Published online on: https://journal.uii.ac.id/jurnalsnati/
DOI: 10.20885/snati.v4.i2.40389

Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi
Vol. 4 No. 2 (2025) 82 - 91 ISSN Media Electronic: 2807-5935

Deep Learning-Based Bidirectional RNN for Cryptocurrency Price Prediction with Hyperparameter Tuning

Dori Gusti Alex Candra[1*], Nurdi Afrianto[2], Idir Fitriyanto[3], Eka Sofiati[4], Budi Permana Putra[5]
[1,3,4,5]Department Of Information Systems, Institut Teknologi Mitra Gama, Riau, Indonesia
[2]Department Of Information Technology, Institut Teknologi Mitra Gama, Riau, Indonesia
[1*]dorigustialexcandra@gmail.com, [2]nurdiafrianto1995@gmail.com, [3]idirfitriyanto45@gmail.com, [4]ekasfti@gmail.com, [5]budipermanaputra96@gmail.com

Abstract

Predicting cryptocurrency is difficult because it has high volatility, where prices can experience spikes or declines due to market dynamics. This study focuses on NameCoin, one of the oldest altcoins originating from Bitcoin. NameCoin was selected because it has relatively stable and extensive historical data. The objective of this study is to evaluate the performance of the Bidirectional Recurrent Neural Network (BiRNN) in predicting NameCoin price movements. This study employs an experimental method using historical data as input for the training process. Hyperparameter tuning is conducted systematically using four different scenarios to obtain the optimal model configuration. The dataset is divided into 80% for training the model and 20% for testing the performance of the trained model. Model performance is evaluated using RMSE, MSE, MAPE, coefficient of determination (R^2), Directional Statistic (D-Stat), and loss value as indicators of model accuracy and stability. The experimental results show that Scenario 1 produces the most optimal performance, with RMSE = 0.0216, MAPE = 2.59%, R^2 = 0.9899, D-Stat = 53.71%, and the smallest loss value of 0.0012. These performance metrics indicate that the BiRNN model effectively captures nonlinear trends and accurately predicts the direction of price movements. Conversely, Scenario 3 had the worst performance, with a MAPE of 10.19%. By comparing these scenarios, it is clear that the configuration in Scenario 1 outperforms the others in terms of prediction accuracy and model stability against data fluctuations.

Keywords: *Deep learning; Bidirectional Recurrent Neural Network; cryptocurrency; Hyperparameter tuning; Evaluation metrics*

1. Introduction

Bank Indonesia categorizes non-cash payment instruments into five types: payment cards, checks, current account deposits, debit instruments, and electronic money [1]. In recent years, the use of non-cash payment systems has steadily increased and become a major trend. This shift has transformed the way people conduct transactions in their daily activities due to its practicality and efficiency [2]. In the ASEAN region, electronic money transactions have contributed positively to economic growth, whereas credit card and check transactions have not shown a significant impact [3]. Non-cash transactions experienced a compound annual growth rate (CAGR) of 48.17% in terms of volume and 16.15% in transaction value from 2014 to 2015 to 2019–2020 [4]. In Eastern countries, non-cash payments have had a positive and significant contribution to per capita GDP growth, particularly in Central and Eastern Europe [5]. However, non-cash payments also present certain drawbacks. One of the most significant limitations is restricted acceptance, as they can only be used at participating merchants [6]. The implementation of non-cash transactions in regional finance is hindered by banking and regulatory issues, as well as the potential threats of cybercrime, which raise concerns among users [7]. Furthermore, electronic money systems face challenges in maintaining user privacy and data confidentiality [8].

Innovations in financial systems to address issues related to electronic money, such as blockchain, have led to the emergence of various alternative currencies known as cryptocurrencies [9]. Cryptocurrency is a digital currency system that utilizes blockchain technology as its underlying storage mechanism [10]. Blockchain offers high-level security and is decentralized by nature, which means that the system is not controlled by any single authority or central entity [11]. All transactions within the blockchain are transparent, making cryptocurrency transactions easily traceable [12]. Among the areas of concern in the digital assets space is the volatility of cryptocurrency prices. Research shows that the root causes of the sort of volatility are the market supply and demand, the

Received: 19-05-2025 | Revised: 01-07-2025 | Accepted: 03-07-2025 | Published: 10-07-2025

mining level, and the advancement of blockchain technology [13]. Another problem in the case of cryptocurrency time-series is its propensity to act like Brownian noise thus being highly complicated and unpredictable [14]. The inherent volatile and complex behavior of cryptocurrency is the prominent setback for the accurate price prediction [15] .

The application of machine learning in cryptocurrency research has continued to grow, particularly in price prediction and trend analysis [16]. Various machine learning techniques, such as classification and time series models, have been explored. The availability of diverse datasets and resources has facilitated the integration of ML algorithms in crypto trading research [17]. In one study [18], linear regression was used to predict Bitcoin prices. Researchers used one of the financial indicators such as Simple Moving Average (SMA) and Exponential Moving Average (EMA). for trend analysis, along with key levels like support and resistance for trend confirmation. The dataset, obtained from Kaggle, represented Bitcoin prices in USD, ranging from December 1, 2014, to April 22, 2020, consisting of 8 columns: Timestamp, Opening Price, Highest Price, Lowest Price, Closing Price, Volume in BTC, Volume in Currency, and the Weighted Average Price. The results showed a model accuracy of 96.97%. Another study [19] implemented the Support Vector Regression (SVR) algorithm to predict cryptocurrency prices using datasets for Bitcoin, XRP, and Ethereum. The focus of the study was on kernel types, namely linear, polynomial, and radial basis function (RBF). The findings indicated that the parameters Gamma and C significantly impacted the SVR model, with Gamma values of 0.00001 and 0.00005 for the polynomial and RBF kernels, respectively. The optimal epsilon value was 0.2. The RBF kernel shows superior performance compared to others when evaluated on the Mean Absolute Error (MAE) metric, Mean Squared Error (MSE) evaluation metric, Root Mean Square Error (RMSE) similarity, as well as the Coefficient of Determination (R^2) metric. Achieving an R^2 score of 78% for Bitcoin, higher than Ethereum's 56%, highlighting RBF's effectiveness for non-linear data. In a different study [20], ARIMAX, FBProphet, and XGBoost algorithms were used to analyze Bitcoin prices. Research evaluation is based on evaluation metrics, namely the Root Mean Square Error (RMSE) metric, the Mean Absolute Error (MAE) evaluation metric, and the R^2 metric. The dataset, sourced from Kaggle, consisted of 8 features. ARIMAX proved to be the most effective for Bitcoin price prediction with an RMSE of 322.4, while FBProphet and XGBoost achieved lower RMSE values of 229.5 and 369, respectively. Another study [21] aimed to predict cryptocurrency market trends. The dataset was compiled from various crypto exchanges beginning in 2018. Key indicators used for movement analysis included Simple Moving Average (SMA), Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and On-Balance Volume (OBV). The study evaluated several machine learning models, including algorithm Quadratic Discriminant Analysis (QDA), algorithm K-Nearest Neighbors (KNN), algorithm Logistic Regression, and Neural Networks. Logistic Regression outperformed the other models with an accuracy score of 0.979, MSE of 0.418, AUC of 0.633, and cross-validation (CV) score of 0.878.

The selection of Namecoin here is because of the age and continuity of its data. As one of the first cryptos and the first altcoin to be created out of Bitcoin's source code, Namecoin offers a relatively long and stable historical dataset, which is highly suited to modeling through time series [22] . Unlike most of the other cryptos either recently developed or highly volatile, Namecoin is less volatile in order to provide more stable data, which is better for prediction. Also, scientific literature that is specifically related to the price forecasting of Namecoin is scarce, and the current work is to meet the growing literature on the price prediction of Namecoin.

Bidirectional Neural Networks (BiNNs) have many advantages in different areas of application. For natural language processing (NLP), BiNNs can be used for improving the diagnostic prediction performance without the employment of additional parameters, and also for enhancing named entity recognition by handling boundary tag sparsity and reducing error propagation [23]. In time-series condition monitoring and fault diagnosis, BiNNs outperform conventional architectures in error detection and classification situations [24]. BiNNs were also successful in speech recognition and text-processing-based applications [25]. The fact that they can process information in the forward and backward directions is particularly important here, in the sense that the model is better able to process long-range dependencies and overall context [23].

2. Research Methods

This study consists of a series of systematically designed steps aimed at achieving optimal results. Each step plays a crucial role and must be carried out sequentially to ensure an effective research process. The process begins with the data collection phase, where relevant historical data related to cryptocurrency is gathered for further analysis.

Once the data has been acquired, the next step is data preprocessing. In this phase, the data is cleaned and prepared to align with the research requirements and to be ready for modeling. This is followed by the model design and implementation stage, where the model is developed according to the objectives of the study. The final stage is the evaluation phase, where the performance of the developed model is tested and analyzed based on predefined evaluation metrics. The overall workflow of this research is illustrated in the flowchart presented in Figure 1.

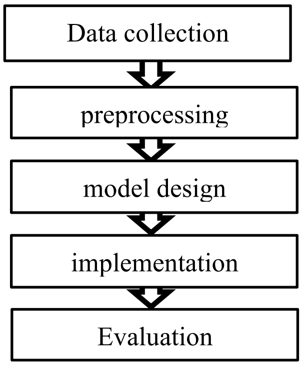
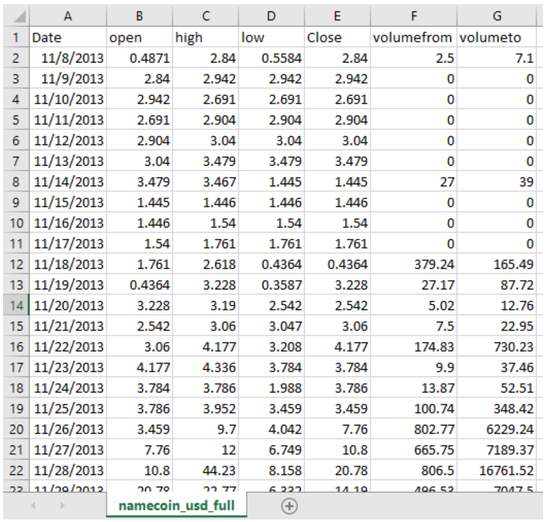


Figure 1. research flow diagram

2.1. Data collection

This study utilizes historical data from the cryptocurrency Namecoin as the primary subject of analysis. The selection of Namecoin is based on its significant role in the evolution of blockchain technology and its influence on the dynamics of the global digital asset market. A total of 4,207 data points were collected from the CryptoCompare website, covering the period from November 8, 2013, to May 14, 2025. This extended time span was intentionally chosen to evaluate the long-term effects of market dynamics on the performance of Namecoin. An illustration of the dataset used in this study is presented in Figure 2, sourced from CryptoCompare.



	A	B	C	D	E	F	G
1	Date	open	high	low	Close	volumefrom	volumeto
2	11/8/2013	0.4871	2.84	0.5584	2.84	2.5	7.1
3	11/9/2013	2.84	2.942	2.942	2.942	0	0
4	11/10/2013	2.942	2.691	2.691	2.691	0	0
5	11/11/2013	2.691	2.904	2.904	2.904	0	0
6	11/12/2013	2.904	3.04	3.04	3.04	0	0
7	11/13/2013	3.04	3.479	3.479	3.479	0	0
8	11/14/2013	3.479	3.467	1.445	1.445	27	39
9	11/15/2013	1.445	1.446	1.446	1.446	0	0
10	11/16/2013	1.446	1.54	1.54	1.54	0	0
11	11/17/2013	1.54	1.761	1.761	1.761	0	0
12	11/18/2013	1.761	2.618	0.4364	0.4364	379.24	165.49
13	11/19/2013	0.4364	3.228	0.3587	3.228	27.17	87.72
14	11/20/2013	3.228	3.19	2.542	2.542	5.02	12.76
15	11/21/2013	2.542	3.06	3.047	3.06	7.5	22.95
16	11/22/2013	3.06	4.177	3.208	4.177	174.83	730.23
17	11/23/2013	4.177	4.336	3.784	3.784	9.9	37.46
18	11/24/2013	3.784	3.786	1.988	3.786	13.87	52.51
19	11/25/2013	3.786	3.952	3.459	3.459	100.74	348.42
20	11/26/2013	3.459	9.7	4.042	7.76	802.77	6229.24
21	11/27/2013	7.76	12	6.749	10.8	665.75	7189.37
22	11/28/2013	10.8	44.23	8.158	20.78	806.5	16761.52

namecoin_usd_full

Figure 2. Dataset

Table 1 presents the column features that capture daily information on the price and trading volume of the Namecoin cryptocurrency. The date column indicates the recording date, which is then converted into a standard date format to facilitate time-based analysis. The price column reflects the average price of Namecoin on that particular day, while the open, high, and low columns represent the opening price, the highest price, and the lowest price during that trading day, respectively. Additionally, there are two volume-related columns: volumefrom and volumeto. The volumefrom column indicates the total number of Namecoin units traded, whereas the volumeto column represents the total value of those trades in US dollars. This combination of data supports the analysis of price trends as well as market activity for Namecoin over time.

Table 1. Dataset Features

No.	Column Name	Data type	Information
1	Date	object	Price recording date
2	Price	float64	Namecoin average price on the day
3	Open	float64	Namecoin opening price on the day
4	High	float64	Namecoin's highest price in one day
5	Low	float64	Namecoin lowest price in one day
6	volumefrom	float64	Number of Namecoin units traded on that day
7	volumeto	float64	Total trading value of Namecoin in USD

2.2. Data Preprocessing

- Missing Value

Missing values often present significant challenges in data analysis. Proper handling of these missing values can lead to more accurate conclusions and have a substantial impact on the performance and development of machine learning models.

- Outlier

Outliers refer to values that significantly deviate from the data distribution pattern within a statistical sample. These values differ markedly from other data points in the same column, and their presence can provide important information for the ongoing data analysis.

- Data Normalization

The Min-Max normalization method is an approach that applies a linear transformation to raw data to compare values before and after

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \quad (1)$$

In the normalization process, x_{new} represents the adjusted value, while x_{old} is the original data value. The smallest value in the dataset is denoted as x_{min}, and the largest value as x_{max}. Through this approach, the data is scaled typically between 0 and 1, which helps accelerate the training process of the analysis model while maintaining stability and accuracy when processing cryptocurrency data.

2.3. Bidirectional RNN

Deep learning techniques, especially algorithm Recurrent Neural Networks (RNN), are commonly used to process various sequential data, including tasks like time series analysis [26]. RNNs have the ability to remember previously learned information, enabling them to predict subsequent outcomes more accurately [27]. Their operation involves feeding back outputs generated in the hidden layers, which allows the information to be retained longer and used to correct errors in subsequent processes through weight updates [28]. Figure 2 illustrates the RNN process before and after being unrolled into a full network. x_t is the input at time t, while h_t is the output of the RNN process. Once unrolled, the entire network becomes fully visible. The process starts by taking x_0 as input, producing h_t, which together with x_t serves as input for the next step. Thus, h_t and x_t become inputs for subsequent steps, as expressed in equation (1)

$$h_t = \sigma_h(w_{xh}\, x_t + w_{hh}\, h_{t-1} + b_h) \quad (1)$$

The matrix w_{hx} represents the set of weights establishing the connection between the input and hidden layers, w_{hh} is the weight matrix for the recurrent connections, The bias vector is represented by b_h, and σ_h corresponds to the activation function most often realized through tanh or ReLU. [29]. The output at each time step t is calculated using equation (2).

$$y_t = \sigma_y\,(w_{hy} h_t + b_y) \quad (2)$$

We use w_{hy} to denote weights contained in the hidden layer as well as the output layer, b_y for a bias vector. σ_y denotes the activation function for use at the output layer output layer of a deep learning model created.

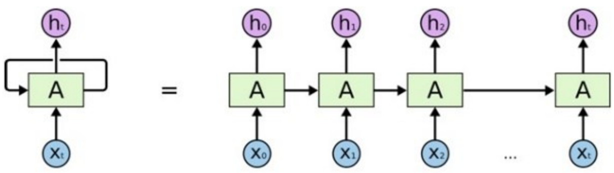


Figure 2. recurrent neural network architecture

Bidirectional recurrent neural network (BiRNN) refers to a form of recurrent neural network that has been improved by processing two directions of input data, with outputs being summed together to generate the final output [30]. Figure 3 illustrates the BiRNN architecture. A pair of RNN cells, called BiRNN (f) for the forward recurrent neural network and BiRNN (b) for the backward recurrent neural network, form the BiRNN [31]. These cells analyze the input data sequence in two different directions, namely forward and backward.

The forward BiRNN processes time steps from $t = T$ to T, while the backward BiRNN processes from $t = T$ to 1. This requires the simultaneous calculation of both the forward hidden sequence of layers in the algorithm h_f and the backward hidden sequence h_b in an RNN. Both the forward and backward hidden sequences h_f and h_b are computed concurrently [30]. These two sequence equations are then combined to update the output sequence, namely y. This is shown in equations (3), (4), and (5).

$$hf = Wxhf\ xt + Whf\ hf\ (t-1) + bhf \quad (3)$$
$$hb = Wxhb\ xt + Whb\ hb(t-1) + bh \quad (4)$$

Matrices w_{xhf} and w_{xhb} are weight parameters which map input x_t into hidden state representations for the forward network and backward network, respectively. Meanwhile, w_{hf} and w_{hb} are the weight matrices responsible for linking the hidden states from the Past time steps lead to hidden state layers used at the current time. The bias known as a vector symbolized b_{hf} and the weights b_{hb} are incorporated into hidden states to enable the model to learn necessary adjustments, enhancing the fit between the data and its predictions. At every moment of time t, the forward hidden state h_{ft} as well as backward hidden state h_{bt} are merged to generate the output sequence y_t.

$$yt = Whf\ hft + Whb\ hbt + by\,. \quad (5)$$

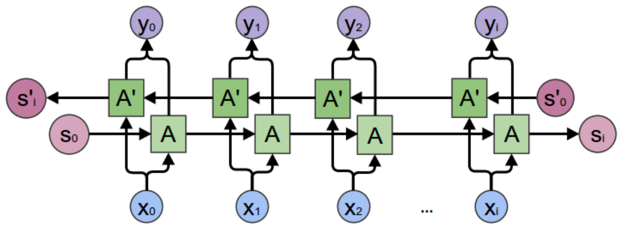


Figure 2. bidirectional recurrent neural network architecture

Following the construction of a model, subsequent steps are to perform a few tests to find out how accurate the model we have created. In this research, two statistical tests are employed, namely the Mean Squared Error (MSE) evaluation metric, and the Root Mean Squared Error (RMSE) metric, both designed to provide a measure of prediction error magnitude.

Using these two measures, one can get a better idea concerning how accurately the model can forecast the data. MSE can be understood as a measure representing the average squared differences between real values versus forecasted outputs [32].

The calculation starts by subtracting the predicted value from the actual value, then squaring the result.

All squared differences are then summed and divided by the total number of data points analyzed. The complete MSE formula is presented in Equation (6), where *Actual* refers to the true values, *Prediction* denotes the predicted values generated by the model, and n represents the total amount of data points analyzed.

$$MSE = \frac{\sum Actual - Prediction^2}{n} \tag{6}$$

RMSE (Root Mean Square Error) refers to a statistical approach to calculate the prediction error generated based on actual values by determining the square root value derived from the average of the squares of the differences between the actual and predicted values. This metric indicates the average magnitude of errors in the model, thereby helping to evaluate the accuracy of the prediction results against the actual data [33]. The formula for RMSE is presented in Equation (7).

$$RMSE = \sqrt{\frac{\sum(Actual - Prediction)}{n}} \tag{7}$$

MAPE (Mean Absolute Percentage Error) is an equivalent representation of MAE. Mean Absolute Percentage Error (MAPE) measures the average degree of deviation between a predicted value and the actual observation, reflecting the relative accuracy of the model's forecasts. This evaluation metric is used to assess the relative size of errors rather than the absolute errors [32]. The formula for MAPE is shown in Equation (8).

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \frac{A_t - F_t}{A_t} \ x \ 100\% \tag{8}$$

A_t represents the actual value observed at time t, while f_t is the value of a prediction based on the same point in time. n indicates the number of total observations in a data set. The coefficient of determination in this case R^2 (R-Square), is a metric from statistics that reflects the amount of variance in the dependent variable that can be explained by the independent variables in the structure created. R^2 varies between 0 and 1, with a value representing how well the group of independent variables affects the dependent variable. The R^2 formula can be observed in equation (9).

$$R^2 = 1 - \frac{\sum_i (y_i - \widehat{y_i})^2}{\sum_i (y_i - \overline{y_i})^2} \tag{9}$$

Where y_i is a data point's actual value, denoting a real value observed. $\widehat{y_i}$ denotes a model's forecast value for a data point i, while $\bar{y}$ denotes a mean value for all actual values y_i contained in a data set, denoting a general value of data. Besides measuring prediction errors, it is also important to assess the direction of cryptocurrency movements. Directional Statistics (Dstat) is used to evaluate how well the model predicts the movement direction [34]. Dstat computation is outlined in Equations (10) and (11) with n representing the training sample quantity, x referring to actual data, and $\hat{x}$ referring to forecasted values. A higher Dstat value indicates better prediction accuracy of the model.

$$Dstat = \frac{1}{n} \sum_{t-1}^{n} at \ x \ 100\% \tag{10}$$

$$a_t = \begin{cases} 1 \ (x_{t+1} - x_t)\left(\hat{x}_{t+1} - x_{t)}\right) \geq 0 \\ 0 \ (x_{t+1} - x_t)\left(\hat{x}_{t+1} - x_{t)}\right) < 0 \end{cases} \tag{11}$$

3. Results and Discussions

The Namecoin dataset utilized in this study was sourced from the CryptoCompare website. The dataset covers the period from November 8, 2013, to May 14, 2025. It contains several attributes, including Date, Price, Open, High, Low, Volume From, and Volume To.

The data had been explored in Python for missing values before training the model. From the results, there were no missing values to be seen in the Close attribute. Outlier detection has also been completed using the application of the Interquartile Range (IQR) method. Data instances which were outside of the range 1.5 IQR were considered to be the outliers. From the original 4,206 entries, 72 were found to be the outliers after detection, therefore giving an otherwise clean 4,134 data instances. Data preprocessing had been completed to address the quality of the data, to minimize the impact of the extreme values, and to also optimize the model's performance while being trained.

After undergoing data preprocessing, which serves to clean the data, the total number of records ready for use in this study is 4,134. The dataset is divided into several segments: 80% is used for the training process and other segments, 20% is reserved for testing, which serves to assess the model's effectiveness.

During the preprocessing stage, detection and handling of outlier values were conducted to ensure the quality of data used in modeling. This procedure utilized the Interquartile Range (IQR) approach, whereby Q1 and Q3 were determined from Close price data. The IQR was then found by deducting Q1 from Q3. The lower limit was where Q1 was reduced by 1.5 IQR, while the upper limit was where Q3 was added by 1.5 IQR. Points outside this selection were classified as outliers and eliminated from the data. The final result is a cleaned dataset free from extreme values, making it more representative and stable for model training. Figure 3 illustrates the IQR process using Python.

```python
# --- OUTLIER DETECTION ---
Q1 = df_close['Log_Close'].quantile(0.25)
Q3 = df_close['Log_Close'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
df_clean = df_close[(df_close['Log_Close'] >= lower_bound) & (df_close['Log_Close'] <= upper_bound)].copy()
```

Figure 3. IQR dengan python

Figure 4 shows the histogram of the dataset. The histogram indicates a normal distribution, characterized by a bell-shaped curve. Most of the data is concentrated around the central value, with frequencies decreasing as values move away from the center. Further analysis using a boxplot reveals that most of the data falls within the range of 1.75 to 1.76, indicating the highest data concentration in this interval. This information is crucial for understanding the data's characteristics and serves as a fundamental basis for the next stage, which is data modeling.

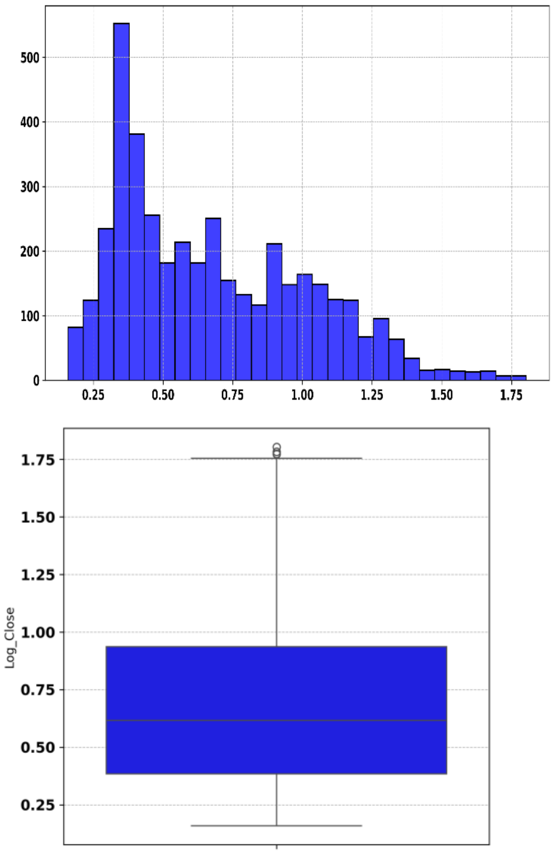


Figure 4. data distribution with histogram and boxplot

To support the implementation of deep learning modeling, this research utilized hardware with an Intel Core i5 8250U processor, 12 GB of RAM, and the TensorFlow framework version 2.4.0. The model developed in this study was programmed in Python using four scenarios. For example, Figure 5 illustrates the architecture of the bidirectional RNN. The model begins with the first Bidirectional RNN layer containing 32 units using activation, one of which is tanh and return_sequences set to True to enable stacked layers. The Dropout layer comes after this stage with a rate of 20% to reduce the risk of overfitting. This structure is repeated for the second and third layers, each consisting of a Bidirectional RNN with 32 units. The final layer is a Dense layer serving as the output, with one unit to produce the prediction. The model is then compiled using the Adam optimization algorithm.

```python
# --- BUILD MODEL SESUAI Scenario 1 ---
model = Sequential()
model.add(Bidirectional(SimpleRNN(32, activation='tanh', return_sequences=True), input_shape=(window, 1)))
model.add(Dropout(0.2))
model.add(Bidirectional(SimpleRNN(32, activation='tanh', return_sequences=True)))
model.add(Dropout(0.2))
model.add(Bidirectional(SimpleRNN(32, activation='tanh')))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')

# --- TRAINING ---
#history = model.fit(trainX, trainY, epochs=100, batch_size=32, verbose=1)
history = model.fit(trainX, trainY, epochs=100, batch_size=32, verbose=1, validation_split=0.2)
```

Figure 5. code algoritma bidiirectional RNN dengan python

In this research, each experiment was formulated to optimize the parameters in training bidirectional RNN. The window size indicates how much prior data are utilized to forecast the next value. The hidden layer indicates how many bidirectional RNNs are utilized in the model. Epoch indicates how many full cycles are conducted forward and backward through the full training data. The number of units in bidirectional RNN refers to the quantity of neurons per RNN layer. Dropout is implemented to prevent overfitting. The lastly utilized optimizers in this research include Adam, Root Mean Square Propagation (RMSprop) optimization, and Stochastic Gradient Descent (SGD) optimization. This research utilized four different hyperparameter tuning scenarios to assess the bidirectional RNN model's performance. Each scenario differed in terms of window size, hidden layer count, neuron quantity, dropout rate, optimizer, and the number of training epochs. Table 2 presents the four scenarios used in this study. Scenario 1 used a window size of 3 with 3 hidden layers and 32 neurons per layer. The dropout rate was set at 0.2 to prevent overfitting, training was conducted for 100 epochs, and Adam was used as the optimizer. Scenario 2 increased complexity with a window size of 5, 4 hidden layers, and 64 neurons per layer. The dropout rate was 0.3, training lasted for 300 epochs, and the optimizer remained Adam. Scenario 3 featured an even more complex configuration: window size of 7, 5 hidden layers, and 100 neurons per layer. Dropout was kept at 0.3, the optimizer switched to RMSprop, and training was extended to 600 epochs. Finally, Scenario 4 applied the most complex parameters: window size of 10, 6 hidden layers, and 128 neurons per layer. The dropout rate was lowered to 0.2, but training ran for 1000 epochs. The optimizer used was SGD with an adjusted learning rate. Table 2 shows the hyperparameter tuning details for each scenario. The evaluation dataset consisted of 827 data points, spanning from February 8, 2023, to May 14, 2025. Model evaluation focused on metrics including MSE, RMSE, MAE, MAPE, R^2, and D-Stat. Table 3 presents the testing results of the bidirectional RNN model's performance across the four scenarios. The evaluation metrics applied in this study include Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), Coefficient of Determination (R^2), Directional Statistic (D-Stat), and Loss.

Table 2. Hyperparameter tuning of bidirectional RNN models

Scenario	Window Size	Number of Layers	Number of Units	Dropout	Optimizer	Epochs
1	3	3	32	0.2	Adam	100
2	5	4	64	0.3	Adam	300
3	7	5	100	0.3	RMSprop	600
4	10	6	128	0.2	SGD (lr=1000)	1000

In Scenario 1, the model achieved its best performance, with an RMSE of 0.0216, MSE of 0.0005, and an MAPE of 2.59%. The R^2 value obtained was 0.9899, which shows that the model takes into account almost 99% of the variance in the actual data. The D-Stat value was fairly high at 53.71%, meaning more than half of the price movements were correctly predicted. The training loss was low, at 0.0012. Scenario 2 still demonstrated good results, although there was a slight decline across all metrics compared to Scenario 1. The RMSE increased to 0.0238, MAPE rose to 2.93%, and R^2 slightly decreased to 0.9878. The D-Stat dropped to 49.45%, with a loss value of 0.0013.

Next, Scenario 3 also encompassed a higher complexity compared to the initial scenarios, but the performance was much poorer. Though the level of complexity in Scenario 4 was even higher, the model performance rose to a slight level. It would therefore show that the introduction of complexity to the model does not normally result in higher precision and the interaction of the hyperparameters such as the optimizer type used can largely define the outcome of the model. RMSE jumped to 0.0643, MAPE increased to 10.19%, and R^2 sharply declined to 0.9108. Although the D-Stat remained around 49%, these results indicate that excessive complexity reduced prediction accuracy. The loss also increased to 0.0034. Finally, Scenario 4 showed improvement again compared to Scenario 3. The RMSE decreased to 0.0432, MAPE improved to 7.20%, and R^2 rose to 0.9597. The D-Stat slightly increased to 50.25%, and the loss dropped to 0.0015.

Table 3. Results of testing four model scenarios

Scenario	RMSE	MSE	MAPE	R^2	D-Stat	Loss
1	0.0216	0.0005	2.59%	0.9899	53.71%	0.0012
2	0.0238	0.0006	2.93%	0.9878	49.45%	0.0013
3	0.0643	0.0041	10.19%	0.9108	49.82%	0.0034
4	0.0432	0.0019	7.20%	0.9597	50.25%	0.0015

Based on the results presented in Table 3 and Figure 4, as well as considering the values of MSE, RMSE, MAPE, and R^2, which do not show significant differences, it can be concluded that the models developed in Scenario 1 and Scenario 2 represent better alternatives compared to the other scenarios. Both scenarios are capable of accurately capturing the patterns of cryptocurrency value changes, producing predictions that closely resemble the actual price movements both visually and numerically. Consequently, these two models demonstrate strong potential for application in future cryptocurrency analysis and forecasting.

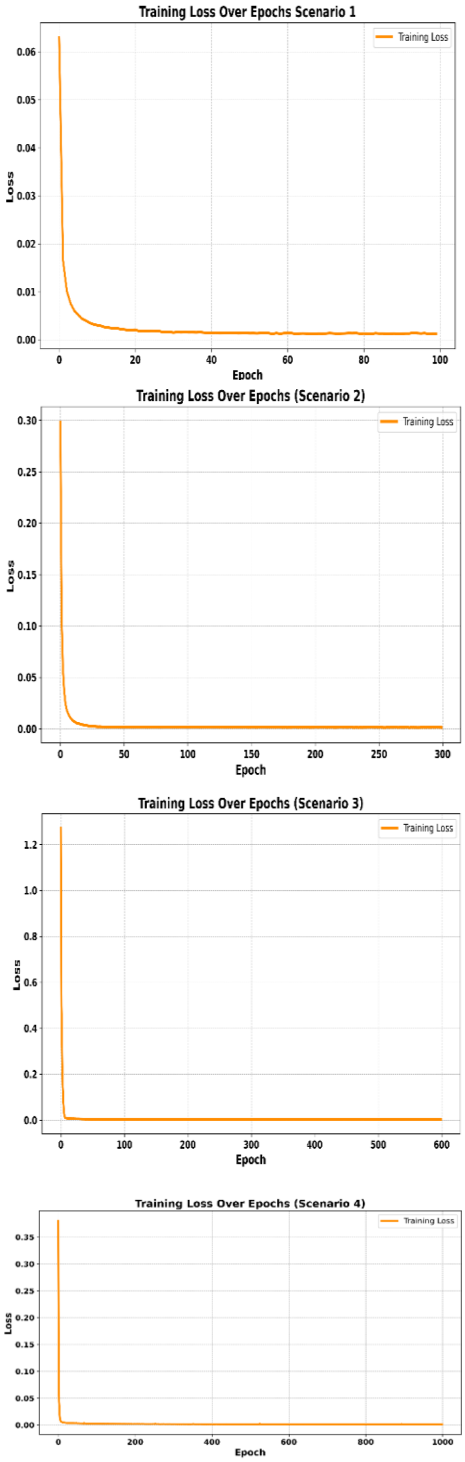


Figure 3. Loss value in the training process

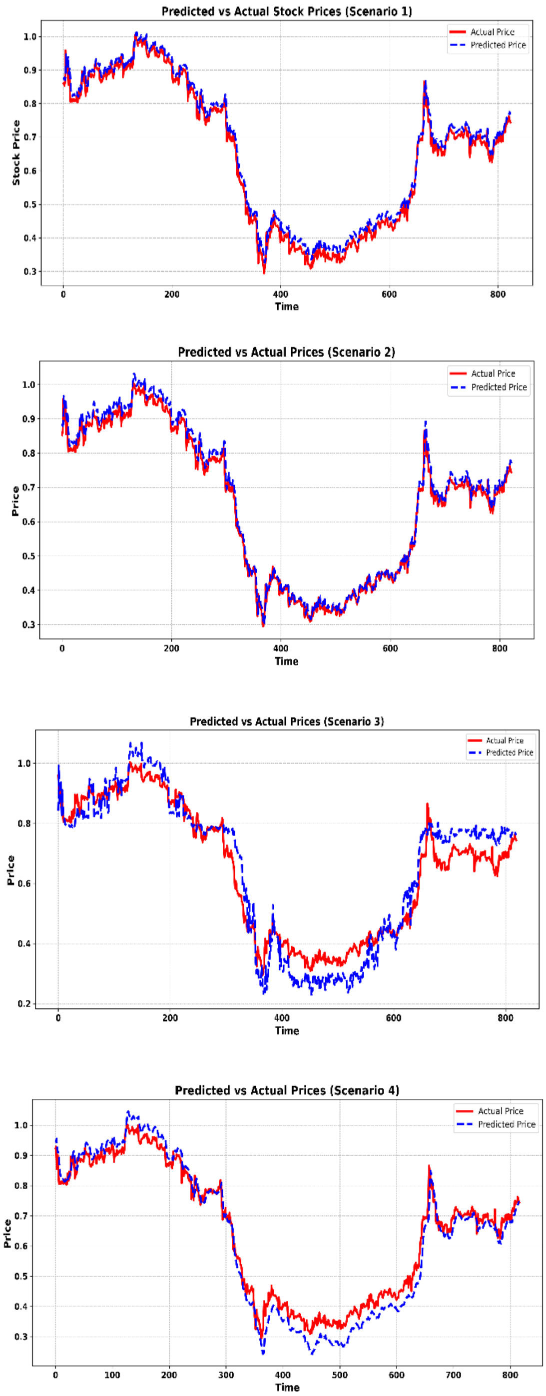


Figure 4. Prediction results for Scenarios 1, 2, 3, and 4 (top to bottom)

Based on the data presented in Table 4, it can be observed that the price predictions generated by the model in Scenario 1 exhibit a high level of closeness to the actual cryptocurrency prices in the test data. For instance, in the first observation, the actual crypto price is 0.859509, while the model's predicted value closely approximates this at 0.85282. This consistency continues in subsequent rows, such as the second and third, where the model predicts prices of 0.848819 and 0.853318 for actual values of 0.861623 and 0.852285, respectively. Although there are differences between the actual and predicted values, these discrepancies remain minor and fall within acceptable limits for time series analysis of volatile crypto data. At row 823, the model predicts a crypto price of 0.747050, which is very close to the actual value of 0.74384. These results indicate that the model developed under Scenario 1 possesses superior capability in recognizing historical price patterns and holds strong potential for future research applications. Therefore, the Scenario 1 model is considered more effective in producing cryptocurrency price estimates that closely resemble actual market values compared to the other scenarios.

Table 4. Best Prediction Result (Scenario 1)

No.	Price crypto test	Price crypto Prediction
1.	0.859509	0.85282
2.	0.861623	0.848819
3.	0.852285	0.853318
4.	0.864155	0.845789
5.	0.958584	0.85328
…	…..	……
819	0.746688	0.74067
820	0.762207	0.741967
821	0.762207	0.75339
822	0.747162	0.751501
823	0.74384	0.747050

The results from Scenarios 1 and 2 were better than those from the more complex Scenarios 3 and 4. This improved performance likely comes from their simpler model design, which created a more stable training environment and helped prevent overfitting. A simpler model fared better in detecting the underlying patterns because Namecoin is subject to mild volatility and not abrupt peaks. On the other hand, Scenarios 3 and 4's complex models often detected noise rather than significant signals. Most of the time, the condition arises when the complexity of the model exceeds the dataset requirements, particularly in the case of the time-series type of data like cryptocurrency rates. While this study focuses on Namecoin, its findings apply to other cryptocurrencies with similar traits. Simpler models are expected to perform well for cryptocurrencies that have plenty of historical data, stable volatility, and steady trading volume. To back up this potential, future research should include other cryptocurrencies like Ethereum, Litecoin, and Ripple, using similar model designs.

4. Conclusion

This study aims to examine the effectiveness of the BiRNN model in analyzing cryptocurrency price movements, specifically Namecoin. For measuring the model's accuracy, hyperparameter optimization was performed through four experimental configurations, utilizing evaluation metrics including regression evaluation metrics RMSE, MSE metrics, Mean Absolute Percentage Error (MAPE) and also R-squared (R^2), Directional Statistic (D-Stat), and Loss. The evaluation results from the four scenarios indicate that Scenario 1 achieved the best results in forecasting

Namecoin prices using the model. Scenario 1 attained an RMSE value of 0.0216, an MSE of 0.0005, and a MAPE of only 2.59%, indicating a very low prediction error rate. Furthermore, the R-squared value was 0.9899, and the Directional Statistic reached 53.71%. Lastly, the Loss value of 0.0012 was the lowest compared to other scenarios.

For future research development, it is recommended to explore alternative models such as Deep RNN or CNN-BiLSTM combinations to compare prediction performance. The inclusion of external variables, such as transaction volume and market sentiment, is also necessary. Additionally, the use of more systematic hyperparameter optimization methods and evaluation of validation loss should be considered to prevent overfitting. Testing the model on other cryptocurrency assets and integrating the model into practical applications are also important to enhance the applicability of the research.

Reference

[1] N. A. Hasniawati, E. R. Lase, and A. R. Hutabarat, "Indonesian Household Payment Choice: A Nested Logit Analysis," J. Cent. Bank. Theory Pract., vol. 9, pp. 291–313, 2020, doi: 10.2478/jcbtp-2020-0032.

[2] A. Karimah, I. U. Widyaningsih, F. A. Shavab, T. N. Sari, and H. Nupus, "Penyelenggaraan Transaksi Keuangan Non Tunai Sebagai Sistem Pembayaran Non Sipil Belanja Dinas Pegawai Negeri Sekretariat Daerah Kota Cilegon," AKUMULASI Indones. J. Appl. Account. Financ., vol. 2, no. 1, pp. 13–23, 2023.

[3] Y. Ferra and S. R. Ajija, "The Influence of Non-Cash Payment Transactions on Economic Growth in 5 ASEAN Countries," J. Ilmu Ekon. Terap., vol. 8, no. 2, pp. 272–283, 2023, doi: 10.20473/jiet.v8i2.51745.

[4] N. Zaware, "an Analytical Study of Cashless Transformation and Growth in Retail Market in India," SSRN Electron. J., vol. c, pp. 19–26, 2021, doi: 10.2139/ssrn.3819230.

[5] M. Grzelczak and M. Soliwoda, "Do non-cash payments affect economic growth? Empirical evidence from EU countries," Sci. Pap. Silesian Univ. Technol. Organ. Manag. Ser., vol. 2023, no. 166, pp. 301–317, 2023, doi: 10.29119/1641-3466.2022.166.20.

[6] E. A. Sinambela and D. Darmawan, "Strengths and Weaknesses of Using Electronic Money as a Substitute for Cash," J. Soc. Sci. Stud., vol. 2, no. 2, pp. 56–68, 2022, doi: 10.56348/jos3.v2i2.27.

[7] W. Charoenwet, P. Thongtanunam, V. T. Pham, and C. Treude, Toward effective secure code reviews: an empirical study of security-related coding weaknesses, vol. 29, no. 4. 2024.

[8] G. Fragkos, C. Minwalla, J. Plusquellic, and E. E. Tsiropoulou, "Artificially Intelligent Electronic Money," IEEE Consum. Electron. Mag., vol. 10, no. 4, pp. 81–89, 2021, doi: 10.1109/MCE.2020.3024512.

[9] Gioia Arnone, "Blockchain and Cryptocurrency Innovation for a Sustainable Financial System," Int. J. Ind. Manag., vol. 15, no. 1, pp. 1–16, 2022, doi: 10.15282/ijim.15.1.2022.8994.

[10] F. Akhtar et al., "Potential of Blockchain Technology in Digital Currency: A Review," 2019 16th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. ICCWAMTIP 2019, pp. 85–91, 2019, doi: 10.1109/ICCWAMTIP47768.2019.9067546.

[11] A. Zwitter and J. Hazenberg, "Decentralized Network Governance: Blockchain Technology and the Future of Regulation," Front. Blockchain, vol. 3, no. March, pp. 1–12, 2020, doi: 10.3389/fbloc.2020.00012.

[12] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, "A Blockchain-Based Multi-Cloud Storage Data Auditing Scheme to Locate Faults," IEEE Trans. Cloud Comput., vol. 10, no. 4, pp. 2252–2263, 2022, doi: 10.1109/TCC.2021.3057771.

[13] H. Gupta and R. Chaudhary, "An Empirical Study of Volatility in Cryptocurrency Market," J. Risk Financ. Manag., vol. 15, no. 11, 2022, doi: 10.3390/jrfm15110513.

[14] F. Puoti, F. Pittorino, and M. Roveri, Quantifying Cryptocurrency Unpredictability: A Comprehensive Study of Complexity and Forecasting, vol. 1, no. 1. Association for Computing Machinery, 2025.

[15] P. Singh, D. P. Goel, and D. R. Bajaj, "A Review on Predictive Analysis of Cryptocurrencies," SSRN Electron. J., 2024, doi: 10.2139/ssrn.4848652.

[16] G. Attanasio, L. Cagliero, P. Garza, and E. Baralis, "Combining news sentiment and technical analysis to predict stock trend reversal," IEEE Int. Conf. Data Min. Work. ICDMW, vol. 2019-Novem, no. i, pp. 514–521, 2019, doi: 10.1109/ICDMW.2019.00079.

[17] Y. S. Ren, C. Q. Ma, X. L. Kong, K. Baltas, and Q. Zureigat, "Past, present, and future of the application of machine learning in cryptocurrency research," Res. Int. Bus. Financ., vol. 63, 2022, doi: 10.1016/j.ribaf.2022.101799.

[18] M. Ali and S. Shatabda, "A data selection methodology to train linear regression model to predict bitcoin price," 2020 2nd Int. Conf. Adv. Inf. Commun. Technol. ICAICT 2020, no. November, pp. 330–335, 2020, doi: 10.1109/ICAICT51780.2020.9333525.

[19] S. Ali Alahmari, "Predicting the Price of Cryptocurrency Using Support Vector Regression Methods," J. Mech. Contin. Math. Sci., vol. 15, no. 4, pp. 313–322, 2020, doi: 10.26782/jmcms.2020.04.00023.

[20] M. Iqbal, M. Iqbal, F. Jaskani, K. Iqbal, and A. Hassan, "Time-Series Prediction of Cryptocurrency Market using Machine Learning Techniques," EAI Endorsed Trans. Creat. Technol., vol. 8, no. 28, p. 170286, 2021, doi: 10.4108/eai.7-7-2021.170286.

[21] V. Saha, "Predicting Future Cryptocurrency Prices Using Machine Learning Algorithms," J. Data Anal. Inf. Process., vol. 11, no. 04, pp. 400–419, 2023, doi: 10.4236/jdaip.2023.114021.

[22] A. Gupta, R. Chopda, B. Chaudhary, P. Dwivedi, and S. Vhatkar, "A Comprehensive Study on Namecoin," Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 9, pp. 523–527, 2022, doi: 10.22214/ijraset.2022.46538.

[23] J. Lee, C. Liu, C. Ta, and C. Weng, "Towards Better Diagnosis Prediction Using Bidirectional Recurrent Neural Networks," Stud. Health Technol. Inform., vol. 290, pp. 1054–1055, 2022, doi: 10.3233/SHTI220264.

[24] G. S. Chadha, A. Panambilly, A. Schwung, and S. X. Ding, "Bidirectional deep recurrent neural networks for process fault classification," ISA Trans., vol. 106, pp. 330–342, 2020, doi: 10.1016/j.isatra.2020.07.011.

[25] R. K. Sharma and M. Casas, "Task-based Acceleration of Bidirectional Recurrent Neural Networks on Multi-core Architectures," Proc. - 2022 IEEE 36th Int. Parallel Distrib. Process. Symp. IPDPS 2022, pp. 941–951, 2022, doi: 10.1109/IPDPS53621.2022.00096.

[26] H. Apaydin, H. Feizi, M. T. Sattari, and M. S. Colak, "Comparative Analysis of Recurrent Neural Network," Water (Switzerland), vol. 12, pp. 1–18, 2020, [Online]. Available: https://www.mdpi.com/2073-4441/12/5/1500.

[27] H. Salman, J. Grover, and T. Shankar, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," vol. 2733, no. March, pp. 2709–2733, 2019, doi: 10.1162/NECO.

[28] K. Devi V, J. Mani, H. Shaker, and L. Jovanovic, "Sunspot Occurrence Forecasting With Metaheuristic Optimized Recurrent Neural Networks," Theor. Appl. Comput. Intell., vol. 1, no. 1, pp. 15–26, 2023, doi: 10.31181/taci1120231.

[29] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," Information, vol. 15, no. 9, p. 517, 2024, doi: 10.3390/info15090517.

[30] M. Ge and Y. Yuan, “Evaluation model design of project construction safety level based on bidirectional recurrent neural network (BiRNN) and bidirectional long short-term memory (BiLSTM),” PeerJ Comput. Sci., vol. 10, 2024, doi: 10.7717/peerj-cs.2351.
[31] S. Geeitha, K. P. R. Prabha, J. Cho, and S. V. Easwaramoorthy, “Bidirectional recurrent neural network approach for predicting cervical cancer recurrence and survival,” Sci. Rep., vol. 14, no. 1, pp. 1–15, 2024, doi: 10.1038/s41598-024-80472-5.
[32] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” PeerJ Comput. Sci., vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.
[33] D. C. Corresp, M. J. Warrens, G. Jurman, D. Chicco, M. J. Warrens, and G. Jurman, “Computer Science Manuscript to be reviewed The coefficient of determination R-squared is more informative than SMAPE , MAE , MAPE , MSE , and RMSE in The coefficient of determination R-squared is more informative than SMAPE , MAE , MAPE , MSE , and RMSE i,” 2021.
[34] A. Bouzid, D. Sierra-Sosa, and A. Elmaghraby, “Directional Statistics-Based Deep Metric Learning for Pedestrian Tracking and Re-Identification,” Drones, vol. 6, no. 11, pp. 1–14, 2022, doi: 10.3390/drones6110328.