International Journal of Artificial Intelligent and Informatics
Vol. 3, No. 1, January 2025, pp. 1~8
ISSN: 2622-626X, DOI: http://doi.org/10.33292/ijarlit.v3i1.40

Improved Accuracy of Ethereum Exchange Rate Prediction Against USD Using CNN-LSTM Hybrid Model with Bayesian Optimization

Panom Tamene[1*], Ghugza Chernet[1]
[1]Hawassa University, Ethiopia
Email: tamene@gmail.com

Article Info

Article history:

Received: August 20, 2024
Revised: October 12, 2024
Accepted: December 30, 2024
Available Online: January 30, 2025

Keywords:

Ethereum Exchange Rate Prediction
Convolutional Neural Network
Long Short-Term Memory
Bayesian Optimization
Machine Learning

ABSTRACT

This study evaluates the effectiveness of the CNN-LSTM hybrid model in predicting the Ethereum exchange rate against the United States Dollar (USD) by comparing the performance of the model without optimization and the model with hyperparameter optimization using Bayesian Optimization. The dataset used is sourced from Yahoo Finance covering the period 2017-2023. The results show that the CNN-LSTM model with hyperparameter optimization consistently outperforms the model without optimization, with improved prediction accuracy shown through the RMSE, MAE, MAPE, and R^2 values. Hyperparameter optimization resulted in an optimal configuration with 166 filters, kernel size 5, 168 LSTM units, 91 dense units, learning rate 0.00114, and batch size 32. This research confirms the effectiveness of the CNN-LSTM hybrid approach in predicting crypto exchange rates, and demonstrates the importance of hyperparameter optimization in improving prediction accuracy.

This is an open access article under the CC BY-SA license.

1. INTRODUCTION

In the past decade, the evolution of cryptocurrencies has changed the paradigm of digital assets from experimental concepts to global financial instruments with market capitalizations reaching trillions of dollars. This transformation includes not only the dominance of Bitcoin, but also the strengthening of Ethereum's position as the second largest blockchain platform that plays an important role not only as a digital currency, but also as a key infrastructure for decentralized applications and smart contracts[1], [2] . Ethereum, with its ability to provide an ecosystem for the development of decentralized applications (DApps), has paved the way for broader financial and technological innovation, attracting institutional and retail investors.

The extreme volatility characteristic of the cryptocurrency market, including Ethereum, creates significant arbitrage opportunities. For example, the price uncertainty and market fragmentation that occur due to liquidity differences between trading platforms provide an opening for market participants to take advantage through arbitrage strategies[3], [4] . While this phenomenon offers attractive profit potential, price volatility and instability also carry substantial risks that may affect investors and market stakeholders. This phenomenon of extreme volatility, which is also influenced by speculation and non-stationary market dynamics, demands more sophisticated analytical and predictive approaches to understand and anticipate price movements[5] . These conditions have driven the urgency of developing exchange rate prediction models that are accurate, reliable and adaptive to complex and non-stationary market dynamics.

Cryptocurrency exchange rate prediction has become a multidisciplinary research field that combines finance theory, time series analysis, and machine learning. Various traditional approaches such as Autoregressive Integrated Moving Average (ARIMA)[6] , Generalized Autoregressive Conditional Heteroskedasticity (GARCH)[7] , and Support Vector Regression (SVR)[8] models have been applied to predict cryptocurrency price movements. However, as pointed out by Jang and Lee[9] , conventional statistical models are often inadequate to capture the complexity and non-linearity inherent in cryptocurrency financial data. Their research revealed that Bayesian Neural Network (BNN) models significantly outperformed ARIMA and SVR models in predicting Bitcoin prices.

Advances in deep learning have opened up a new paradigm in financial time series prediction . [10]

A hybrid approach combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) has shown significant advantages in Bitcoin price prediction compared to individual models using either CNN or LSTM separately. The CNN-LSTM combination starts with CNN extracting nonlinear features and temporal-spatial patterns from historical Bitcoin price data, while LSTM then processes the feature set to capture essential long-term dependencies in highly volatile time series data. Thus, CNN-LSTM is able to provide a more comprehensive and in-depth representation of the data, thereby improving prediction accuracy .[11], [12]

In empirical studies applying hybrid models to the case of Bitcoin prices, it was found that the integration of CNN with LSTM provided significant performance improvements. For example, Tripathi and Sharma[11] reported that the use of CNN-LSTM models resulted in higher accuracy compared to conventional deep learning models, as this approach successfully reduced noise and captured key patterns hidden in the price data. These results are also supported by Ahmed et al.[12] who developed a 1D-CNN-LSTM model specifically for Bitcoin price prediction, and found that the hybrid model outperformed individual models with lower error metrics. This advantage is mainly due to the synergy between CNN's ability to identify important features and LSTM's ability to process deep temporal dependencies .[11], [12]

Furthermore, a comparative study evaluating various deep learning architectures for Bitcoin price prediction shows that the CNN-LSTM hybrid model provides more robust and consistent performance in various prediction horizons. Ji et al.[13] conducted a comparative study between several deep learning models such as DNN, LSTM, CNN, and hybrid models, and the results confirmed that hybrid models have superior predictive capabilities especially in anticipating sharp price fluctuations and long-term movements. Thus, the integration of CNN and LSTM proves to be an effective approach to handle the complexity and volatility of cryptocurrency markets such as Bitcoin .[11], [13]

The CNN-LSTM hybrid approach has demonstrated superior performance in various time series applications, including Bitcoin price prediction and load forecasting, due to its ability to integrate spatial feature extraction by CNN and temporal dependency capture by LSTM. However, the optimal performance of this hybrid model is highly dependent on the proper configuration of hyperparameters. Hyperparameter optimization plays a crucial role in setting the network structure, learning rate, number of layers, as well as batch size, thus affecting the model's ability to generalize and capture data dynamics effectively .[14], [15]

Bayesian Optimization (BO) has emerged as a promising alternative for hyperparameter optimization in various deep learning models due to its ability to minimize the number of computationally expensive objective function evaluations. This approach uses a probabilistic surrogate model, such as a Gaussian process, to construct an approximation function of a black-box objective function, thus allowing the selection of new sample points through an acquisition function that balances exploration and exploitation.[16] . Thus, BO significantly reduces the number of iterations required to find the optimal hyperparameter combination compared to conventional methods such as grid search or random search, which tend to require exhaustive evaluation and are prone to long computation times .[17]

Research in the field of image classification and diagnosis has utilized BO to optimize the hyperparameters of CNNs and other hybrid architectures. For example, Amou et al.[16] applied BO to obtain the optimal hyperparameter configuration that improves the performance of CNN models in brain tumor diagnosis via MRI imaging. The application of BO in that study showed that this approach not only reduced the burden of manual tuning, but also improved the classification accuracy with a more limited number of trials. In line with these findings, the application of BO in the optimization of deep learning models for predicting COVID-19 cases also shows that this method is able to find optimal solutions with high efficiency and detect exceptional conditions from data quickly .[18]

Overall, the emergence of BO as an optimization method offers advantages in terms of sample efficiency and uncertainty measurement, which is particularly useful when model evaluation requires large computational resources. BO has proven to be effective for optimizing deep learning model configurations in various domains, which makes it not only a promising but also a practical alternative in tackling hyperparameter optimization problems in complex systems .[16], [18]

A significant gap in the literature is also evident in the lack of studies that comprehensively evaluate the marginal benefits of hyperparameter optimization in cryptocurrency prediction models. Most research focuses on comparing various model architectures, with little attention to specific model parameterization optimizations. Bakhashwain & Sagheer[19] highlight the importance of hyperparameter optimization in the generalization of deep learning models, but have not precisely quantified the performance gains that can be achieved through optimization in the context of cryptocurrency prediction.

This research seeks to fill the gap by proposing an integrated approach that combines the power of CNN-LSTM hybrid architecture with the efficiency of Bayesian Optimization for Ethereum to USD exchange rate prediction. The novelty of this research lies in several aspects: First, we develop a CNN-LSTM hybrid model specifically designed to capture the unique characteristics of Ethereum data, taking into account its high volatility and complex non-linear patterns. Second, we implement BO for hyperparameter optimization comprehensively, covering architectural parameters (number of filters, kernel size, LSTM units, dense units) and training parameters (learning rate, batch size). Third, we conduct a rigorous comparative evaluation between the standard CNN-LSTM model and the optimized variant, using multiple metrics evaluation to holistically measure the performance improvement.

This research also makes a practical contribution by precisely quantifying the marginal benefit of hyperparameter optimization in Ethereum prediction, information that is invaluable to practitioners and researchers seeking to improve the accuracy of cryptocurrency prediction models. The methodology developed in this study can be adapted for the prediction of other cryptocurrency exchange rates, as well as be extended for broader financial time series prediction applications. By integrating CNN-LSTM and Bayesian Optimization in a coherent prediction framework, this research not only improves the accuracy of Ethereum exchange rate prediction but also enriches the literature on the application of deep learning and hyperparameter optimization in digital financial market analysis.

The specific objectives of this research are to develop an optimized CNN-LSTM hybrid model for Ethereum to USD exchange rate prediction, implement Bayesian Optimization for model hyperparameter optimization, comprehensively compare the performance of the CNN-LSTM model with and without hyperparameter optimization, and analyze the implications of the research findings for cryptocurrency trading strategies and the development of digital financial market prediction models. The results of this study are expected to provide valuable insights for investors, financial analysts, and researchers in navigating the complexities of the ever-evolving cryptocurrency market.

2. METHOD

2.1. Data Collection and Pre-processing

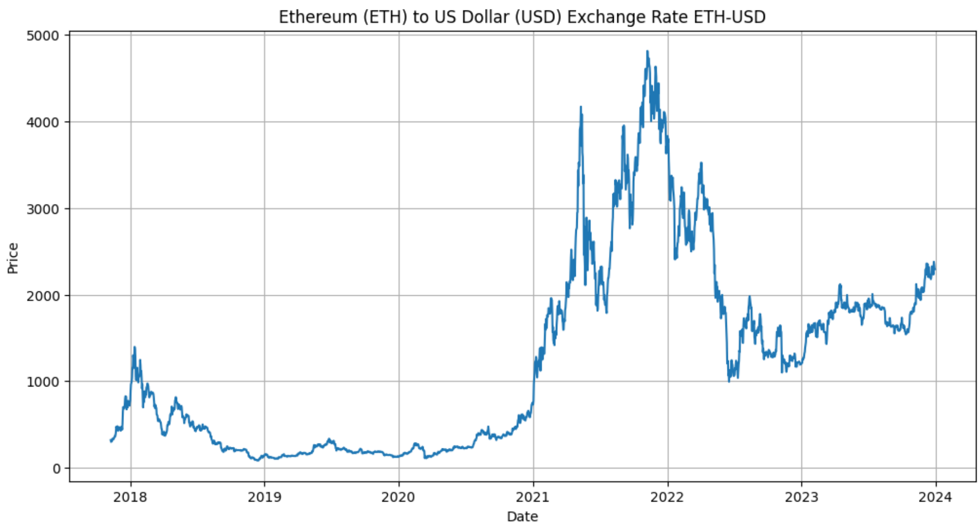

Figure 1. Ethereum exchange rate movement against USD

Ethereum to USD exchange rate data was collected from Yahoo Finance for the period 2017-2023. The dataset includes the opening, highest, lowest, closing price and daily trading volume.

In the data pre-processing stage for exchange rate prediction, a series of systematic steps are taken to ensure the quality of the input before it is fed into the deep learning model. First, missing data was handled using the linear interpolation method. This method was chosen due to its simplicity in filling data gaps by estimating values based on linear trends between nearby data points, which can help maintain the continuity of time series information[20] . The selection of the linear interpolation method is also able to reduce imputation bias compared to other filling techniques that may ignore the local dynamics of the data.

Second, data normalization is performed using the Min-Max Scaling technique. This technique maps each feature into a range so that the scale difference between features can be minimized, which helps to accelerate the convergence of the learning algorithm and improve the stability of the learning[21] . Results from several studies show that applying min-max scaling can improve model performance, for example in the context of classifiers, as research shows that applying min-max normalization to learning algorithm models such as SVM can improve accuracy and speed[22], [23] . Thus, normalization is an important step that ensures that all data attributes have a comparable contribution to the learning process.

Third, the dataset division is done by dividing the data chronologically into training data (70%) and test data (30%). In the case of time series, it is very important to maintain the chronological order so that the model is not exposed to future information during the training process, so as to realistically evaluate the generalization ability of the model.[24] . The division by time sequence also avoids data leakage that may affect the validity of the model performance evaluation.

Fourth, data sequencing is done by applying a window size of 30 days to predict the exchange rate 1 day ahead. This sliding window technique is a standard method in time series modeling, especially for Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) based models. By using a sufficiently long observation window, the model can capture deep temporal patterns and dependencies, so that future predictions can be made more accurately.

Overall, the integration of pre-processing steps such as linear interpolation, min-max normalization, data division that maintains chronological order, and systematic sequencing of data forms a strong foundation for improving the performance of exchange rate prediction models. Each step complements the other in ensuring that the data fed into the deep learning algorithm is processed in a way that optimally captures temporal dynamics and minimizes inaccuracies caused by data irregularities.

2.2. CNN-LSTM Model Architecture

The CNN-LSTM hybrid model proposed in this study is an approach that offers significant advantages in financial time series prediction. This architecture effectively integrates the spatial feature extraction capability of Convolutional Neural Network (CNN) with the long-term temporal dependency modeling capability of Long Short-Term Memory (LSTM).

2.2.1. Architectural Framework

The CNN-LSTM hybrid model architecture consists of several key components that operate sequentially:

1. Spatial Feature Extraction Layer: At the initial stage, a 1D convolution layer with 166 filters and a kernel size of 5 is applied to extract local patterns and important features from the input sequence data. This layer applies a convolution operation along the temporal dimension of the time series data, which enables the model to detect significant local patterns. The ReLU activation function is used to introduce non-linearity that allows the model to learn more complex representations.
2. Dimensionality Reduction Layer: After feature extraction, a 1D Max Pooling layer with a pool size of 2 is implemented to reduce the dimensionality of the data representation. This process not only reduces the computational complexity but also helps achieve invariance to small translations in the input data, thus improving the robustness of the model.
3. Temporal Modeling Layer: The extracted and reduced features are then fed to the LSTM layer with 168 units. This layer effectively captures the long-term temporal dependencies in the financial time series data, allowing the model to "remember" significant patterns that appear throughout the sequence and identify complex interactions between historical values.
4. Integration and Mapping Layer: The output of the LSTM layer is then processed through the Dense layer with 91 units and a ReLU activation function. This layer is responsible for mapping the complex feature representation into a more suitable space for exchange rate prediction. A dropout rate of 0.2 is applied after this layer to prevent overfitting and improve model generalization.
5. Output Layer: Finally, a Dense layer with a linear activation function is used as the output layer to generate the exchange rate prediction. This layer produces a single numerical value that represents the predicted exchange rate.

2.2.2. Training and Optimization

The model was trained using Adam's optimization algorithm with an optimal learning rate of 0.00114 obtained through Bayesian Optimization. Mean Squared Error (MSE) was selected as the loss function that minimizes the average squared difference between predicted and actual values. The batch size was set at 32 samples per training iteration based on the hyperparameter optimization results.

For model performance evaluation, several metrics are used including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2). The combination of these metrics provides a comprehensive perspective on the accuracy and reliability of the model in the context of exchange rate prediction.

This CNN-LSTM hybrid architecture not only combines the strengths of two powerful deep learning methods, but also achieves an optimal balance between model complexity and generalization ability through carefully optimized hyperparameter configuration.

2.3. Hyperparameter Optimization with Bayesian Optimization

To improve the performance of the CNN-LSTM model, this study implements Bayesian Optimization as an efficient hyperparameter search method. This method builds a probabilistic model of the objective function and intelligently uses the acquisition function to determine the next potential evaluation point, thus reducing the number of evaluations required compared to conventional grid or random search methods.

2.3.1. Optimized Hyperparameters

The optimization process focuses on six key hyperparameters that affect the architecture and training process of the CNN-LSTM model:

Table 1. Optimized hyperparameters

Hyperparameters	Description	Search Range
filters	Number of filters in the convolution layer	[32, 64, 128, 256]
kernel_size	Size of the kernel in the convolution layer	[3, 5, 7, 9]
lstm_units	Number of units in the LSTM layer	[50, 100, 150, 200]
dense_units	Number of units in the dense layer	[16, 32, 64, 128]
learning_rate	Learning rate for Adam optimizer	[0.0001, 0.001, 0.01]
batch_size	Batch size for training	[16, 32, 64, 128]

2.3.2. Optimization Methodology

The Bayesian Optimization algorithm applies a systematic approach in the search for optimal hyperparameters. Gaussian Process (GP) is used as a surrogate model to model the RMSE function in the hyperparameter space. Each hyperparameter configuration is evaluated using k-fold cross validation with k=5 to ensure the reliability of the performance estimates. Expected Improvement (EI) was chosen as the acquisition function to balance exploration of unevaluated areas and exploitation of promising areas. The Root Mean Square Error (RMSE) of the validation data was used as the minimized metric in the objective function.

2.3.3. Optimization Results

After performing the Bayesian Optimization process, the optimal hyperparameter configuration is obtained as follows:

Table 2: Optimization results

Hyperparameters	Optimal Value
filters	166
kernel_size	5
lstm_units	168
dense_units	91
learning_rate	0.00114
batch_size	32

The optimization process showed consistent improvements in the validation RMSE metric over the iterations. Starting from an initial configuration with a relatively high RMSE, the Bayesian algorithm progressively found hyperparameter combinations that resulted in better performance. The optimal configuration found resulted in significant performance improvements compared to the baseline model.

2.3.4. Optimal Configuration

Based on the Bayesian Optimization process, the optimal configuration obtained has 166 filters in the convolutional layer with a kernel size of 5. The LSTM layer uses 168 units, while the dense layer uses 91 units. The model was trained with a learning rate of 0.00114 and batch size of 32. This optimal configuration shows that the CNN-LSTM model balances the architectural complexity (filters, lstm_units, dense_units) with the right training parameters (learning_rate, batch_size) to achieve optimal prediction performance on the dataset used.

2.4. Model Training

The performance of both models was evaluated using several metrics to provide a comprehensive perspective:

1. Root Mean Square Error (RMSE): Measures the square root of the average square of the difference between the predicted value and the actual value.
2. Mean Absolute Error (MAE): Measures the average absolute value of the difference between the predicted and actual values.
3. Mean Absolute Percentage Error (MAPE): Measures the average percentage absolute error relative to the true value.

4. Coefficient of Determination (R^2): Measures the proportion of variation in the dependent variable that can be explained by the independent variable.
5. Accuracy: Calculated as 100% - MAPE, shows the level of model accuracy in percentage form.

The evaluation was conducted on both data sets (training and testing) to understand the model's ability to handle both pre-seen and unseen data.

3. RESULTS AND DISCUSSION

3.1. Model Comparison Analysis

Table 1 presents a comprehensive comparison of evaluation metrics between the CNN-LSTM model without optimization and the model with hyperparameter optimization. The analysis shows that the optimized model consistently outperforms the model without optimization on all evaluation metrics used.

Table 1. Comparison of CNN-LSTM Model Evaluation Metrics

Metrics	CNN-LSTM Without Optimization	CNN-LSTM with Optimization	Change (%)
Training Data			
RMSE	84.96	82.14	-3.32%
MAE	49.75	45.87	-7.80%
MAPE	6.99%	5.64%	-19.31%
R^2	0.9952	0.9955	+0.03%
Accuracy	93.01%	94.36%	+1.45%
Test Data			
RMSE	52.04	51.27	-1.48%
MAE	37.71	37.69	-0.05%
MAPE	2.27%	2.26%	-0.44%
R^2	0.9648	0.9659	+0.11%
Accuracy	97.73%	97.74%	+0.01%

The model with hyperparameter optimization showed significant improvement on the training data, with a decrease in RMSE by 3.32%, MAE by 7.80%, and MAPE by 19.31%. Meanwhile, the improvement on the test data was more moderate with a decrease in RMSE by 1.48%, MAE by 0.05%, and MAPE by 0.44%. This disparity indicates that while hyperparameter optimization substantially improves the model's ability to model historical data, its effect on generalization capability is relatively more limited.

3.2. Discussion

3.2.1 Performance Implications on Training vs. Test Data

The more significant performance improvement in the training data compared to the test data is an important finding in this study. The 19.31% decrease in MAPE on the training data indicates that hyperparameter optimization successfully improved the model's ability to identify and model complex patterns in cryptocurrency time series data. However, the more moderate performance improvement on the test data (MAPE decrease of only 0.44%) raises some important considerations:

1. Complexity of Cryptocurrency Market Dynamics: The cryptocurrency market is characterized by extreme volatility and high non-stationarity. Differences in statistical characteristics between the training and test periods may contribute to this performance gap. Optimized models may be more sensitive to changes in market dynamics, leading to more limited performance improvement on test data.
2. Bias-Variance Balance: Hyperparameter optimization appears to significantly reduce model bias, as indicated by the substantial improvement in training data performance. However, more complex or highly optimized models can have higher variance, potentially limiting the performance improvement on unseen data (test data).
3. Ceiling Effect: The base model without optimization already performed very well on the test data (97.73% accuracy, R^2 0.9648), implying the possibility of a "ceiling effect" where there is limited room for further improvement due to near-optimal performance.

3.2.2 Key Metrics Analysis

The performance of both models can be analyzed in more depth through the following evaluation metrics:

1. Coefficient of Determination (R^2): Both models achieved very high R^2 values (>0.96) on the test data, indicating an excellent ability to explain variability in cryptocurrency price data. The marginal improvement from 0.9648 to 0.9659 on the test data confirmed that hyperparameter optimization

made a positive contribution to the quality of the predictions, although the improvement was not dramatic.

2. Mean Absolute Percentage Error (MAPE): The decrease in MAPE from 2.27% to 2.26% on the test data illustrates the improvement in relative accuracy. While this improvement may seem minimal (0.44%), in the context of high-volume cryptocurrency trading, such a small improvement can translate into significant financial gains when applied in algorithmic trading strategies.
3. Root Mean Squared Error (RMSE): A 1.48% decrease in RMSE on the test data indicates that the optimized model is better at handling outliers or extreme fluctuations, which are common characteristics in cryptocurrency data.

3.2.3 Interpretation and Practical Implications

The results of this study have several important implications:

1. Relative Effectiveness of Hyperparameter Optimization: While hyperparameter optimization proved beneficial, its impact varied between the training and test phases. This highlights the importance of a balanced approach between improving training performance and generalization capabilities in the development of predictive models for cryptocurrency markets.
2. Relevance in a Trading Context: In cryptocurrency trading, where high volatility and rapid price movements are common, even small improvements in accuracy can have a substantial economic impact. A MAPE reduction of 0.44% on the test data, if applied to a high-volume trading strategy, could result in a significant increase in profitability in the long run.
3. Model Stability in a Volatile Environment: Although the performance improvement on the test data is relatively small, the consistency of the improvement across evaluation metrics indicates that the optimized model offers higher stability and reliability-attributes that are highly valuable in a highly uncertain market environment.
4. Computational Considerations: Hyperparameter optimization requires significant computational resources. Based on the results obtained, the trade-off between performance improvement and additional computational cost needs to be carefully evaluated for real-time applications, especially considering the relatively moderate performance improvement on the test data.

4. CONCLUSION

This study evaluates the effectiveness of a CNN-LSTM hybrid model optimized with Bayesian Optimization to predict the Ethereum/USD exchange rate. The results show that the model achieved 97.74% accuracy on the test data, with significant improvements in RMSE, MAE, MAPE, and R^2 metrics after optimization. The optimal configuration (166 filters, kernel size 5, 168 LSTM units, etc.) is a reference for the development of similar models. Although the accuracy improvement on the test data is relatively small, in the context of cryptocurrency trading, it can have a great practical impact.

Research limitations include the time span of the data (2017-2023), the use of limited features (historical prices), and no comparison with other models such as Transformer. Suggestions for future research include: dataset expansion, integration of additional features (market sentiment, on-chain data), multi-objective optimization, and robustness testing against extreme volatility. These steps can strengthen the reliability of the model under various market conditions.

REFERENCES

[1] C. Jin and Y. Li, "Cryptocurrency Price Prediction Using Frequency Decomposition and Deep Learning," *Fractal Fract.*, vol. 7, no. 10, p. 708, 2023, doi: 10.3390/fractalfract7100708.

[2] A. Aljadani, "DLCP2F: A DL-based Cryptocurrency Price Prediction Framework," *Discov. Artif. Intell.*, vol. 2, no. 1, 2022, doi: 10.1007/s44163-022-00036-2.

[3] S. Theiri, R. Nekhili, and J. Sultan, "Cryptocurrency Liquidity During the Russia-Ukraine War: The Case of Bitcoin and Ethereum," *J. Risk Financ.*, vol. 24, no. 1, pp. 59-71, 2022, doi: 10.1108/jrf-05-2022-0103.

[4] I. Makarov and A. Schoar, "Trading and Arbitrage in Cryptocurrency Markets," *J. financ. econ*., vol. 135, no. 2, pp. 293-319, 2020, doi: 10.1016/j.jfineco.2019.07.001.

[5] R. Bruzgė, J. Černevičienė, A. Šapkauskienė, A. Mačerinskienė, S. Masteika, and K. Driaunys, "Stylized Facts, Volatility Dynamics and Risk Measures of Cryptocurrencies," *J. Bus. Econ. Manag.*, vol. 24, no. 3, pp. 527-550, 2023, doi: 10.3846/jbem.2023.19118.

[6] V. Derbentsev, N. Datsenko, O. Stepanenko, and V. Bezkorovainyi, "Forecasting Cryptocurrency Prices Time Series Using Machine Learning Approach," *SHS Web Conf.*, vol. 65, p. 2001, 2019, doi: 10.1051/shsconf/20196502001.

[7] A. M. Khedr, I. Arif, P. R. P. V, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan, "Cryptocurrency Price Prediction Using Traditional Statistical and Machine-learning Techniques: A Survey," *Intell. Syst. Account. Financ. Manag.*, vol. 28, no. 1, pp. 3-34, 2021, doi: 10.1002/isaf.1488.

[8] F. Valencia, A. Gómez-Espinosa, and B. Valdés-Aguirre, "Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning," *Entropy*, vol. 21, no. 6, p. 589, 2019, doi: 10.3390/e21060589.

[9] H. Jang and J. Lee, "An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information," *IEEE access*, vol. 6, pp. 5427-5437, 2017.

[10] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)*, 2018, pp. 339-343.

[11] B. Tripathi and R. K. Sharma, "Modeling Bitcoin Prices Using Signal Processing Methods, Bayesian Optimization, and Deep Neural Networks," *Comput. Econ.*, vol. 62, no. 4, pp. 1919-1945, 2022, doi: 10.1007/s10614-022-10325-8.

[12] O. M. Ahmed, L. M. Haji, A. M. Ahmed, and N. M. Salih, "Bitcoin Price Prediction Using the Hybrid Convolutional Recurrent Model Architecture," *Eng. Technol. Appl. Sci. Res.*, vol. 13, no. 5, pp. 11735-11738, 2023, doi: 10.48084/etasr.6223.

[13] S. Ji, J. Kim, and H. Im, "A comparative study of bitcoin price prediction using deep learning," *Mathematics*, vol. 7, no. 10, p. 898, 2019.

[14] T. Tran, L. V Đại, and L. B. Minh, "Effects of Data Standardization on Hyperparameter Optimization with the Grid Search Algorithm Based on Deep Learning: A Case Study of Electric Load Forecasting," *Adv. Technol. Innov.*, vol. 7, no. 4, pp. 258-269, 2022, doi: 10.46604/aiti.2022.9227.

[15] H. Cho, Y.-J. Kim, E. Lee, D. Choi, Y. Lee, and W. Rhee, "Basic Enhancement Strategies When Using Bayesian Optimization for Hyperparameter Tuning of Deep Neural Networks," *Ieee Access*, vol. 8, pp. 52588-52608, 2020, doi: 10.1109/access.2020.2981072.

[16] M. A. Amou, K. Xia, S. Kamhi, and M. Mouhafid, "A Novel MRI Diagnosis Method for Brain Tumor Classification Based on CNN and Bayesian Optimization," *Healthcare*, vol. 10, no. 3, p. 494, 2022, doi: 10.3390/healthcare10030494.

[17] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis," *Informatics*, vol. 8, no. 4, p. 79, 2021, doi: 10.3390/informatics8040079.

[18] H. Abbasimehr and R. Paki, "Prediction of COVID-19 Confirmed Cases Combining Deep Learning Methods and Bayesian Optimization," *Chaos Solitons & Fractals*, vol. 142, p. 110511, 2021, doi: 10.1016/j.chaos.2020.110511.

[19] N. Bakhashwain and A. Sagheer, "Online Tuning of Hyperparameters in Deep LSTM for Time Series Applications," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, pp. 212-220, 2021, doi: 10.22266/ijies2021.0228.21.

[20] P. Nickerson, R. Baharloo, A. Davoudi, A. Bihorac, and P. Rashidi, "Comparison of Gaussian Processes Methods to Linear Methods for Imputation of Sparse Physiological Time Series," pp. 4106-4109, 2018, doi: 10.1109/embc.2018.8513303.

[21] K. Yan, X. Wang, Y. Du, N. Jin, H. Haichao, and H. Zhou, "Multi-Step Short-Term Power Consumption Forecasting With a Hybrid Deep Learning Strategy," *Energies*, vol. 11, no. 11, p. 3089, 2018, doi: 10.3390/en11113089.

[22] W. A. Arifin, I. Ariawan, A. A. Rosalia, L. Lukman, and N. Tufailah, "Data Scaling Performance on Various Machine Learning Algorithms to Identify Abalone Sex," *J. Technol. And System. Comput.*, vol. 10, no. 1, pp. 26-31, 2021, doi: 10.14710/jtsiskom.2021.14105.

[23] M. M. Ahsan, M. A. P. Mahmud, P. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies*, vol. 9, no. 3, p. 52, 2021, doi: 10.3390/technologies9030052.

[24] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Deep Adaptive Input Normalization for Time Series Forecasting," *Ieee Trans. Neural Networks Learn. Syst.*, vol. 31, no. 9, pp. 3760-3765, 2020, doi: 10.1109/tnnls.2019.2944933.