Journal of Information and Organizational Sciences
Volume 48, Number 2 (2024)
Journal homepage: jios.foi.hr

JIOS

DOI: 10.31341/jios.48.2.7
Open Access

UDC 004.85:336.74:004-047.72
Original Scientific Paper

Comparative Analysis of Machine Learning Techniques for Cryptocurrency Price Prediction

Sara Salehi[1*]

[1] Department of Computer Engineering, Cyprus International University, Lefkosa, North Cyprus
*Correspondence: ssalehi@ciu.edu.tr

PAPER INFO

Paper history:
Received 6 March 2024
Accepted 1 August 2024
Published 18 December 2024

Citation:
Salehi, S. (2024). Comparative Analysis of Machine Learning Techniques for Cryptocurrency Price Prediction. In Journal of Information and Organizational Sciences, vol. 48, no. 2, pp. 341-352

ABSTRACT

The significant increase in cryptocurrency trading on digital blockchain platforms has led to a growing interest in employing machine learning techniques for the effective prediction of highly nonlinear and nonstationary data, becoming increasingly popular among both individual and institutional market participants. The aim of this research is to deal with the challenging task of predicting the closing prices of two prominent cryptocurrencies, Binance Coin (BNB) and Ethereum (ETH), utilizing machine-learning techniques. This study evaluates the efficacy of various machine learning models in predicting cryptocurrency prices, with a particular focus on Support Vector Machines for Regression (SVR), least-squares Boosting (LSBoost), and Artificial Neural Networks and Adaptive Neuro-Fuzzy Inference System (ANFIS). These models are compared under various metrics. ANFIS models exhibited superior predictive performance on both training and testing datasets based on diverse performance metrics. Comparatively, SVR with a linear kernel demonstrated strong generalization capabilities, particularly on the testing set. LSBoost, while showing promise in training accuracy, indicated results with higher test errors. ANN models maintained a balance between training and testing. This comparison showed the models' effectiveness, particularly the robustness of ANFIS in capturing the volatile cryptocurrency market trends. The experimental data suggest that certain of the above models can be utilized to predict the ETH and BNB closing price in real time with promising accuracy and experimentally proven profitability.

Keywords: Machine learning, Hybrid method, Predictions, Cryptocurrency

1. Introduction

Over the past years, there has been a noticeable shift from traditional printed currency to virtual currency, leading to the emergence of cryptocurrency [1]. Cryptocurrency, a digital form of currency, utilizes intricate encryption algorithms to regulate its units. This cryptographic approach enhances the security of transactions, eliminating the need for centralized institutions like banks and enabling electronic fund transfers [2].

Since the launch of Bitcoin in 2009, cryptocurrencies have rapidly gained popularity as investment assets, despite their volatile nature. The instability in cryptocurrency values arises from various factors, including transaction costs, market sentiment, mining difficulty, alternative coin prices, user demand, and legal considerations [3]. This volatility makes the prediction of cryptocurrency prices a challenging yet critical task for investors aiming to reduce risks and optimize their portfolios [4]. Although the unpredictable nature of cryptocurrencies complicates these predictions, the increasing interest among traders, investors, and financial analysts has spurred numerous attempts to develop intelligent forecasting models. Accurate prediction models are essential for navigating the complexities of the cryptocurrency market, highlighting the ongoing efforts to understand and anticipate price movements in this dynamic field.

When considering time-series forecasting models, there are three primary categories: pure models, explanatory models, and machine learning-based approaches [5]. Pure models, exemplified by methods like Autoregressive Integrated Moving Average (ARIMA), rely solely on past data of the variable under consideration. They are most suitable for stationary and univariate time-series data [6]. In contrast, explanatory models incorporate predictor variables to forecast the target variable into the future. However, due to the nonlinear and nonstationary nature of cryptocurrency prices, making assumptions about data distributions can significantly affect forecasting accuracy. Non-stationary time-series models exhibit shifting statistical distributions over time, leading to variations in the relationship between input and output variables. Machine learning methods utilize the intrinsic non-linear and non-stationary characteristics of data, taking into account the underlying factors that affect the variable being predicted, by including explanatory features [6,7,8].

While both pure and explanatory models have their applications, they come with limitations. These include the requirement for careful examination of the stationarity of time-series data, the need to make assumptions about data distributions, dependence on historical data for accuracy, the necessity for manual selection and evaluation of functions, difficulty in recognizing long-term relationships in the presence of significant volatility, and limited utility for planning and policy-making due to lack of economic theory foundation [6, 9,10,11,12].

On the other hand, the development of machine learning methods is becoming increasingly popular for financial market forecasts, as they offer solutions to overcome the previously mentioned limitations [13]. In recent years, scholars have increasingly turned to machine learning approaches to forecast cryptocurrencies, coinciding with the rapid increase of artificial intelligence across various industries, including finance [14,15,16,17,18]. Despite the volatility of cryptocurrency prices, machine learning strategies excel at capturing long-term dependencies to identify optimal solutions that align with the available data [11]. Unlike traditional linear statistical models, machine learning techniques can capture the nonlinear aspects of cryptocurrency price volatility, as they have inherent learning capabilities [19,20]. Nonlinearity has been integrated into machine learning algorithms for predicting asset prices and returns, resulting in improved predictive accuracy [21,22].

In response to the complexities inherent in forecasting cryptocurrency prices, this study conducts an extensive analysis of state-of-the-art machine learning techniques. It compares the forecasting accuracy of several models, including SVR, LSBoost, ANN, and a hybrid method, ANFIS, focusing on Ethereum (ETH) and Binance Coin (BNB). The choice of these models is based on their unique strengths in addressing the challenges of cryptocurrency price prediction. SVR is employed for its robustness in handling high-dimensional data and capturing non-linear relationships through kernel functions. ANN is utilized for its ability to model intricate, non-linear patterns inherent in financial data. LSBoost is included for its effectiveness in improving predictive performance through iterative corrections and managing complex feature interactions. ANFIS is selected for its integration of neural network learning with fuzzy logic, which allows it to handle uncertainties and model complex, volatile market behaviors. This diverse set of models provides a comprehensive approach to evaluating forecasting accuracy in the dynamic cryptocurrency market.

The results demonstrate that ANFIS outperforms the other models in accuracy. This superiority is evident in the tests conducted with alternative cryptocurrencies, ETH and BNB, where the results were evaluated through different metrics such as RMSE, MSE, and R^2. These metrics provide a comprehensive insight into the predictive accuracy of each model, further corroborating the exceptional performance of ANFIS in the unpredictable cryptocurrency market. This finding underscores the potential of hybrid machine learning methods, particularly ANFIS, in navigating the volatility of digital asset markets and improving investment decisions.

Structured into four main sections, the paper begins with an introduction to the study's goals, followed by a detailed explanation of the methodology in Section 2. Section 3 results and discussion, showcasing the comparative effectiveness of the examined models, while the final section concludes with the implications of the findings for both researchers and practitioners in the field of financial forecasting.

2. Proposed Methodology

The approach outlined in this paper is depicted in Figure 1, illustrating: (1) data collection, (2) preprocessing data, (3) data partitioning for training and testing, (4) the problem formulation involving input-output pairs, (5) employing machine learning techniques to forecast closing prices, and (6) the evaluation metrics utilized.

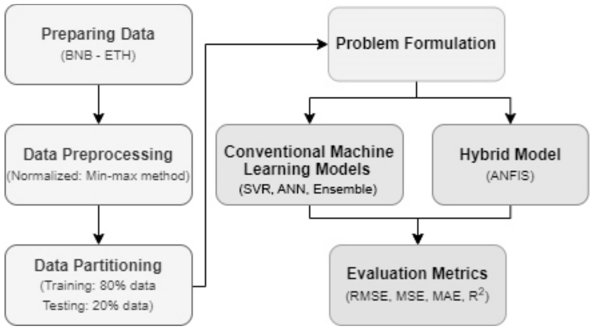

Figure 1. Overview of the suggested approach

2.1. Dataset

Data for the two most capitalized cryptocurrencies, Ethereum (ETH) and Binance Coin (BNB), were collected from coinmarketcap.com, a global portal ranked by market capitalization, providing prices and charts for the top cryptocurrencies. At the time of writing this article, these currencies were among the top five in terms of market capitalization.

This research utilizes daily historical time series data of ETH and BNB closing prices in USD, analyzing datasets for both cryptocurrencies collected from October 1, 2020, to September 30, 2023. This period covers a total of 1,095 observations for each cryptocurrency, providing a comprehensive basis for analysis. The plots of closing prices versus time for ETH and BNB are illustrated in Figures 2a and 2b, respectively.

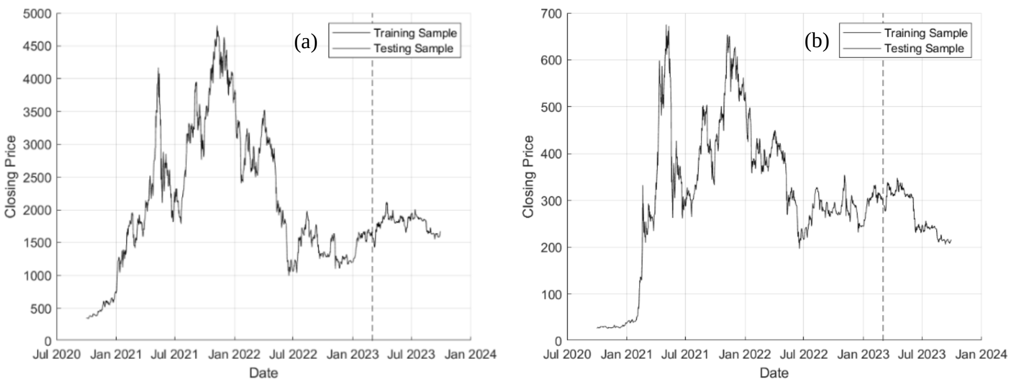

Figure 2. (a) ETH closing price (b) BNB closing price

The dataset has six features, detailed in Table 1. The table outlines characteristics of a cryptocurrency dataset, including daily average, opening, highest, lowest, and closing prices, along with the total volume traded. Each feature is described in terms of its role in representing the day's trading dynamics.

Feature	**Description**
Price	Average daily price
Open	Day's initial price
High	Peak price of the day
Low	Minimum price of the day
Close	Final price of the day
Volume	Total daily trading volume

Table 1. Features of the cryptocurrency dataset

Both ETH and BNB price series demonstrate non-stationary characteristics, as confirmed by the Augmented Dickey-Fuller (ADF) tests. The results of these tests are detailed in Table 2.

For ETH, the ADF test statistic is -2.206544, not surpassing the critical thresholds for significance levels of 1%, 5%, or 10%, indicating the series is likely non-stationary as we cannot reject the null hypothesis. The accompanying p-value of 0.203879 reinforces this conclusion, suggesting a high probability of a unit root in the series. Similarly, for BNB, the ADF test statistic of -2.364639 fails to exceed critical values,

supporting the non-rejection of the null hypothesis at conventional significance levels and implying non-stationarity. The p-value of 0.151966 further supports the likelihood of a unit root in the BNB closing price series.

	t-Statistic	p-value
ETH	-2.206544	0.203879
BNB	-2.364639	0.151966

Table 2. Augmented Dickey-Fuller test statistic

Following the initial data preparation, the subsequent action entails normalizing the data. This process modifies the data values to conform to a designated range. Utilizing the min-max normalization approach, the data is calibrated to align within a range from 0 to 1, ensuring uniformity and comparability across the dataset. Where y denotes a single point of a feature and Y the feature's vector.

$$y_{rescaled} = \frac{y - \min(Y)}{\max(Y) - \min(Y)}$$

After normalization, the data is divided into training and testing datasets, the first 80% of the collected data allocated for training, and the remaining 20% set aside for testing. The division of the data into training and testing samples is depicted in Figures 2a for ETH and 2b for BNB.

2.2. Problem Formulation

This article focuses on utilizing historical cryptocurrency data, specifically closing price records, to predict future values. The dataset comprises sequential price points, represented as $\{p_0, p_1, p_2, \cdots, p_n\}$, where each p_i corresponds to the price at timestamp i. We define an input window of length w, constructing input vectors v and output o as follows:

$$v = [p_{i-w+1}, p_{i-w+2}, p_{i-w+3}, \cdots, p_{i-1}, p_i]$$

$$o = [p_{i+1}]$$

The objective is to forecast the value of p_{i+1} using an input vector containing past values. The data is organized into pairs of input-output as shown above. For the models discussed in this article, we consider a prediction window of 3 days.

2.3. Conventional machine learning models

In this paper, Support Vector Machine for Regression (SVR), Artificial Neural Networks (ANN), and least-squares Boosting (LSBoost) are employed for analysis. Furthermore, K-fold validation is utilized to ensure the robustness and applicability of the models.

To prevent overfitting, this study adopts K-fold cross-validation. This method divides the training dataset D into, K equal parts, using $K - 1$ parts for training and the remaining part for validation. This cycle repeats K times, each with a different validation set, to produce K models. The final model's prediction is the average of these K models' predictions. In this study, the number of folds in the K-fold cross-validation is set to 5 since it is widely used in practice.

2.3.1. Support Vector Machines for regression (SVR)

Support vector machine (SVM) is a supervised machine-learning technique. The pioneer of SVM was Varpik [23,24]. Initially focused on binary classification, SVM can be extended to regression tasks, known as Support Vector Regression (SVR) [25,26]. In SVR, a kernel function is applied to transform the input data into a higher-dimensional space. This transformation allows SVR to develop a hyperplane that best fits the input vectors to the desired output values [27]. The main idea of SVR is to map data into a high-dimensional feature space. It uses Vapnik-Chervonenkis Dimension theory as its operational foundation to produce an optimal hyperplane that converts a low-dimensional input vector to a higher-dimensional feature space, ensuring high generalization capability [28,29].

Since the challenge of applying an SVR algorithm is selecting the proper kernel function, Linear and Gaussian (Radial Basis Function (RBF)) kernels were applied. The equations for each of the kernels in the SVR algorithm are listed in Table 3.

SVR	Kernel	Description
Linear	$K(x_1, x_2) = x_1^T x_2$	Two class learning.
Gaussian	$K(x_1, x_2) = \exp\left(\frac{-\parallel x_1 - x_2 \parallel^2}{2\sigma^2}\right)$	One-class & multi-class learning.

K: Kernel function
x_1, x_2: Input vectors
σ: Width of the kernel

Table 3. SVR types

2.3.2. Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) are a highly valuable method for nonlinear data modeling within Artificial Intelligence, integrating mathematics and computations [30,31]. ANNs simulate biological neurons [32,33] through an adaptive system that acquires knowledge using artificial neurons arranged in a multi-layered structure. The fundamental components of an ANN include several processing elements across multiple layers [34]. By utilizing interconnected processing elements (artificial neurons), ANNs process system inputs and generate outputs by applying specific nonlinear functions [32].

The network's artificial neurons are organized into three main layers: input, hidden, and output. In the first layer, each input a_i is assigned a weight w_i, and each artificial neuron receives a bias b_i, treated as an additional input. The effectiveness of a_i is determined by its weight w_i; a higher value indicates a greater impact on the system. The ANN computes the values of weights to establish relationships between input and output data [35]. An activation function computes the sum of the weighted inputs to produce the output [36].

This paper concentrates on the application of a feedforward neural network for regression tasks, utilizing a Multilayer Perceptron (MLP) model. The MLP is a kind of feedforward neural network characterized by its layered structure, where neurons are arranged in multiple layers. Within this architecture, each neuron in a given layer is fully connected to all neurons in the next layer, creating a densely connected network.

2.3.3. Regression Tree Ensembles

Regression tree ensembles are predictive models that are formed by combining multiple individual regression trees, each weighted differently. Least-squares boosting (LSBoost) is one of these ensemble techniques, primarily aimed at minimizing the mean squared error [37]. LSBoost is an ensemble-learning boosting algorithm designed specifically for estimating continuous target variables. In each iteration, it trains a new learner to capture the discrepancy between the observed target values and the aggregated predictions from all previously trained learners [38]. The final output of LSBoost is derived from the combination of predictions across numerous base learners. These base learners are iteratively trained to address the residual error not accounted for by the outputs from preceding learners [39].

2.4. Hybrid Machine Learning Model

ANFIS, a hybrid machine learning methodology that combines the robustness of fuzzy logic with the computational prowess of Artificial Neural Networks (ANN), was introduced by Jang [40]. This integration utilizes the strengths of both domains to enhance the model's ability to: i) raise its capacity for learning the complexities inherent in processes [41], ii) achieve an optimal relationship between the inputs and outputs of a system, iii) approximates the nonlinear functions, with learning capability [42], iv) provide precise mapping between predictor and dependent variables, facilitating accurate predictions [43]. ANFIS has been employed to solve different real-world problems, including time series forecasting and simulation of engineering processes [41,44,45].

2.4.1. Adaptive Neuro-Fuzzy Inference System (ANFIS)

The primary ANFIS structure comprises Takagi–Sugeno (TKS) inference (or Sugeno first-order FIS) rule-based learning algorithms, and the "IF-THEN rules" are used for generating mapping inputs and outputs mapping. A typical architecture of ANFIS is shown in Figure 3, which consists of five layers. For simplicity, let's assume the structure with two inputs as x_1 and x_2. Then, the rules R_a and R_b in FIS stated as:

$$R_a: IF\ \ x_1\ \ is\ \ \widetilde{M}_1\ and\ \ x_2\ is\ \widetilde{N}_1\ ,\text{THEN}\ \ g_1(x_1,x_2)\ = p_1x_1 + q_1x_2 + r_1$$

$$R_b: IF\ \ x_1\ \ is\ \ \widetilde{M}_2\ and\ \ x_2\ is\ \widetilde{N}_2\ ,\text{THEN}\ \ g_2(x_1,x_2)\ = p_2x_1 + q_2x_2 + r_2$$

where $\widetilde{M}$ and $\widetilde{N}$ are fuzzy sets for x_1 and x_2, respectively. The function $g_i(x_1,x_2)$ which is the output of the system is the first order polynomial with parameters p_i, q_i, r_i $(i\ =\ 1,2)$.

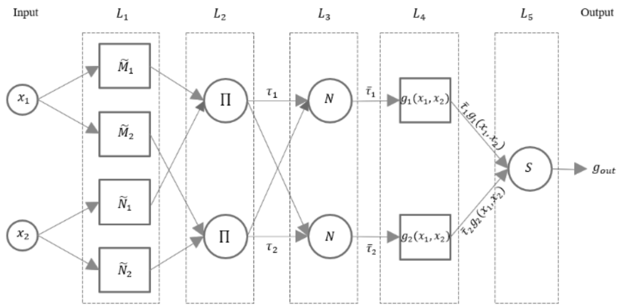

Figure 3. ANFIS structure

ANFIS structure has adaptive (square) and fixed (circle) types of nodes. The output of the *i*th node is defined as $O_{j,i}$. In ANFIS's structure, each layer is described as follows:

L_1 – Fuzzy layer: Fuzzification of inputs, x_1 and x_2, are done in the first layer. The output of the first layer is computed as:

$$O_{1,i} = \mu_{\widetilde{M}_i(x_1)}, \qquad i = 1,2 \tag{1}$$

$$O_{1,i} = \mu_{\widetilde{N}_{i-2}(x_2)}, \qquad i = 3,4 \tag{2}$$

where $\mu_{\widetilde{M}_i(x_1)}$ and $\mu_{\widetilde{N}_i(x_2)}$ are the membership functions of the fuzzy sets $\widetilde{M}_i$ and $\widetilde{N}_i$, respectively.

The inputs are fuzzified by using the triangle membership function described below, the MATLAB function `y=trimf(x,params)` returns fuzzy membership values:

$$f(x;\ \alpha_1,\alpha_2,\alpha_3) = \begin{cases} 0, & x \le \alpha_1, \\ \dfrac{x-\alpha_1}{\alpha_2-\alpha_1}, & \alpha_1 \le x \le \alpha_2, \\ \dfrac{\alpha_3-x}{\alpha_3-\alpha_2}, & \alpha_2 \le x \le \alpha_3, \\ 0, & \alpha_3 \le x \end{cases} \tag{3}$$

where $\alpha\ =\ (\alpha_1,\alpha_2,\alpha_3)$ is a triangular fuzzy number.

L_2 – Product layer: The nodes in this layer are fixed ones. The operator is a product marked with (II) that computes the trigger force of a rule in the ANFIS structure. The output of L_2 is computed as:

$$O_{2,i} = \tau_i = \mu_{\widetilde{M}_i(x_1)} \times \mu_{\widetilde{N}_i(x_2)}, \qquad i = 1,2 \tag{4}$$

L_3 – Normalized layer: By the nodes marked with (N) in Figure 3, the outputs in L_2 are normalized. The output of L_3 is computed as:

$$O_{3,i} = \bar{\tau}_i = \frac{\tau_i}{\sum_{i=2}^{2} \tau_i}, \qquad i = 1,2 \tag{5}$$

Also, the nodes in this layer are fixed ones.

L_4 – Defuzzification layer: The output of L_4 is calculated by the multiplication of a first-order polynomial, $g_i(x_1, x_2)$, and normalized firing strength, $\bar{\tau}_i$, as follows:

$$O_{4,i} = \bar{\tau}_i \times g_i(x_1, x_2) = \bar{\tau}_i \times (p_i x_1 + q_i x_2 + r_i), \qquad i = 1,2 \tag{6}$$

The nodes are adaptive in L_4.

L_5 – Total output layer: In the last layer, there is only one fixed node identified as S, which is the output (cumulative sum, Σ) layer. The output of L_5 is generated by:

$$O_{5,i} = g_{out} = \sum_{i=1}^{2} \bar{\tau}_i g_i(x_1, x_2), \qquad i = 1,2 \tag{7}$$

The final output in the ANFIS is a linear combination of consequent parameters.

2.5. Evaluation Performance Metrics

The effectiveness, precision, and reliability of the predictive models are assessed using metrics such as root mean square error (RMSE), mean squared error (MSE), mean absolute error (MAE), and coefficient of determination (R^2). The formulas for these metrics are as follows:

$$RMSE = \left[\frac{1}{N}\sum_{k=1}^{N}\left(Y_k^{pr.} - Y_k^{exp.}\right)^2\right]^{0.5} \tag{8}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left(Y_k^{pr.} - Y_k^{exp.}\right)^2 \tag{9}$$

$$MAE = \frac{1}{N}\sum_{k=1}^{N}\left|Y_k^{pr.} - Y_k^{exp.}\right| \tag{10}$$

where $Y_k^{pr.}$ is the prediction value, and $Y_k^{exp.}$ is the experimental value of kth testing data to be defined from the model, and N is the number of data used in testing.

$$\mathrm{R}^2 = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}} \tag{11}$$

Where RSS is the sum of squares of residuals. TSS is the total sum of squares, which measures the total variance in the observed data.

3. Results and Discussion

This study emphasizes the development, improvement, and comparative analysis of various machine learning models for cryptocurrency price prediction, with a special focus on deploying the ANFIS model. In the MATLAB environment, customized code for the development of the techniques has been constructed, including SVR with Linear and Gaussian kernels, ANN in the form of MLP, LSBoost, and the ANFIS model. These models offer a comprehensive toolkit for predictive modeling, each with unique capabilities to model complex data relationships.

SVRs, with their different kernels, offer the flexibility to model linear and non-linear relationships in the data. The linear kernel is typically preferred for linearly separable data, enabling straightforward decision boundaries, whereas the Gaussian kernel is suitable for complex, nonlinear data structures, allowing the model to capture intricate patterns. ANNs, particularly MLPs, enabling the model to learn deep representations of the data through layers of neurons and non-linear activation functions. This makes MLPs highly adaptable to various data types and complexities. the LSBoost algorithm enhances prediction accuracy by utilizing the concept of boosting with ensemble regression trees. LSBoost combines multiple weak regression tree models to form a strong predictor, systematically reducing bias and variance by focusing on difficult-to-predict instances. This ensemble method is particularly effective for regression tasks, offering robustness against overfitting and improving predictive performance across diverse datasets. The finding of this paper for SVR, LSBoost and ANN are summarized in Tables 4 and 5 for ETH and BNB, respectively.

ANFIS model combined fuzzy inference systems and neural network methods to enhance the fuzzy inference system structure, resolve deficiencies in neural networks, and increase accuracy and computation speed [46]. To create the fuzzy inference system (FIS) model, the Sugeno-type is employed in the MATLAB neuro-fuzzy toolbox. The FIS was created using the grid partition algorithm. When there are fewer than six input variables, the problem can be solved by using the grid partition algorithm appropriately [47]. For ANFIS learning, a hybrid learning algorithm that makes use of the least square and the gradient method was used. The membership function determines how well each point in the input space is mapped to an acceptable membership value between 0 and 1. The trimf membership function was selected for its simplicity and popularity among scholars [48,49,50]. The ANFIS dataset was trained at 100 epoch iterations with an error tolerance of zero. The finding is summarized in Table 6.

Based on the findings presented in Tables 4, 5, and 6, the following observations are made:

The SVR method demonstrated strong performance, with the linear kernel generally outperforming the Gaussian kernel in terms of lower errors (RMSE, MSE, MAE) and higher (R^2) values across both training and testing phases. This indicates SVR's robustness in linearly separable data scenarios and its capability in generalizing predictions.

Method		**Data type**	**RMSE**	**MSE**	**MAE**	**R^2**
SVR	linear	train	0.0256	0.0007	0.0170	0.9888
		test	0.0091	0.0001	0.0062	0.8992
	Gaussian	train	0.0295	0.0009	0.0204	0.9851
		test	0.0103	0.0001	0.0077	0.8702
LSBoost		train	0.0335	0.0011	0.0243	0.9807
		test	0.0179	0.0003	0.0159	0.6039
ANN		train	0.0282	0.0008	0.0191	0.9864
		test	0.0137	0.0002	0.0107	0.7698

Table 4. Comparative performance of machine learning methods on ETH

Method		**Data type**	**RMSE**	**MSE**	**MAE**	**R^2**
SVR	linear	train	0.0285	0.0008	0.0183	0.9859
		test	0.0099	0.0001	0.0071	0.9783
	Gaussian	train	0.0329	0.0011	0.0202	0.9811
		test	0.0108	0.0001	0.0078	0.9744
LSBoost		train	0.0364	0.0013	0.0254	0.9769
		test	0.0178	0.0003	0.0146	0.9302
ANN		train	0.0346	0.0012	0.0199	0.9791
		test	0.0122	0.0001	0.0093	0.9672

Table 5. Comparative performance of machine learning methods on BNB

The LSBoost method exhibited higher error rates on the test data compared to SVR and ANN. However, its performance varied, with relatively competitive R^2 values, indicating that while it may not consistently outperform other models in precision, it retains some predictive reliability.

The ANN showed balanced performance with a notable capacity for generalization, as indicated by the performance metrics across training and testing datasets. This balance suggests ANN's versatility and potential as a reliable predictive model for cryptocurrency prices.

The ANFIS outperformed conventional models in several key metrics, particularly in achieving lower RMSE and higher R^2 values in the test phase, which highlights its superior generalization capability. The ANFIS model's integration of fuzzy logic with neural networking allows it to effectively capture and model the complex, non-linear relationships typical of cryptocurrency price movements. Figure 4 and 5 comparing predicted and actual cryptocurrency prices for ETH and BNB, respectively. The Figures demonstrate the models' strong fit to the training and testing data suggesting that the model has successfully captured the historical price movements.

Method	Data type	Dataset	RMSE	MSE	MAE	R^2
ANFIS	train	ETH	0.0248	0.0006	0.0163	0.9898
	test	ETH	0.0090	0.0001	0.0062	0.9027
ANFIS	train	BNB	0.0259	0.0007	0.0164	0.9872
	test	BNB	0.0104	0.0001	0.0076	0.9811

Generate FIS: Grid Partition
Train FIS: Hybrid
Function: Constant
Epochs:100

Table 6. Performance of the ANFIS method on ETH and BNB datasets

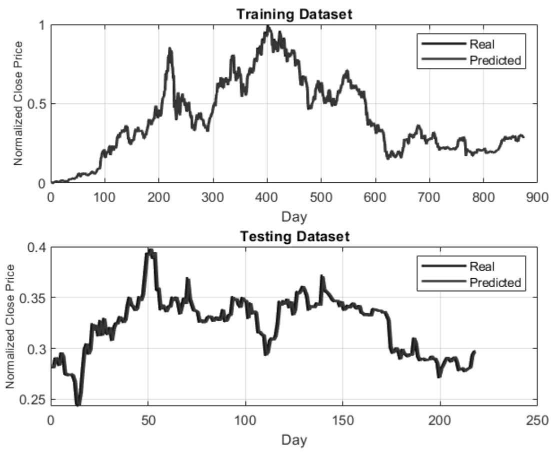

Figure 4. ANFIS model evaluation: real vs. predicted ETH prices in training and testing phases

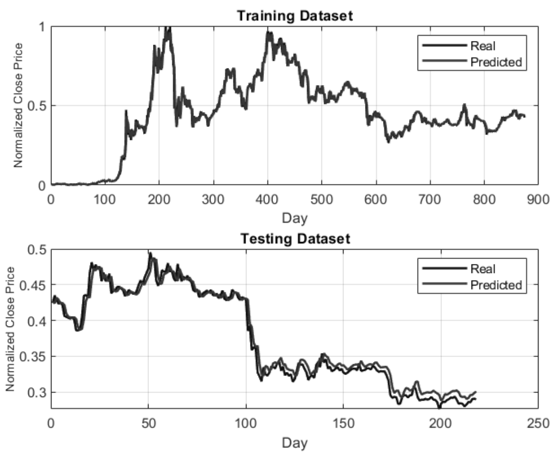

Figure 5. ANFIS model evaluation: real vs. predicted BNB prices in training and testing phases

The analysis indicates that while conventional approaches such as SVR and ANN demonstrate significant predictive power, the hybrid ANFIS model excels in terms of its remarkable accuracy and ability to generalize across various datasets. ANFIS's effectiveness stems from its intricate design, which combines the advantages of fuzzy inference systems and neural networks. This unique architecture makes ANFIS especially suitable for navigating the volatile and unpredictable dynamics of cryptocurrency markets.

The higher test errors observed in LSBoost compared to other models underscore the importance of model selection and parameter optimization in achieving optimal performance. LSBoost's variability in effectiveness suggests that ensemble methods require careful tuning to balance the bias-variance trade-off inherent in predictive modeling.

4. Conclusion

In the dynamic and volatile world of cryptocurrencies, predicting the closing prices of assets is a complex and challenging task. This research set out to address this challenge by applying a various machine learning method, such as SVR, ANN, LSBoost, and ANFIS, to predict the closing prices of BNB and ETH cryptocurrencies. The objective was to compare the accuracy of these models and determine which ones were the most effective for cryptocurrency price prediction.

Based on the findings, this research highlights the effectiveness of hybrid methodologies, particularly ANFIS, in enhancing predictive precision and model generalization, especially within the domain of cryptocurrency price forecasting. The results underscore the importance of employing sophisticated modeling approaches capable of navigating the complex dynamics inherent in financial markets. Moreover, the proven effectiveness of these models indicates a promising opportunity to expand the application of these techniques to additional cryptocurrencies.

References

[1] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies*: a comprehensive introduction. Princeton University Press, 2016.

[2] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[3] Y. Sovbetov, "Factors influencing cryptocurrency prices: Evidence from bitcoin, ethereum, dash, litcoin, and monero," *Journal of Economics and Financial Analysis*, vol. 2, no. 2, pp. 1–27, 2018.

[4] M. M. Patel, S. Tanwar, R. Gupta, and N. Kumar, "A deep learning-based cryptocurrency price prediction scheme for financial institutions," *Journal of information security and applications*, vol. 55, p. 102583, 2020.

[5] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[6] M. Mudassir, S. Bennbaia, D. Unal, and M. Hammoudeh, "Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach," *Neural computing and applications*, pp. 1–15, 2020.

[7] C. Shi and Y. Wang, "Non-parametric machine learning methods for interpolation of spatially varying non-stationary and non-gaussian geotechnical properties," *Geoscience Frontiers*, vol. 12, no. 1, pp. 339–350, 2021.

[8] W. Hao, X. Sun, C. Wang, H. Chen, and L. Huang, "A hybrid EMD-LSTM model for non-stationary wave prediction in offshore China," *Ocean Engineering*, vol. 246, p. 110566, 2022.

[9] C.-H. Wang and L.-C. Hsu, "Constructing and applying an improved fuzzy time series model: Taking the tourism industry for example," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2732–2738, 2008.

[10] J. Shahrabi, E. Hadavandi, and S. Asadi, "Developing a hybrid intelligent model for forecasting problems: Case study of tourism demand time series," *Knowledge-Based Systems*, vol. 43, pp. 112–122, 2013.

[11] T. H. Aldhyani and H. Alkahtani, "A bidirectional long short-term memory model algorithm for predicting covid-19 in gulf countries," *Life*, vol. 11, no. 11, p. 1118, 2021.

[12] B. Peng, H. Song, and G. I. Crouch, "A meta-analysis of international tourism demand forecasting and implications for practice," *Tourism Management*, vol. 45, pp. 181–193, 2014.

[13] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European journal of operational research*, vol. 270, no. 2, pp. 654–669, 2018.

[14] B. M. Henrique, V. A. Sobreiro, and H. Kimura, “Literature review: Machine learning techniques applied to financial market prediction,” *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[15] M. Leippold, Q. Wang, and W. Zhou, “Machine learning in the Chinese stock market,” *Journal of Financial Economics*, vol. 145, no. 2, pp. 64–82, 2022.

[16] N. Nazareth and Y. V. R. Reddy, “Financial applications of machine learning: A literature review,” *Expert Systems with Applications*, vol. 219, p. 119640, 2023.

[17] M. N. Ashtiani and B. Raahemi, “News-based intelligent prediction of financial markets using text mining and machine learning: A systematic literature review,” *Expert Systems with Applications*, vol. 217, p. 119509, 2023.

[18] S. Aziz, M. Dowling, H. Hammami, and A. Piepenbrink, “Machine learning in finance: A topic modeling approach,” *European Financial Management*, vol. 28, no. 3, pp. 744–770, 2022.

[19] V. D’Amato, S. Levantesi, and G. Piscopo, “Deep learning in predicting cryptocurrency volatility,” *Physica A: Statistical Mechanics and its Applications*, vol. 596, p. 127158, 2022.

[20] M. D. Lal and R. Varadarajan, “A review of machine learning approaches in synchrophasor technology,” *IEEE Access*, 2023.

[21] M. A. Ammer and T. H. Aldhyani, “Deep learning algorithm to predict cryptocurrency fluctuation prices: Increasing investment awareness,” *Electronics*, vol. 11, no. 15, p. 2349, 2022.

[22] 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)*. IEEE, 2018, pp. 339–343.

[23] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[24] V. Vapnik and V. Vapnik, “Statistical learning theory wiley,” *New York*, vol. 1, no. 624, p. 2, 1998.

[25] N. A. Almansour, H. F. Syed, N. R. Khayat, R. K. Altheeb, R. E. Juri, J. Alhiyafi, S. Alrashed, and S. O. Olatunji, “Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study,” *Computers in biology and medicine*, vol. 109, pp. 101–111, 2019.

[26] X. Pan, X. Pang, H. Wang, and Y. Xu, “*A safe screening based framework for support vector regression*,” Neurocomputing, vol. 287, pp. 163–172, 2018.

[27] M. Wauters and M. Vanhoucke, “Support vector machine regression for project control forecasting,” *Automation in Construction*, vol. 47, pp. 92–106, 2014.

[28] D. A. Otchere, T. O. A. Ganat, R. Gholami, and S. Ridha, “Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models,” *Journal of Petroleum Science and Engineering*, vol. 200, p. 108182, 2021.

[29] F. Anifowose, J. Labadin, and A. Abdulraheem, “Improving the prediction of petroleum reservoir characterization with a stacked generalization ensemble model of support vector machines,” *Applied Soft Computing*, vol. 26, pp. 483–496, 2015.

[30] C. C. Obi, J. T. Nwabanne, C. A. Igwegbe, P. E. Ohale, and C. O. R. Okpala, “Multi-characteristic optimization and modeling analysis of electrocoagulation treatment of abattoir wastewater using iron electrode pairs,” *Journal of Water Process Engineering*, vol. 49, p. 103136, 2022.

[31] M. T. Gaudio, G. Coppola, L. Zangari, S. Curcio, S. Greco, and S. Chakraborty, “Artificial intelligence-based optimization of industrial membrane processes,” *Earth Systems and Environment*, vol. 5, no. 2, pp. 385–398, 2021.

[32] J. K. Chaudhary et al., “Hybridization of ANFIS and fuzzy logic for groundwater quality assessment,” *Groundwater for Sustainable Development*, vol. 18, p. 100777, 2022.

[33] J. Peng, G. Yan, Y. Zandi, A. S. Agdas, T. Pourrostam, I. E. El-Arab, N. Denic, Z. Nesic, B. Cirkovic, and M. A. Khadimallah, “Prediction and optimization of the flexural behavior of corroded concrete beams using adaptive neuro fuzzy inference system,” in *Structures*, vol. 43. Elsevier, 2022, pp. 200–208.

[34] M. Hema, D. Toghraie, and F. Amoozad, “Prediction of viscosity of MWCNT-AL2O3 (20: 80)/SAE40 nano-lubricant using multi-layer artificial neural network (MLP-ANN) modeling,” *Engineering Applications of Artificial Intelligence*, vol. 121, p. 105948, 2023.

[35] K. Linka and E. Kuhl, “A new family of constitutive artificial neural networks towards automated model discovery,” *Computer Methods in Applied Mechanics and Engineering*, vol. 403, p. 115731, 2023.

[36] L. Feng, K. Zhong, J. Liu, and A. Ghanbari, “Applying supervised intelligent scenarios to numerical investigate carbon dioxide capture using nanofluids,” *Journal of Cleaner Production*, vol. 381, p. 135088, 2022.

[37] Y. Zhang and X. Xu, “Modulus of elasticity predictions through LSBoost for concrete of normal and high strength,” *Materials Chemistry and Physics*, vol. 283, p. 126007, 2022.

[38] Y. Zhang and X. Xu, “Solid particle erosion rate predictions through LSBoost,” *Powder Technology*, vol. 388, pp. 517–525, 2021.

[39] G. C. Gutiérrez -Tobal, D. Alvarez, F. Vaquerizo-Villar, A. Crespo, L. Kheirandish-Gozal, D. Gozal, F. del Campo, and R. Hornero, “Ensemble learning regression to estimate sleep apnea severity using at-home oximetry in adults,” *Applied Soft Computing*, vol. 111, p. 107827, 2021.

[40] J.-S. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE transactions on systems, man, and cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.

[41] S. Shahid, S. K. Nath, and A. Maksud Kamal, “GIS integration of remote sensing and topographic data using fuzzy logic for ground water assessment in Midnapur District, India,” *Geocarto International*, vol. 17, no. 3, pp. 69–74, 2002.

[42] A. Abraham, “Adaptation of fuzzy inference system using neural learning,” in *Fuzzy Systems Engineering: Theory and Practice*. Springer, 2005, pp. 53–83.

[43] R. Devaraj, S. K. Mahalingam, B. Esakki, A. Astarita, and S. Mirjalili, “A hybrid GA-ANFIS and F-Race tuned harmony search algorithm for Multi-Response optimization of non-traditional machining process,” *Expert Systems with Applications*, vol. 199, p. 116965, 2022.

[44] M. A. Al-qaness, A. A. Ewees, H. Fan, L. Abualigah, and M. Abd Elaziz, “Boosted ANFIS model using augmented marine predator algorithm with mutation operators for wind power forecasting,” *Applied Energy*, vol. 314, p. 118851, 2022.

[45] S. Salehi, “Employing a time series forecasting model for tourism demand using ANFIS,” *Journal of Information and Organizational Sciences*, vol. 46, no. 1, pp. 157–172, 2022.

[46] Y. Mehri, M. Nasrabadi, and M. H. Omid, “Prediction of suspended sediment distributions using data mining algorithms,” *Ain Shams Engineering Journal*, vol. 12, no. 4, pp. 3439–3450, 2021.

[47] A. M. Abdulshahed, A. P. Longstaff, and S. Fletcher, “The application of ANFIS prediction models for thermal error compensation on cnc machine tools,” *Applied soft computing*, vol. 27, pp. 158–168, 2015.

[48] I. S. Ike, C. O. Asadu, C. A. Ezema, T. O. Onah, N. O. Ogbodo, E. U. Godwin-Nwakwasi, and C. E. Onu, “ANN-GA, ANFIS-GA and thermodynamics base modeling of crude oil removal from surface water using organic acid grafted banana pseudo stem fiber,” *Applied Surface Science Advances*, vol. 9, p. 100259, 2022.

[49] J. Li, G. Yan, L. H. Abbud, T. Alkhalifah, F. Alturise, M. A. Khadimallah, and R. Marzouki, “Predicting the shear strength of concrete beam through ANFIS-GA–PSO hybrid modeling,” *Advances in Engineering Software*, vol. 181, p. 103475, 2023.

[50] C. E. Onu, P. K. Igbokwe, J. T. Nwabanne, and P. E. Ohale, “ANFIS, ANN, and RSM modeling of moisture content reduction of cocoyam slices,” *Journal of Food Processing and Preservation*, vol. 46, no. 1, p. e16032, 2022.