid ARIMA – ANN

1. Zhang's Method

Zhang [5] proposed the ARIMA-ANN hybrid model for time series forecasting. This model posits that time series data comprises both linear and non-linear components, as represented by Equation (3).

$$y_t = L_t + N_t \tag{3}$$

where:

L_t = linear component

N_t = non-linear component

This method combines ARIMA for forecasting linear components with ANN for capturing non-linear components, enhancing the overall forecasting performance by integrating both models. The resulting approach demonstrates superior forecasting accuracy compared to using ARIMA or ANN individually [37].

2. Khashei and Bijari Method

Khashei and Bijari [38] introduced an alternative ARIMA-ANN hybrid model, which shares similarities with Zhang's approach, assuming that time series data comprises both linear and non-linear components. In this model, ARIMA is employed to capture the linear components and generate forecasts, while the non-linear residuals are input into the ANN along with the original data. The key distinction between this model and Zhang's is that it does not assume an additive relationship between the linear and non-linear components. Instead, it establishes functional relationships between these components, as illustrated in Equation (4).

$$y_t = f(L_t, N_t) \tag{4}$$

where L_t is the linear component of the ARIMA prediction, and N_t is the non-linear component of the residual.

3. Babu and Reddy Method

Babu and Reddy [39] introduced an ARIMA-ANN hybrid method that integrates a moving average filter into the ANN-ARIMA model. Similar to other approaches, this model assumes that time series data comprises both linear and non-linear components. However, they argue that Zhang's and Khashei and Bijari's methods do not separate the original time series data into its linear and non-linear components. Instead, these methods use the linear ARIMA model to extract the linear components and treat the errors as non-linear. In contrast, Babu and Reddy's method first separates the linear and non-linear components and assigns them to the appropriate models. Finally, as in Zhang's approach, the decomposed components are fed into ARIMA and ANN, and the forecasting results are combined.

4. Proposed Method

The proposed model builds upon previous research where hybrid methods were employed to address both linear and non-linear components in time series data. ARIMA was utilized to generate residual values and predictions, which were subsequently modeled using ANN. Two distinct ANN models were applied: the first to forecast the linear component of the ARIMA predictions, and the second to predict the non-linear aspects of the ARIMA residuals. The hybrid approach combines the forecasting results of the linear and non-linear components through the use of ANN.

$$L_t = f\left(y_{(t-1)}, y_{(t-2)}, \dots, y_{(t-n)}\right) \tag{5}$$

$$N_t = f\left(e_{(t-1)}, e_{(t-2)}, \dots, e_{(t-n)} + \varepsilon_t\right) \tag{6}$$

$$y_t = L_t + N_t \tag{7}$$

L_t is a linear component of ARIMA model represented by an ANN over periods t-1 to t-n. N_t is a non-linear component of the ARIMA residual who also modeled using ANN across periods t-1 to t-n. The combined model incorporates both the linear and non-linear components at the t-th period.

F. Model Evaluation

Model evaluation involves calculating the difference between actual and predicted values [40]. The primary metric used in this study to assess model performance is accuracy. However, accuracy alone may not provide a complete assessment, as it can be misleading in the presence of imbalanced data. Consequently, additional evaluation metrics are necessary. As noted by Botchkarev [41], , commonly used metrics for this purpose include Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

a. Mean Absolute Percentage Error

The MAPE is a metric used for calculating the percentage difference between predicted and actual values.

b. Mean Squared Error dan Root Mean Squared Error

MSE and RMSE are metrics used to quantify the average discrepancy between predicted and actual values, with a greater emphasis on larger errors compared to smaller ones when assessing model performance. The optimal values for both MSE and RMSE are zero. The mathematical formulations for MAPE, MSE, and RMSE are as follows [42]:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - X_i}{Y_i}\right| \tag{8}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \tag{9}$$

$$RMSE = \sqrt{MSE} \tag{10}$$

where:

X_i = predicted i-th value

Y_i = i-th actual value

n = a number of experiments.

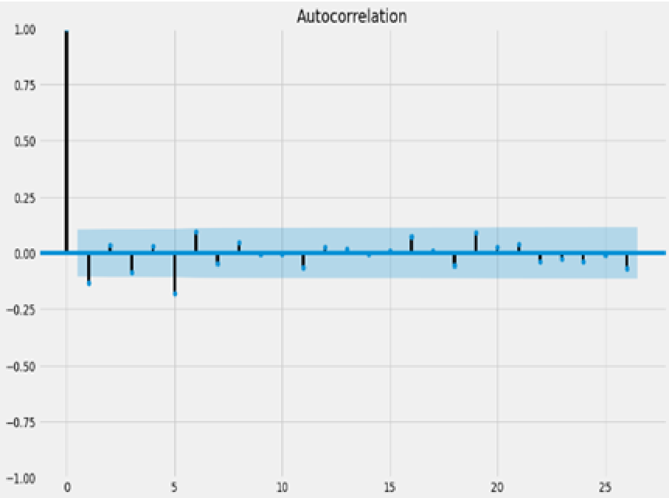
III. Results and Discussion

A. ARIMA Predictions

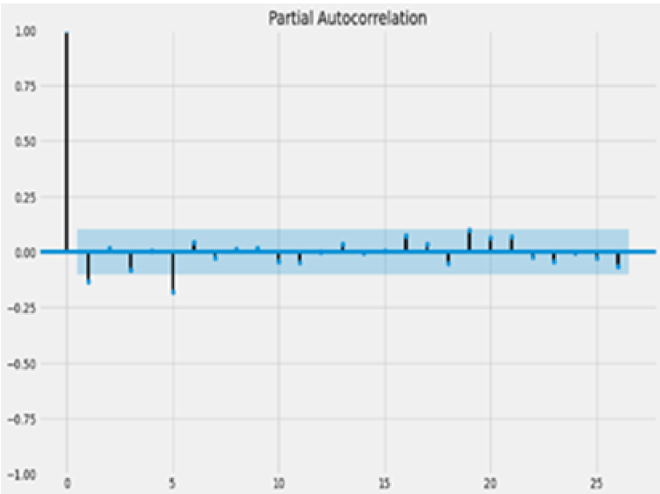
The linear component of the hybrid model is derived through forecasting with the ARIMA model, which involves parameters such as p, d, and q. The p parameter is determined from the Partial Autocorrelation Function (PACF) plot, while the d parameter reflects the number of differencing operations applied to the data. The q parameter is obtained from the Autocorrelation Function (ACF) plot. The value of d is established after testing the data for stationarity. If the data is not stationary, differencing is applied repeatedly until the data becomes stationary. This number of differencing operations,

Dian Kurniasari, Farrel Rizkiansyah, Notiragayu, and Warsono, "Hybrid ARIMA-ANN Model for Cryptocurrency Price Forecasting" *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 7, Issue 7, pp. 131-139, 2025.

denoted as n, is then used as the d parameter in the ARIMA model. The stationarity of the data was assessed using the Augmented Dickey-Fuller (ADF) test, which returned a p-value of 0.3443, indicating non-stationarity. After applying one differencing operation, the p-value decreased to 1.769^{-18}, confirming that the data became stationary.



(a)



(b)
Fig. 2. Results plot for (a) ACF, and (b) PACF.

Based on Figure 2, the ACF and PACF values at the first lag indicate that the ARIMA model's order for p and q is 1. As a result, the ARIMA prediction model for Matic cryptocurrency data is determined as ARIMA (1, 1, 1). Following the prediction, the next step involves reversing the data back to its original values through undifferentiation. The undifferentiated data is then compared with the actual data.

Residual data from the ARIMA model is extracted by computing the difference between the actual data and the predicted values. The result is presented in Figure 4.

Fig. 3. ARIMA prediction plot.

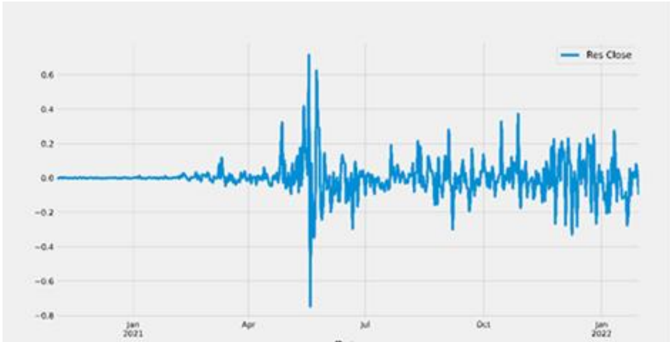


Fig. 4. ARIMA residuals.

Figures 3 and 4 demonstrate that the predicted data aligns closely with the actual data distribution, indicating that the ARIMA model is suitable for forecasting cryptocurrency prices. However, prior to applying the ARIMA model for forecasting, it is crucial to assess the model and determine its accuracy.

TABLE II. ARIMA model evaluation.

Evaluation Metrics	ARIMA Model
MSE	0.0119
RMSE	0.1089
MAP	0.0629
Accuracy	99.9371

Given that the ARIMA model demonstrates strong accuracy, forecasting was performed for the subsequent 90-day period.

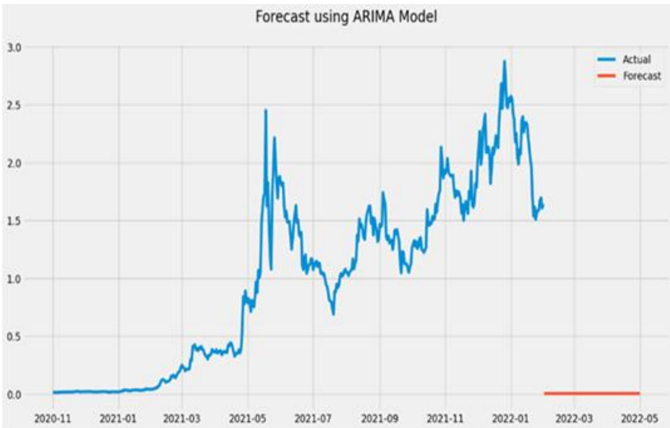


Fig. 5. Forecasting with ARIMA models.

Dian Kurniasari, Farrel Rizkiansyah, Notiragayu, and Warsono, "Hybrid ARIMA-ANN Model for Cryptocurrency Price Forecasting" *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 7, Issue 7, pp. 131-139, 2025.

International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

As noted by Zhang [5], Figure 5 demonstrates that while the ARIMA model exhibits a relatively high level of accuracy, its performance deteriorates over time due to the presence of non-linear components in the time series data, which do not align with the model's linear structure. Consequently, the forecasting results are suboptimal. In light of these findings, a Hybrid ARIMA-ANN model is proposed, which combines ARIMA predictions with residuals processed using an ANN.

B. ARIMA Predictions

The hybrid model integrates ARIMA prediction data with residual data processed by an ANN, combining two distinct models. The first model is an ANN designed to predict the linear component, representing the outcome of the ARIMA predictions. This model features a single hidden layer utilizing the ReLU activation function and a linear activation function at the output layer. The second model predicts the residual data from the ARIMA model, representing the non-linear component, with two hidden layers using the ReLU activation function and a Sigmoid function at the output layer. Hyperparameter tuning is performed to optimize both models, adjusting parameters such as the number of hidden layer nodes, dropout rate, and batch size. Data splitting is executed using various schemes, including 80% training and 20% testing, 70% training and 30% testing, and 60% training and 40% testing.

TABLE III. Hyperparameter tuning model 1.

Parameter	**60/40**	**70/30**	**80/20**
Nodes	64	32	64
Batch size	32	16	16
Dropout	0.3	0.2	0.2

TABLE IV. Hyperparameter tuning model 2.

Parameter	**60/40**	**70/30**	**80/20**
Nodes	64	16	32
Batch size	64	64	16
Dropout	0.1	0.3	0.3

Once the optimal parameters are determined, the next step involves evaluating the model using the loss function graphs presented in Figures 6-8. As shown in Figures 6, 7, and 8, the ANN model is constructed using three data splitting schemes: 60% training and 40% testing, 70% training and 30% testing, and 80% training and 20% testing. None of these schemes result in overfitting, indicating that the model is sufficiently robust for making predictions. However, the loss graph corresponding to the 80% training and 20% testing scheme demonstrates a better fit compared to the other data splitting schemes. Subsequently, the linear and non-linear components of the prediction data generated by the ANN model are combined to produce hybrid prediction data. Figures 9-11 provide a visualization of the hybrid prediction results for cryptocurrency price data.

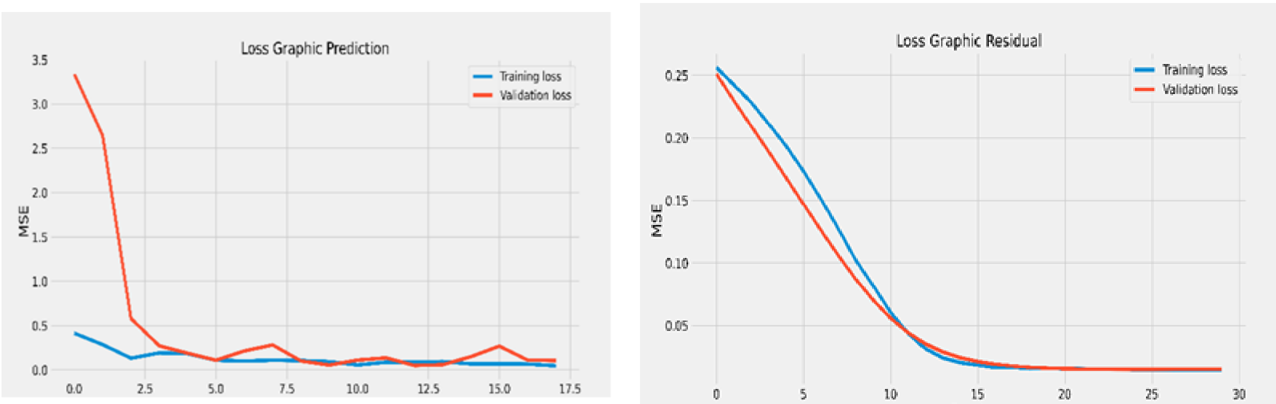


Fig. 6. Loss graphic for 60% train & 40% test scheme.

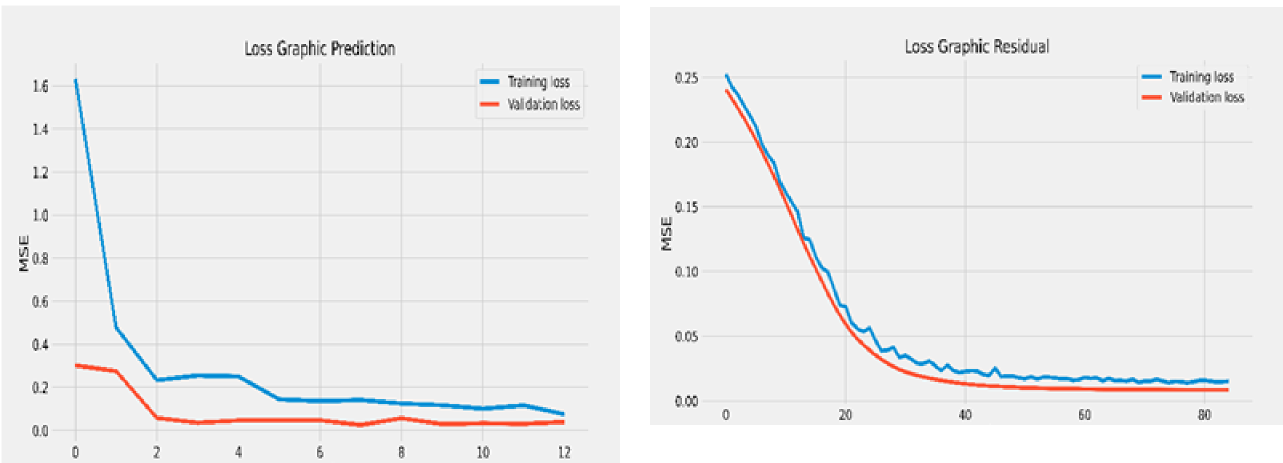


Fig. 7. Loss graphic for 70% train & 30% test scheme.

Dian Kurniasari, Farrel Rizkiansyah, Notiragayu, and Warsono, "Hybrid ARIMA-ANN Model for Cryptocurrency Price Forecasting" *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 7, Issue 7, pp. 131-139, 2025.

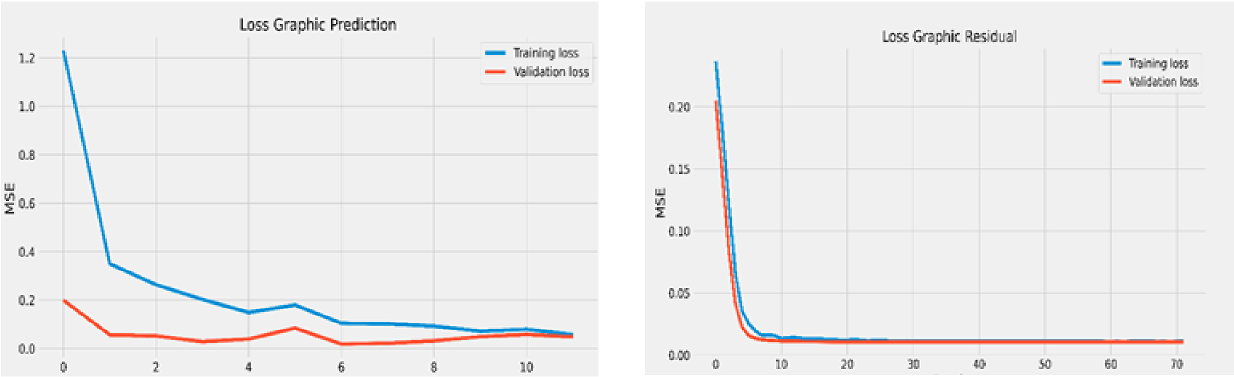


Fig. 8. Loss graphic for 80% train & 20% test scheme.

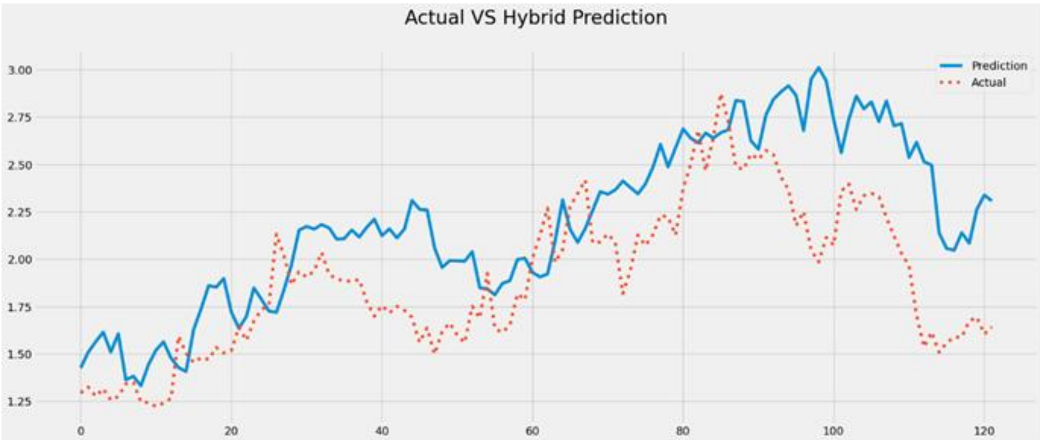


Fig. 9. Prediction plot for 60% train and 40% test scheme.

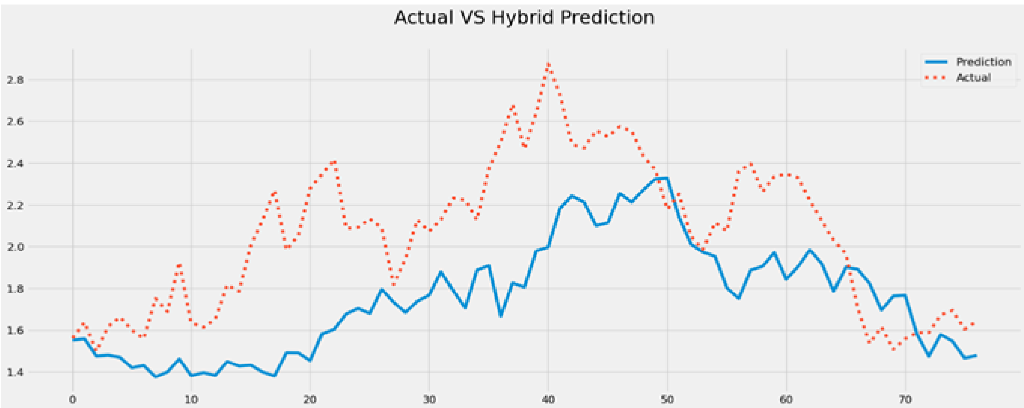


Fig. 10. Prediction plot for 70% train and 30% test scheme.

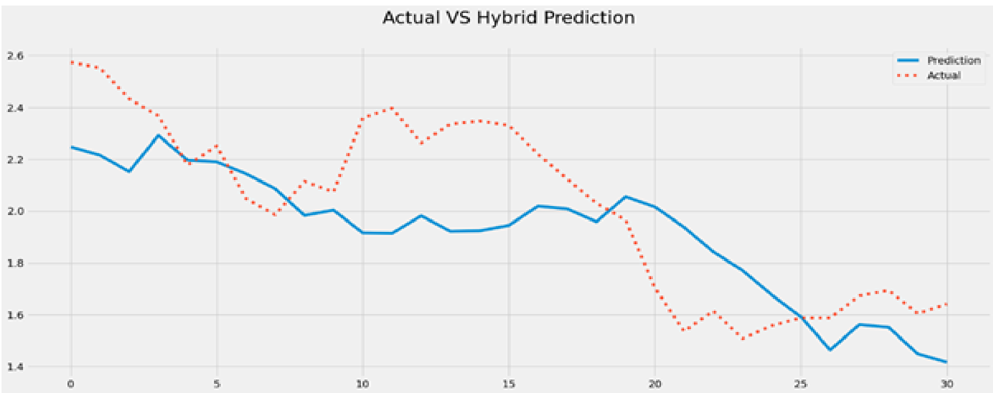


Fig. 11. Prediction plot for 80% train and 20% test scheme.

The prediction accuracy and feasibility of the hybrid forecasting model were assessed using RMSE, MSE, and MAPE as evaluation metrics, with the results presented in Table 5.

TABLE V. Hybrid model evaluation.

Evaluation Metrics	60/40	70/30	80/20
MSE	0.1637	0.1676	0.0625
RMSE	0.4046	0.4094	0.2501
MAP	0.1876	0.1573	0.1032
Accuracy	99.8124	99.8427	99.8968

The evaluation of the hybrid ARIMA-ANN model, as presented in Table 5, reveals notable variations in model performance across different data splits. The 80/20 data division yielded the lowest Mean Squared Error (MSE) of 0.0625, indicating superior model accuracy compared to the 60/40 and 70/30 splits, which showed MSE values of 0.1637 and 0.1676, respectively. Similarly, the Root Mean Squared Error (RMSE) was lowest for the 80/20 split (0.2501), signifying better predictive performance. In terms of Mean Absolute Percentage Error (MAPE), the 80/20 split also demonstrated the most favorable result of 0.1032, outperforming the 60/40 and 70/30 splits, which exhibited higher MAPE values. Furthermore, the accuracy of the model was highest at 99.8968% for the 80/20 split, reflecting the model's robust predictive capability under this data division. These findings suggest that the 80/20 data split provides the most effective balance for optimizing the hybrid ARIMA-ANN model's forecasting accuracy. The forecasting results for each model are illustrated in Figures 12-14.

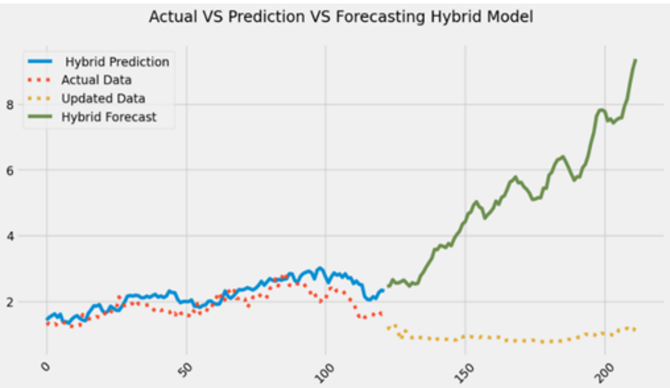


Fig. 12. Forecasting plot for 60% train & 40% test scheme.

Based on Figures 12-14, which depict the forecasting results for the different data splits, the 80/20 scheme stands out as the most effective model. In these figures, the actual data closely aligns with the predicted values of the 80/20 model, demonstrating its high forecasting accuracy. The visual comparison reveals minimal discrepancies between the observed and forecasted data, particularly in terms of trend consistency and peak alignment. This strong match underscores the model's ability to capture the underlying patterns and fluctuations of the data effectively. In contrast, the 60/40 and 70/30 schemes show more noticeable deviations between the actual and predicted values, particularly during periods of rapid change or volatility. These discrepancies suggest that the 80/20 data split better accommodates the model's capacity to generalize and forecast accurately across the entire dataset.

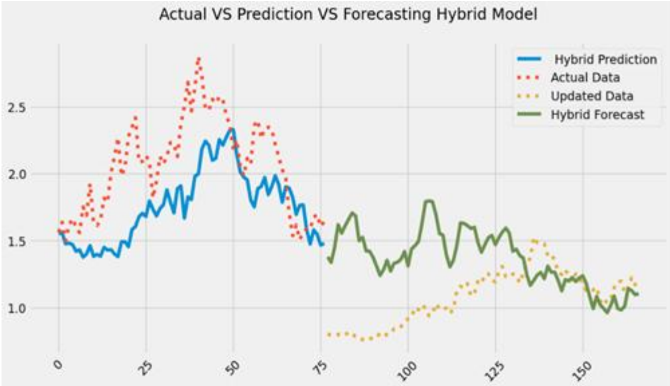


Fig. 13. Forecasting plot for 70% train & 30% test scheme.

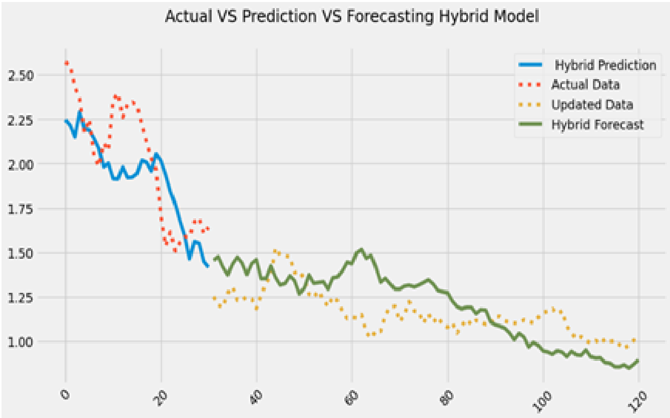


Fig. 14. Forecasting plot for 80% train & 20% test scheme.

IV. CONCLUSION

This study aims to find the best model for forecasting cryptocurrency prices. The data used is daily data on cryptocurrency prices on the Close variable obtained on the Kaggle website from November 2020 to January 2022, with a total of 457 data. The data is univariate, so we tried to apply the ARIMA model to forecasting. However, because the time series data contains non-linear components that do not match the characteristics of the linear model, the forecasting results are not satisfactory. Therefore, we propose the Hybrid ARIMA-ANN model to forecast the price of cryptocurrencies.

We built two ANN models by applying three different data splitting schemes, namely 60% train & 40% test, 70% train & 30% test, and 80% train & 20% test. The first model is used to forecast ARIMA predicted data, while the second is used to forecast ARIMA residual data. Hyperparameter tuning is performed to improve model performance on several parameters, including the number of nodes in the hidden layer, batch size, and drop out.

The model that obtained the best performance was the ANN model with a data sharing scheme of 80% train & 20% test. The optimal parameters for the first model include 64 nodes, 16 batch sizes, and 0.2 dropouts. Meanwhile, the second model has 32 nodes, 16 batch sizes, and 0.3 dropouts. The accuracy of the model reached 99.89%, MSE of 0.0625, RMSE of 0.2501, and MAPE of 0.1032, which shows that this model is good enough to do forecasting. As a result, the forecast for the price of cryptocurrencies from February 2022 to April 2023 has decreased.

REFERENCES

[1] R. C. Noorsanti, H. Yulianton, and K. Hadiono, "Blockchain - Teknologi Mata Uang Kripto (Crypto Currency)," *Pros. SENDI_U 2018*, no. July, pp. 978–979, 2018.

[2] I. Ulumuddin and A. Fadlil, "Bitcoin Price Prediction Using Long Short Term Memory (LSTM)," 2020. [Online]. Available: https://iocscience.org/ejournal/index.php/mantik

[3] S. M. Shakeel, N. S. Kumar, P. P. Madalli, R. Srinivasaiah, and D. R. Swamy, "Covid-19 prediction models: A systematic literature review," *Osong Public Health and Research Perspectives*, vol. 12, no. 4. pp. 215–229, 2021. doi: 10.24171/J.PHRP.2021.0100.

[4] K. H. Gusti, I. Irhamah, and H. Kuswanto, "Hybrid Double Seasonal ARIMA and Support Vector Regression in Short-Term Electricity Load Forecasting," *IPTEK J. Proc. Ser.*, vol. 0, no. 6, p. 322, 2021, doi: 10.12962/j23546026.y2020i6.11117.

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

[6] A. Y. Al-Bakri and M. Sazid, "Application of Artificial Neural Network (ANN) for Prediction and Optimization of Blast-Induced Impacts," *Mining*, vol. 1, no. 3. pp. 315–334, 2021. doi: 10.3390/mining1030020.

[7] L. Shan, Y. Liu, M. Tang, M. Yang, and X. Bai, "CNN-BiLSTM hybrid neural networks with attention mechanism for well log prediction," *J. Pet. Sci. Eng.*, vol. 205, no. December 2020, p. 108838, 2021, doi: 10.1016/j.petrol.2021.108838.

[8] D. A. Patel and K. N. Jha, "Neural Network Model for the Prediction of Safe Work Behavior in Construction Projects," *J. Constr. Eng. Manag.*, vol. 141, no. 1, 2015, doi: 10.1061/(asce)co.1943-7862.0000922.

[9] A. Leśniak and M. Juszczyk, "Prediction of site overhead costs with the use of artificial neural network based model," *Arch. Civ. Mech. Eng.*, vol. 18, no. 3, pp. 973–982, 2018, doi: 10.1016/j.acme.2018.01.014.

[10] S. Ali, "How do artificial neural networks lead to developing an optimization method?," *Trends Comput. Sci. Inf. Technol.*, pp. 067–069, 2020, doi: 10.17352/tcsit.000026.

[11] R. Dastres and M. Soori, "Artificial Neural Network Systems," *Int. J. Imaging Robot.*, vol. 21, no. 2, pp. 13–25, 2021, [Online]. Available: https://hal.science/hal-03349542%0Ahttps://hal.science/hal-03349542/document

[12] M. Madhiarasan and M. Louzazni, "Analysis of Artificial Neural Network: Architecture, Types, and Forecasting Applications," *J. Electr. Comput. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/5416722.

[13] F. Machado Cavalcanti, C. Emilia Kozonoe, K. André Pacheco, and R. Maria de Brito Alves, "Application of Artificial Neural Networks to Chemical and Process Engineering," in *Deep Learning Applications*, 2021. doi: 10.5772/intechopen.96641.

[14] E. Xhumari, "Comparing the Technologies used for the Application of Artificial Neural Networks," pp. 21–28, Apr. 2021.

[15] C. Nontapa, C. Kesamoon, N. Kaewhawong, and P. Intrapaiboon, "A Comparative of a New Hybrid Based on Neural Networks and SARIMA Models for Time Series Forecasting," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2021, pp. 94–105. doi: 10.1007/978-3-030-80253-0_9.

[16] R. L. Fraiha Lopes, S. G. C. Fraiha, H. S. Gomes, V. D. Lima, and G. P. S. Cavalcante, "Application of Hybrid ARIMA and Artificial Neural Network Modelling for Electromagnetic Propagation: An Alternative to the Least Squares Method and ITU Recommendation P.1546-5 for Amazon Urbanized Cities," *Int. J. Antennas Propag.*, vol. 2020, 2020, doi: 10.1155/2020/8494185.

[17] P. Unnikrishnan and V. Jothiprakash, "Hybrid SSA-ARIMA-ANN Model for Forecasting Daily Rainfall," *Water Resour. Manag.*, vol. 34, no. 11, pp. 3609–3623, 2020, doi: 10.1007/s11269-020-02638-w.

[18] N. Hema Priya, S. M. Adithya Harish, N. Ravi Subramanian, and B. Surendiran, "Covid-19: Comparison of Time Series Forecasting Models and Hybrid ARIMA-ANN," in *Lecture Notes in Networks and Systems*, 2022, pp. 567–577. doi: 10.1007/978-981-19-1122-4_59.

[19] A. Chahal, P. Gulia, N. S. Gill, and I. Priyadarshini, "A Hybrid Univariate Traffic Congestion Prediction Model for IoT-Enabled Smart City," *Inf.*, vol. 14, no. 5, 2023, doi: 10.3390/info14050268.

[20] E. Antony, N. S. Sreekanth, R. K. Sunil Kumar, and T. Nishanth, "Data preprocessing techniques for handling time series data for environmental science studies," *Int. J. Eng. Trends Technol.*, vol. 69, no. 5, pp. 196–207, 2021, doi: 10.14445/22315381/IJETT-V69I5P227.

[21] A. Asok, *Generalized approach to linear data transformation*. 2016. doi: 10.1109/ICDSE.2016.7823937.

[22] G. E. P. . Box, G. M. . and Jenkins, and G. Reinsel, "Time Series Analysis,Forecasting and Control, 4th edition. Wiley," *Prentice-Hall, Englewood Cliffs*, 2008.

[23] L. Ma, C. Hu, R. Lin, and Y. Han, "ARIMA model forecast based on EViews software," in *IOP Conference Series: Earth and Environmental Science*, 2018. doi: 10.1088/1755-1315/208/1/012017.

[24] V. Kotu and B. Deshpande, "Data Science Process," *Data Sci.*, no. 1, pp. 19–37, 2019, doi: 10.1016/b978-0-12-814761-0.00002-2.

[25] S. Khan and H. Alghulaiakh, "ARIMA model for accurate time series stocks forecasting," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 7, 2020, doi: 10.14569/IJACSA.2020.0110765.

[26] D. Kurniasari, R. Vidia Mahyunis, Warsono, and A. Nuryaman, "Implementation of Artificial Neural Network (ANN) using Backpropagation Algorithm", [Online]. Available: https://finance.yahoo.com/.

[27] N. Kushawaha and A. Roy, "A Study of Artificial Neural Network and Its Implementation from Scratch," *J. Robot. Autom. Res.*, vol. 3, no. 2, 2022, doi: 10.33140/jrar.03.02.05.

[28] M. Adil, R. Ullah, S. Noor, and N. Gohar, "Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design," *Neural Comput. Appl.*, vol. 34, no. 11, pp. 8355–8363, 2022, doi: 10.1007/s00521-020-05305-8.

[29] H. V. T. Mai, T. A. Nguyen, H. B. Ly, and V. Q. Tran, "Investigation of ANN Model Containing One Hidden Layer for Predicting Compressive Strength of Concrete with Blast-Furnace Slag and Fly Ash," *Adv. Mater. Sci. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/5540853.

[30] N. Alifiah, D. Kurniasari, Amanto, and Warsono, "Prediction of COVID-19 Using the Artificial Neural Network (ANN) with K-Fold Cross-Validation," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 9, no. 1, pp. 16–27, 2023, doi: 10.20473/jisebi.9.1.16-27.

[31] P. Ramachandran, B. Zoph, and Q. V Le, "Searching for Activation Functions. arXiv Prepr," *arXiv Prepr. arXiv1710.05941*, pp. 1–13, 2017.

[32] M. A. Mercioni and S. Holban, "P-Swish: Activation Function with Learnable Parameters Based on Swish Activation Function in Deep Learning," in *2020 14th International Symposium on Electronics and Telecommunications, ISETC 2020 - Conference Proceedings*, 2020. doi: 10.1109/ISETC50328.2020.9301059.

[33] T. Devi and N. Deepa, "A novel intervention method for aspect-based emotion Using Exponential Linear Unit (ELU) activation function in a Deep Neural Network," in *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, 2021, pp. 1671–1675. doi: 10.1109/ICICCS51141.2021.9432223.

[34] P. P. Ippolito, "Hyperparameter Tuning: The Art of Fine-Tuning Machine and Deep Learning Models to Improve Metric Results," in *Tourism on the Verge*, vol. Part F1051, 2022. doi: 10.1007/978-3-030-88389-8_12.

[35] M. A. Zöller and M. F. Huber, "Benchmark and Survey of Automated Machine Learning Frameworks," *Journal of Artificial Intelligence Research*, vol. 70. pp. 409–472, 2021. doi: 10.1613/JAIR.1.11854.

[36] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *Int. J. Comput. Appl.*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.

[37] Ü. Ç. Büyükşahin and Ş. Ertekin, "Improving forecasting accuracy of time series data using a new ARIMA-ANN hybrid method and empirical mode decomposition," *Neurocomputing*, vol. 361, pp. 151–163, 2019, doi: 10.1016/j.neucom.2019.05.099.

[38] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," in *Applied Soft Computing Journal*, 2011, pp. 2664–2675. doi:

10.1016/j.asoc.2010.10.015.
[39] C. N. Babu and B. E. Reddy, "A moving-average filter based hybrid ARIMA-ANN model for forecasting time series data," *Appl. Soft Comput. J.*, vol. 23, pp. 27–38, 2014, doi: 10.1016/j.asoc.2014.05.028.
[40] J. Lee, T. Yoon, S. Kwon, and J. Lee, "Model evaluation for forecasting traffic accident severity in rainy seasons using machine learning algorithms: Seoul city study," *Appl. Sci.*, vol. 10, no. 1, 2020, doi: 10.3390/app10010129.
[41] A. Botchkarev, "A new typology design of performance metrics to measure errors in machine learning regression algorithms," *Interdiscip. J. Information, Knowledge, Manag.*, vol. 14, pp. 45–76, 2019, doi: 10.28945/4184.
[42] D. Kim, T. Kim, J. Jeon, and Y. Son, "Soil-Surface-Image-Feature-Based Rapid Prediction of Soil Water Content and Bulk Density Using a Deep Neural Network," *Appl. Sci.*, vol. 13, no. 7, 2023, doi: 10.3390/app13074430.