Accounting 11 (2025) ****_****

Contents lists available at GrowingScience

Accounting

homepage: www.GrowingScience.com/ac/ac.html

Using artificial intelligence techniques and econometrics model for crypto-price prediction

Abhidha Verma[a] and Jeewesh Jha[b*]

[a]*Bachelor Of Science in Accounting, University Of Southern California, LA, California, USA*
[b]*Post Graduate Diploma In Artificial Intelligence & Deep Learning, IIT Ropar, India*

C H R O N I C L E

Article history:
Received January 3, 2025
Received in revised format February 25 2025
Accepted March 23 2025
Available online March 23 2025

Keywords:
Cryptocurrency Artificial Intelligence Optimization Algorithm Econometric Methods Ethereum Price

A B S T R A C T

In today's financial landscape, individuals face challenges when it comes to determining the most effective investment strategies. Cryptocurrencies have emerged as a recent and enticing option for investment. This paper focuses on forecasting the price of Ethereum using two distinct methods: artificial intelligence (AI)-based methods like Genetic Algorithms (GA), and econometric models such as regression analysis and time series models. The study incorporates economic indicators such as Crude Oil Prices and the Federal Funds Effective Rate, as well as global indices like the Dow Jones Industrial Average and Standard and Poor's 500, as input variables for prediction. To achieve accurate predictions for Ethereum's price one day ahead, we develop a hybrid algorithm combining Genetic Algorithms (GA) and Artificial Neural Networks (ANN). Furthermore, regression analysis serves as an additional prediction tool. Additionally, we employ the Autoregressive Moving Average (ARMA) model to assess the relationships between variables (dependent and independent variables). To evaluate the performance of our chosen methods, we utilize daily historical data encompassing economic and global indices from the beginning of 2019 until the end of 2021. The results demonstrate the superiority of AI-based approaches over econometric methods in terms of predictability, as evidenced by lower loss functions and increased accuracy. Moreover, our findings suggest that the AI approach enhances computational speed while maintaining accuracy and minimizing errors.

1. Introduction

The rising popularity of cryptocurrencies has garnered significant interest from individuals worldwide. One crucial factor contributing to the increasing attractiveness of the cryptocurrency market is its improved market capitalization. Market capitalization (market cap) refers to the total value of all mined coins and can be calculated by multiplying the number of total coins by the current market price. Statistics and evidence indicate a substantial surge in cryptocurrency market cap, growing from approximately 1.5 billion dollars in April 2013 to 1.5 trillion dollars in February 2022 (Coinmarketcap.com). For instance, developing countries in Africa, such as Nigeria and Kenya, exhibit a notable inclination to invest in cryptocurrencies. These countries rank second and eighth globally in terms of BTC volume, with 60,215 and 8,895 respectively. The increasing global acceptance of cryptocurrencies has spurred extensive research in this field.

In Iran, cryptocurrencies have gained a significant user base and advocates who utilize them for various purchases, including automobiles and houses. A study conducted by the blockchain association reveals that daily transactions in this market in Iran amount to at least 10 million dollars.

Various methods are available for forecasting cryptocurrency prices, including artificial intelligence (AI) based methods like machine learning (ML), econometric models such as linear regression or auto-regressive integrated moving average (ARIMA) analysis, and mathematical models like statistical, empirical, or simulation methods. One crucial aspect of accurate prediction

* Corresponding author. Tel.: +91 70440 56613
E-mail address: jeewesh.jha@bajajfinserv.in (J. Jha)

ISSN 2369-7407 (Online) - ISSN 2369-7393 (Print)

doi: 10.5267/j.ac.2025.3.003

and estimating prediction equations is the selection of appropriate input variables. Numerous economic and financial indicators can influence and impact cryptocurrency prices. By identifying and incorporating these indicators into the model, we can enhance its accuracy. In this paper, we employ economic and financial indicators as input variables. To gauge the predictive power of our model, we compare its performance with other models, particularly econometric models. Unlike AI-based models, econometric models often make more assumptions regarding various conditions such as stationarity and linearity. Consequently, econometric models offer greater control over the final solution by considering different parameters and assumptions.

At the end, there are some questions which can be answered by this study:

- What is the level of accuracy in predicting the price of Ethereum?
- Do AI-based methods demonstrate high accuracy in predicting prices within the cryptocurrency market?
- To what extent do genetic algorithms contribute to identifying the most significant and appropriate variables?
- Which models exhibit greater predictive power?

2. Introduction

Cryptocurrency is a digital asset that utilizes cryptography technology and operates without centralized control. The origins of cryptocurrencies can be traced back to the 1980s, but it was in 2009 that Bitcoin emerged as the first decentralized cryptocurrency (Chang et al., 2022). Bitcoin is often hailed as a groundbreaking revolution of the modern era due to its transformative impact on traditional trading systems and the elimination of intermediaries. This has resulted in reduced transaction costs and decreased dependence on traditional banking systems.

The trading of cryptocurrencies involves a unique process that encompasses several steps and actions when buying and selling. One of the distinguishing features of cryptocurrencies like Bitcoin (BTC) and Ethereum (ETH) is their utilization of blockchain technology, which offers secure recording and transfer of information.

Blockchain, characterized by its distributed public ledger, serves as the underlying framework for cryptocurrencies. It facilitates all activities and transactions within the cryptocurrency ecosystem, enabling currency holders to update records and engage in transactions (Bai & Vahedian, 2023). Mining is a process employed to create new units of cryptocurrency. This involves solving complex mathematical problems using powerful computers (Xu and Zhang, 2023). Alternatively, individuals can purchase cryptocurrencies from brokers and store them in digital wallets for future use. This market poses several risks, including but not limited to cyber theft and hacks, unregulated trading platforms/exchanges, regional regulations, and currency conversion risks.

A study conducted in the United States revealed that 63% of respondents working in the banking industry held a negative view of cryptocurrencies due to the associated risks discussed earlier. Ironically, banks themselves often contribute to high inflation by excessively issuing currency notes (Dapp, 2021). However, they can also play a role in strengthening the cryptocurrency market, as explained below.

- Custody services: Banks can securely hold cryptocurrencies or digital wallets on behalf of individuals, providing easy access whenever needed.
- Easy onboarding and expert assistance: Banks can act as trusted third parties, offering support to investors or clients who are new to or have limited knowledge about cryptocurrencies. They can provide services such as wallet recharging and guidance on choosing the right cryptocurrency passwords.
- Regulatory oversight: Banks can establish regulations to swiftly identify any suspicious or illicit activities, expedite payment processes through the use of public blockchains, and potentially incorporate stable coins.

By embracing these roles, banks have the potential to bridge the gap between traditional financial systems and the world of cryptocurrencies, promoting wider adoption and enhancing the overall security of digital assets. In summary, cryptocurrencies are not only a threat to banks but also present opportunities for them by facilitating cross-border financial communications. However, investing in cryptocurrencies entails both advantages and disadvantages. The structure of the paper is as follows: Section 2 presents a review of relevant literature and the latest research papers in the field. Section 3 outlines the methodology adopted for making predictions. Computational results are presented in Section 4. Finally, the last section offers concluding remarks and suggests future research directions.

3. Literature review

Metaheuristic algorithms are a methodology used to facilitate the search process in the solution space of optimization problems. These algorithms aim to explore a wide range of possibilities to find optimal or near-optimal solutions. They can be classified into different groups based on their characteristics, such as bio-stimulated algorithms, nature-inspired algorithms, physics-based algorithms, evolutionary algorithms, and swarm-based algorithms. Genetic algorithm (GA) is a widely used metaheuristic algorithm (Azadeh & Farrokhi-Asl, 2019). Within the field of forecasting, GA has been employed by numerous researchers, and its effectiveness has been demonstrated through numerical results (Zhang et al., 2022; Lai et al., 2022). Feature selection plays a crucial role in enhancing the robustness of the forecasting model by reducing the number of input variables (Abellana and Lao, 2023). In this study, we utilize GA as an efficient evolutionary algorithm for feature selection, aiming to identify the most important indicators in the forecasting process.

Furthermore, cryptocurrency is a network-based digital asset that is distributed across numerous computers. Many digital currencies operate on decentralized networks built on Blockchain technology. Blockchain serves as a shared and unalterable ledger that facilitates the recording of transactions and tracking of assets in a business network. Ethereum, represented by the symbol ETH, is a decentralized and open-source blockchain platform that ranks second in terms of market capitalization, following Bitcoin. Similar to Bitcoin, individuals seek to predict the price of ETH using various models, including machine learning techniques (Kim et al., 2021), deep learning (Politis et al., 2021), and statistical models (Rathan et al., 2019), with the aim of making profitable investments. In this paper, we also employ different methods to forecast the price of ETH and compare the results obtained through these approaches.

According to Antonakakis et al., (2019) and Fang et al. (2022), research papers in the field of cryptocurrency encompass various content structures, including (1) Introducing Cryptocurrency trading, (2) Surveys, and (3) Empirical analysis. Researchers have explored diverse topics such as the total market capitalization of the cryptocurrency market, cryptocurrency trading strategies, and trading tools utilizing econometric and mathematical models, as well as machine learning technology (Yuan and Wang, 2018). They have also investigated opportunities and challenges in the field (Al-Amri et al., 2018). Atsalakis et al. (2019) proposed a hybrid technique called PATSOS, based on a neuro-fuzzy controller, for the prediction of daily bitcoin price changes. The results demonstrated the robustness and superiority of the PATSOS technique over simpler neuro-fuzzy and artificial neural network models. They also achieved higher returns using the buy and hold strategy based on PATSOS. Maleki, et al. (2020) applied machine learning algorithms such as Random Forest Regressor and Gradient Boosting Regressor, along with time series models like AutoRegressive (AR), Moving Average (MA), and AutoRegressive Integrated Moving Average (ARIMA), to forecast bitcoin prices using three other well-known coins: Ethereum, Litecoin, and Zcash. The results revealed that among the three cryptocurrencies considered, Zcash had the best performance in forecasting Bitcoin's price. More recently, Hamayel and Owda (2021) proposed three types of Recurrent Neural Network (RNN) models: Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), and Bidirectional LSTM (bi-LSTM) models for predicting the prices of Bitcoin, Litecoin, and Ethereum. Based on Mean Absolute Percentage Error (MAPE), GRU demonstrated the most accurate and acceptable predictions. Finally, Koki et al. (2022) employed Hidden Markov Models to forecast the returns of Bitcoin, Ether, and Ripple. They also examined the impact of various specific predictors, including financial and economic factors, on the cryptocurrency return series. Their findings indicated that the Non-Homogeneous Hidden Markov (NHHM) model with four states outperformed all other considered models for all three series.

4. Methodology

Several economic indicators like the 10-Year Breakeven Inflation Rate and the 12-Month London Interbank Offered Rate, are utilized as input variables. Additionally, stock market indicators such as the S&P 500 and FTSE 100 are incorporated as input variables. These input variables are summarized in Table 1. To predict the price of ETH, we employ a combination of AI and econometric methods. We utilize econometric models like ARIMA as a prediction model. Subsequently, we compare the results. In this analysis, the price of ETH serves as the dependent variable, while the other variables are treated as independent variables. Figure 1 illustrates the research methodology. We gather daily data spanning from the beginning of 2019 to the end of 2021, encompassing three years of historical data. The Federal Reserve Economic Data and Yahoo Finance databases are the primary sources from which we access and adopt the necessary data. We take into account several factors when choosing these indicators:

- Indicators that share similar mechanisms and behaviors with ETH, such as Bitcoin price and Litecoin price.
- Key economic indicators that have an impact on ETH price, such as crude oil price and the 10-Year Breakeven Inflation Rate.
- Major global stock market indices like the S&P 500, Nasdaq, and FTSE 100.
- Financial indicators like the Federal Funds Effective Rate and the Secured Overnight Financing Rate.

Table 1
Main indicators as input variables

Row	Indicator	Variable scale	Symbols	Nature (Input Variable Vs. Target Variable)
1	Coinbase Bitcoin	U.S. Dollars	CBBTCUSD	IV
2	Coinbase Bitcoin	U.S. Dollars	CBBTCUSD	IV
3	Crude Oil Prices: Brent - Europe	Dollars per Barrel	DCOILBRENTEU	IV
4	U.S. Dollars to Euro Spot Exchange Rate	U.S. Dollars to One Euro	DEXUSEU	IV
5	Federal Funds Effective Rate	Percent	DFF	IV
6	Market Yield on U.S. Treasury Securities at 2-Year Constant Maturity	Percent	DGS2	IV
7	Nominal Emerging Market Economies	U.S. Dollar Index, Index Jan 2006=100	DTWEXEMEGS	IV
8	Secured Overnight Financing Rate	Percent	SOFR	IV
9	10-Year Breakeven Inflation Rate	Percent	T10YIE	IV
10	12-Month London Interbank Offered Rate (LIBOR)	Based on U.S. Dollar, Percent	USD12MD156N	IV
11	DAX Performance-Index	-	^GDAXI	IV
12	KOSPI Composite Index	-	^KS11	IV
13	Nikkei 225	-	^N225	IV
14	^NIFTY50	-	^NSE	IV
15	Dow Jones Industrial Average	-	DJIA	IV
16	Financial Times Stock Exchange 100 Index	-	FTSE100	IV
17	Standard and Poor's 500	-	S&P500	IV
18	Nasdaq Composite	-	Nasdaq	IV
19	SSE Composite Index	-	000001.SS	IV
20	Singapore Stock Market	-	STI	IV
21	Ethereum price	-	ETH-USD	TV

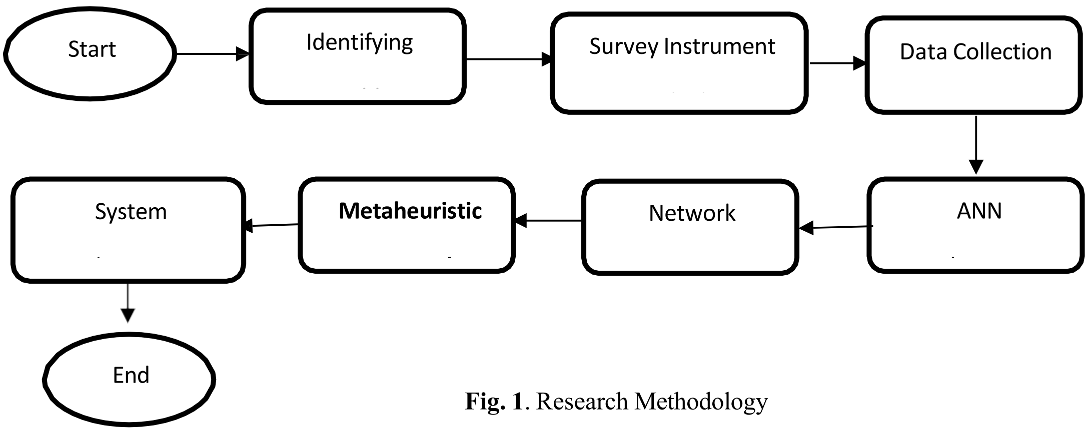

Fig. 1. Research Methodology

5. Artificial Neural Network

Artificial neural network (ANN) simulates human thinking and can be enhanced through training (Naseri et al., 2022; Kasaei and Rajendran, 2023). One of its key advantages is its compatibility with complex data structures, making it a suitable approach for analyzing ETH price, which exhibits high volatility. ANN consists of three main layers. The first layer is the input layer, where data is initially fed into the network. The volume of input data plays a crucial role in network performance. A smaller amount of input data results in faster and more accurate network computations. To achieve this, we employ Genetic Algorithm (GA) for feature selection, which helps identify the optimal and most relevant input variables. Data then flows from the input layer to the hidden layer. However, between these two layers, an activation function is utilized to determine linearity or non-linearity. In this study, the chosen activation function is the non-linear Tan-Sigmoid, which yields outputs ranging from -1 to 1. This function is employed to standardize numbers within the specified range. In this paper, a linear activation function is used to extract linear features. Finally, the obtained information is transferred to the output layer, where the target variable is located. In conclusion, a multi-layer perceptron (MLP) with three layers is employed as the network structure. The number of hidden layers is determined through trial and error. This involves investigating different network structures and selecting the one that exhibits the lowest error and highest predictability. The error-backpropagation method is utilized for training the ANN. Additionally, the Levenberg-Marquardt (LM) algorithm is employed as an optimization algorithm to enhance the learning process of the model. The parameters of the ANN are presented in Table 2.

Table 2
Network properties

Parameters	Explanation
Training	Back-propagation (BP)
Optimization Algorithm	Levenberg-Marquardt (LM)
Training rate	0.01
Iterations	1000
Activation function	Tan-Sigmoid Pure line

It is crucial to normalize and scale the data prior to training the network. This can be achieved using Equation (1). In this equation, numerator i shows the amount of data. The structure of the network has been represented in Figure 2 (Shahvaroughi Farahani and Razavi, 2021).

$$\tilde{S_i} = \frac{(S_i - S_{min})}{S_{max} - S_{min}} . i = 1 \ldots . N \tag{1}$$

where:

$\tilde{S_i}$: Normalized data

S_i: Each observation of each variable

S_{min}: Minimum value of each variable

S_{max}: Maximum value of each variable

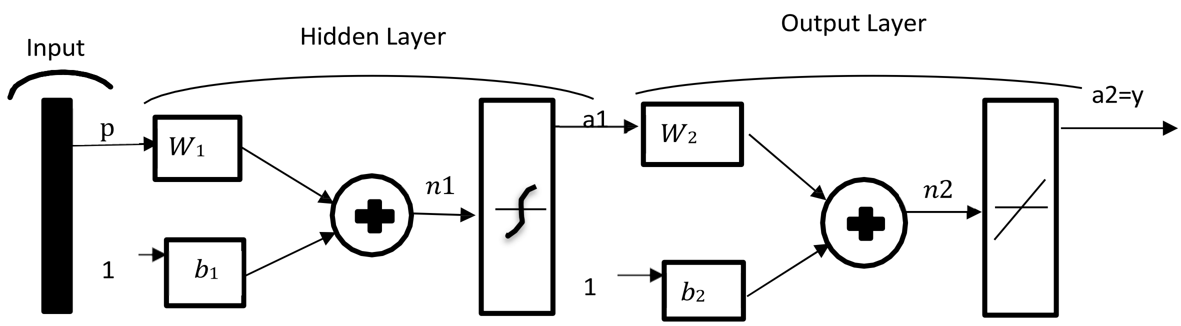

Fig 2. Architecture of the proposed Neural Network

As previously mentioned, the input variables (p) consist of weights (W_1)) and biases (b_1)). Within each layer, these values are summed (+) and then passed through a non-linear activation function. Subsequently, they are summed once again to create new weights (W_2)) and biases (b_2)), which are then passed through a linear activation function.

6. Genetic algorithm

Genetic Algorithm (GA) is a global search-based optimization technique utilized to approximate optimal solutions (Katoch et al, 2021). GA begins with an initial set of solutions and iteratively improves this set, referred to as the solution population. The GA flowchart is provided in Figure 3. In this paper, each solution is represented by a string of binary numbers (0s and 1s) known as a chromosome. Therefore, we employ a binary solution representation using chromosomes consisting of 24 bits. The first 19 bits represent the 19 independent variables, where 1 indicates the existence of an input variable, while 0 denotes its absence. The remaining 5 bits (25=32) are used to determine the number of neurons in the hidden layers. GA also employs two main operators, mutation and crossover, to explore the solution space. In GA, the Roulette wheel method is employed to select individuals for the next generation. This selection process is determined by the fitness of each individual, meaning that individuals with higher fitness have a greater chance of being chosen for producing the next generation. Typically, the crossover rate ranges from 0.8% to 0.95%. Additionally, new or diverse solutions can be obtained through mutation, which involves flipping certain digits in a string. The crossover and mutation processes are illustrated in Figure 4. GA parameters and their value are also shown in Table 5. For getting more information about the GA pseudo-code (i.e., steps and how to get the parameters), interested readers can refer to Shahvaroughi Farahani and Razavi (2021).

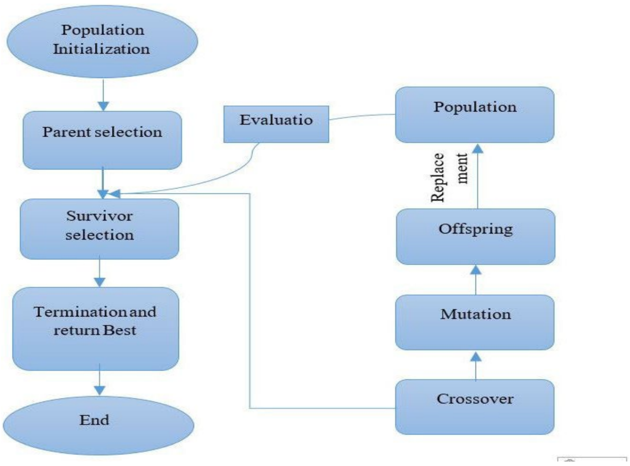

Fig. 3. Genetic algorithm process

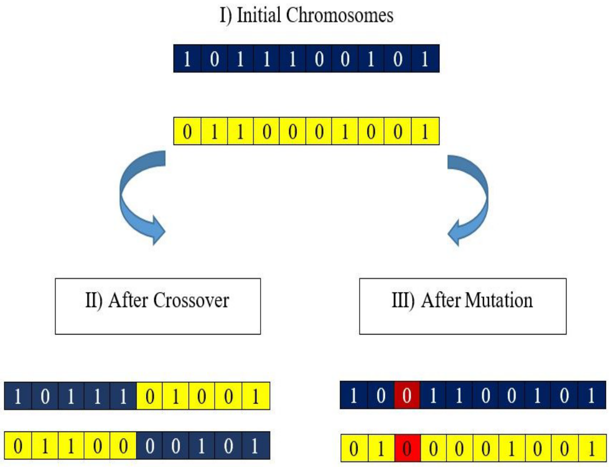

Fig. 4. Cross-over and mutation operator

Table 3
GA parameters

Output Error	Output Activation Function	Input Activation Function	Mutation Rate	Crossover Rate	Number of Generation	Population size	Max Iterations
MSE	Logistic	Logistic	0.1	0.9	50	20	1000
Selection parents			**Mutation**		**Crossover**		
Roulette wheel method			Binary Method		One-point method		

7. Econometric model

Econometric models are statistical models commonly employed in econometrics (Shobana and Umamaheswari, 2021). These models are useful when dealing with multiple independent variables and the aim is to examine their individual impacts on the dependent variable. Selecting an appropriate econometric model is a critical task that requires understanding the model's properties and qualifications before application. In this study, we utilize the Jarque-Bera (J-B) test, a goodness-of-fit test, which offers advantages for symmetric distributions with medium to long tails and slightly skewed distributions with long tails.

We employ the Augmented Dickey-Fuller test to test for stationarity in our research. It should be noted that for non-stationary time series, differencing can be employed to transform them into stationary series.

A simple linear regression with one independent variable and two dependent variables is presented below:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i. \qquad i = 1. \ldots . n \tag{2}$$

where:

y_i: dependent variable i

β_0: intercept

β_1: x_i coefficient

x_i: independent variable

Lastly, we utilize the ARIMA model to predict the price for the next day. ARIMA, which stands for Autoregressive Integrated Moving Average, is a statistical analysis model employed for analyzing time series data and making future trend predictions. The ARIMA model comprises three fundamental components: [p, d, q]. Each component serves a specific purpose. An AR(p) model is an autoregressive model that utilizes lagged values of Y_t as predictor variables. The lag parameter (p) indicates how many previous periods' shocks affect future periods. The (d) component in ARIMA refers to differencing, which is employed to achieve stationarity in the time series data. By taking differences between consecutive observations, non-stationary data can be transformed into a stationary form. The (q) component represents the moving average component of the ARIMA model. It considers the influence of the residual errors or shocks from previous predictions when forecasting future values.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \varepsilon_1 \tag{3}$$

where, $(Y_{t-1}. Y_{t-2}.... . Y_{t-p})$ are the previous series values (lags), $(\beta_1. \beta_2.... ., \beta_p)$ are the coefficient of lag that the model approximate and α is the intercept, also estimated by the model.

Moving Average (MA) model is a model which Y_t depends only on the lagged forecast errors

$$Y_t = \alpha + \varepsilon_t + \emptyset_1 \varepsilon_{t-1} + \emptyset_2 \varepsilon_{t-2} + \cdots + \emptyset_q \varepsilon_{t-q} + \varepsilon_{t-q} \tag{4}$$

where the error terms are the errors of the autoregressive models of the respective lags. The errors ε_t and ε_{t-1} are the errors from the following equations :

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_0 Y_0 + \varepsilon_1 \tag{5}$$

$$Y_{t-1} = \beta_1 Y_{t-2} + \beta_2 Y_{t-3} + \cdots + \beta_0 Y_{t-n} + \varepsilon_{t-1} \tag{6}$$

ARIMA model can form by integration of both AR(p) and MA(q) with a differencing. So the equation becomes:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \varepsilon_1 + \emptyset_1 \varepsilon_{t-1} + \emptyset_2 \varepsilon_{t-2} + \cdots + \emptyset_q \varepsilon_{t-q} \tag{7}$$

Now, Fig. 5 shows the regression analysis and ARIMA process:

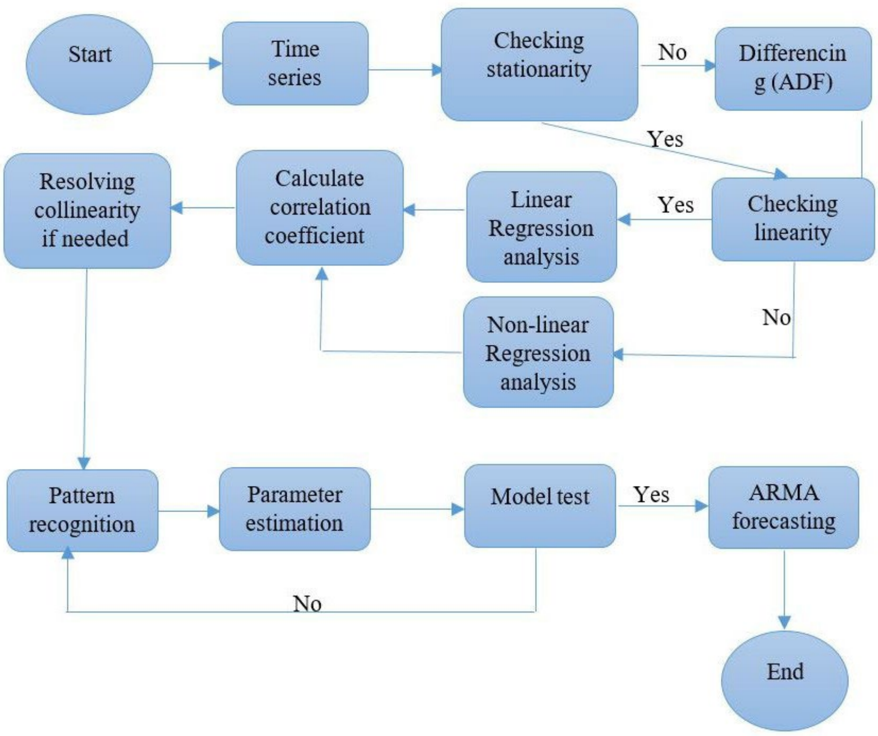

Fig. 5. Regression analysis Flowchart

According to the above Figure, testing for stationarity is crucial because the presence of trends and seasonality can cause variations in the values of a time series at different points in time. To address this issue, differencing can be employed. Furthermore, it is important to assess linearity based on the assumptions of econometric models, as it can impact the choice of appropriate analysis methods. Depending on the linearity, you can opt for either linear or non-linear regression analysis. Several criteria, such as R-squared and Durbin Watson, can be used to evaluate the goodness of fit. In some cases, there may be collinearity among predictor variables, which requires correction as these variables cannot independently predict the value of the dependent variable. By calculating the correlation coefficient and checking for collinearity, you can ensure a favorable condition for regression. For the application of the ARIMA model as a predictive model, a correlogram should be examined. These figures allow for the diagnosis of the ARMA model and the identification of patterns to determine the orders of AR (p) and MA (q) models.

5. Findings and results

In our study, we work with two types of variables: (1) Inputs, which represent the dependent variables, and (2) Target, which represents the independent variable (specifically, the price of ETH).

To demonstrate the distribution of instances within the dataset, we present a pie chart (Figure 6). The chart showcases the usage rate of instances for training data, selection data, and testing data. Out of the total 708 instances, 496 instances (70.1%) are allocated for training, 106 instances (15%) are used for selection, another 106 instances (15%) serve as testing data, and there are no instances left unused (0%).

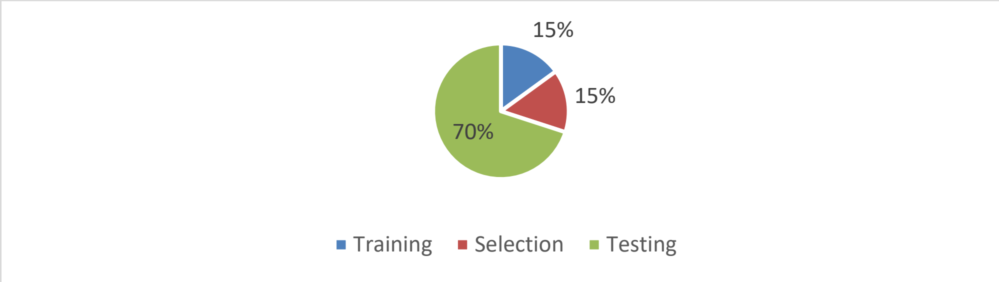

Fig. 6. Instances pie chart

The correlation coefficient between the input and target variables is a valuable tool for assessing the significance of each variable in predicting the price of ETH. We employ the Pearson correlation coefficient formula, which is expressed as Formula (8):

$$r = \frac{\sum (x_i - \overline{x})(y - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 (y - \overline{y})^2}} \tag{8}$$

where

r: Correlation coefficient

x_i: Values of the x-variable in a sample

$\bar{x}$: Mean of the values of the x-variable

y_i: Values of the y-variable in a sample

$\bar{y}$: Mean of the values of the x-variable

Table 4 shows the value of the correlations between all inputs and target variables. The maximum correlation (0.886806) is yield between the input variable CBBTCUSD and the target variable ETH price. The variables in this table are sorted in descending correlation order.

Table 4

Correlation coefficient between input variables and target

Variables	Type	ETH price
CBBTCUSD	Linear	0.886806
DJIA	Linear	0.811489
NIFTY50	Linear	0.807325
S&P500	Linear	0.795083
GDAXI	Linear	0.755396
NASDAQ	Linear	0.744184
T10YIE	Linear	0.664956
KS11	Linear	0.653814
N225	Linear	0.635221
SHANGHAI	Linear	0.594489
DFF	Linear	-0.55862
USD12MD156N	Linear	-0.51393
DEXUSEU	Linear	0.481729
DCOILBRENTEU	Linear	0.468803
DGS2	Linear	-0.42246
SOFR	Linear	-0.41098
CBLTCUSD	Linear	0.293068
SINGAPORE	Linear	0.243838
FTSE100	Linear	0.21915

9. Artificial Neural Network results

To determine the optimal solution for predicting the next day's ETH price, we employ Artificial Neural Networks (ANN). The first step in this process is to identify the most suitable network architecture. The parameters taken into consideration for this purpose are as Table 5. As mentioned previously, we utilize a trial-and-error approach to identify the most effective architecture. Fig. 7 illustrates the networks that have been tested in this process. The best network includes 27 neurons in hidden layer along with the highest R-Squared means 0.992363. Five dataset error can see in Fig. 8.

Table 5
Network parameters

Parameter	Value
Logistic	Input activation FX
Output name	ETH price
Output error FX	Sum of Squares
Output activation function	Logistics

ID	Architecture	# of Weights	Fitness	Train Error	Validation Error	Test Error	AIC	Correlation	R-Squared	Stop Reason
1	[19-3-1]	64	0.085758	11.713454	11.872656	11.66066	-1668.411552	0.990417	0.980679	All iterations done
2	[19-48-1]	1009	0.114776	7.620323	8.363061	8.712622	13.937429	0.995617	0.991216	All iterations done
3	[19-30-1]	631	0.119875	7.393431	9.237349	8.342046	-756.662155	0.995897	0.991772	All iterations done
4	[19-19-1]	400	0.108983	7.452969	8.538849	9.175784	-1214.788201	0.995888	0.991754	All iterations done
5	[19-41-1]	862	0.111703	7.771595	9.123082	8.95234	-270.568378	0.995569	0.991141	All iterations done
6	[19-36-1]	757	0.116472	7.095478	8.795599	8.585729	-524.530005	0.996226	0.992397	All iterations done
7	[19-25-1]	526	0.125653	7.07761	8.47767	7.958408	-987.747843	0.996429	0.992846	All iterations done
8	[19-22-1]	463	0.120647	7.524922	9.193093	8.288648	-1084.147584	0.995805	0.991594	All iterations done
9	[19-28-1]	589	0.105677	8.239238	9.352911	9.46277	-788.345572	0.995082	0.99016	All iterations done
10	[19-26-1]	547	0.130337	7.233181	7.932706	7.672423	-935.246134	0.996009	0.992	All iterations done
11	**[19-27-1]**	**568**	**0.133881**	**7.16557**	**8.206067**	**7.469302**	**-897.782156**	**0.996184**	**0.992363**	**All iterations done**

Fig. 7. Top 5 best network architecture

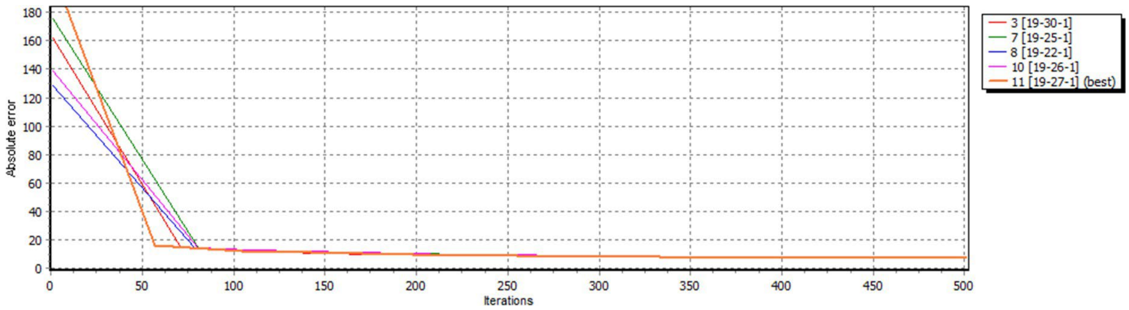

Fig. 8. Top 5 best Dataset errors

After finding appropriate structure, we need to train the network. Table 8 presents the training parameters. The next step is to test the network with 30 percentage of data. Testing error graph is depicted and presented in Fig. 9. In this figure, the red-line shows the actual data (Target) and the blue-line shows the predicted. As you can see, both lines are very coincident.

Table 6
Training parameters

Parameters	Training	Validation
Absolute error	0.114011	14.739959
Network error	6.96E-08	0
Error improvement	1.17E-12	
Iteration	329	
Training speed, ite/sec	0.39752	
Architecture	[19-27-1]	
Training algorithm	Levenberg-Marquardt	
Training stop reason	No error improvement	

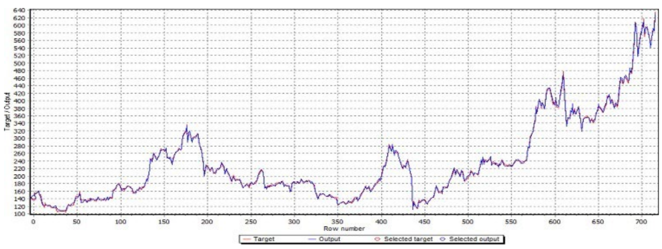

Fig. 9. Testing error (Actual vs. output)

For more information and details about training graph please refer to the appendix (Fig. I). Fig. 10 shows the performance of network (i.e., training, validation and testing) in each iteration and the best validation. The best validation performance is 0.00025943 at epoch 22. You can also see the regression analysis for each dataset i.e., training, validation and testing separately.

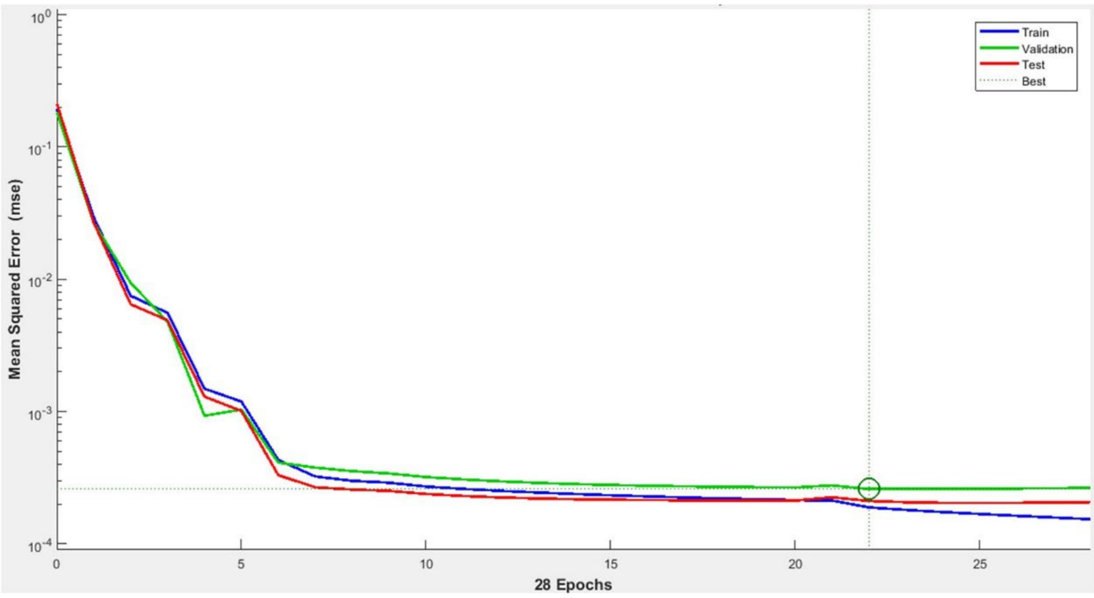

Fig. 10. Network Performance

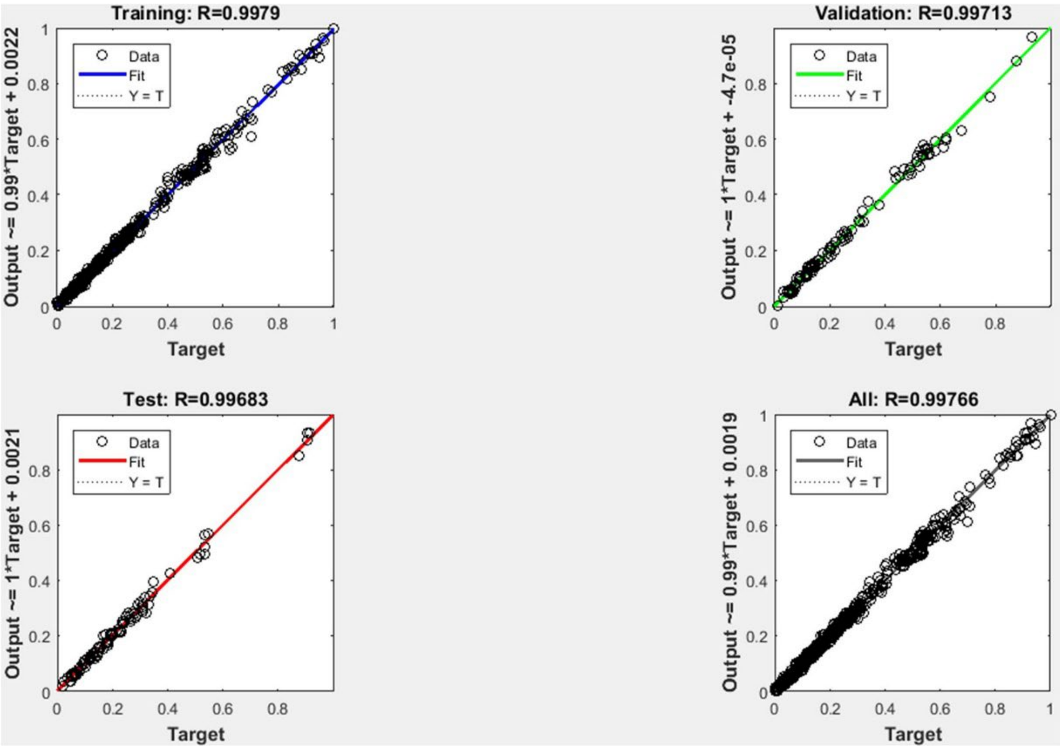

Fig. 11. ANN Regression

For more information about the training state, more parameters such as gradient, mu and etc. please see appendix (Figures II, III). Finally, Table 9 shows different error functions and statistical measurement such as mean, std. dev etc.

Table 7
Error estimations and functions

Statistical Measurement	Target	output	AE	ARE
Mean	235.5870	235.5534	2.8968	0.0138
Std.Dev	110.4408	110.6292	1.5571	0.0088
Min	104.5353	108.1162	0.0108	0.00008
Max	636.1818	629.6741	9.3178	0.0578

10. Genetic algorithm results

In ANN, model selection is employed to discover a neural network topology that optimizes the error on new data. There are two types of model selection: (1) Order selection and (2) Input selection. Order selection aims to determine the appropriate and optimal number of hidden layers, while input selection identifies the most important and optimal input variables. To perform order selection, an incremental order method is typically applied. This method begins with the minimum order and gradually adds a specified amount of perceptron in each iteration. By iteratively adjusting the network architecture, we can find the configuration that achieves the best performance. The parameters of the incremental order selection method, which guide the process of incrementally adding perceptron, are presented in Table 8. The error history during the incremental order selection is depicted in Fig. 12. The training error is represented by the blue line, while the selection error is symbolized by the orange line. In Table 9, the order selection results obtained from the incremental order algorithm are presented. These results include the final states of the neural network, the error functional, and the order selection algorithm employed.

Table 8
Order selection algorithm parameters

Parameters	Description	Value
Minimum order	Number of minimum hidden perceptron's to be evaluated.	1
Maximum order	Number of maximum hidden perceptron's to be evaluated.	10
Step	Number of hidden perceptron's added in each iteration.	1
Trials number	Number of trials for each neural network.	3
Tolerance	Tolerance for the selection error in the trainings of the algorithm.	0.01
Selection loss goal	Goal value for the selection error.	0
Maximum selection failures	Maximum number of iterations at which the selection error increases.	5
Maximum iterations number	Maximum number of iterations to perform the algorithm.	1000
Maximum time	Maximum time for the order selection algorithm.	3600
Plot training error history	Plot a graph with the training error of each iteration.	TRUE
Plot selection error history	Plot a graph with the selection error of each iteration.	TRUE

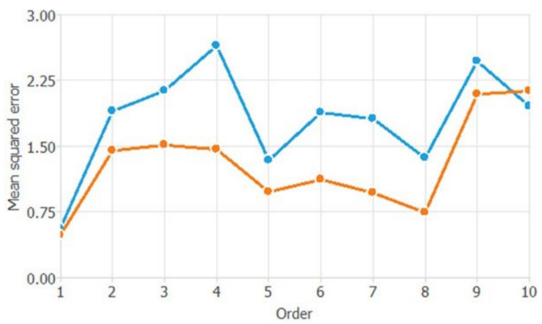

Fig. 12. Incremental Order error plot

Table 9
Incremental order results

Parameters	Value
Optimal order	1
Optimum training error	0.55698
Optimum selection error	0.48018
Iterations number	10
Elapsed time	0:02

At this time, the GA is employed to select the most optimal and fitting input variables in ANN. The parameters associated with GA are outlined in Table 10. Throughout the GA input selection process, the error history is illustrated in Fig. 13. The blue line corresponds to the training error, starting with an initial value of 0.257581 and concluding at 0.461958 after 100 iterations. Conversely, the orange line represents the selection error, with an initial value of 0.161565 and a final value of 0.170137 after 100 iterations. Table 11 provides the training, validation, and testing errors, as well as the network architecture obtained through the utilization of the GA. In summary, GA enables a reduction in the number of input variables, leading to an increase in the R-Squared value.

Table 10
GA parameters as input selection

Parameters	Description	Value
Trials number	Number of trials for each neural network.	1
Tolerance	Tolerance for the selection error in the trainings of the algorithm.	0.01
Population size	Size of the population of each generation.	20
Initialization method	Initialization method used in the algorithm.	Random
Fitness assignment method	Fitness assignment method used in the algorithm.	Rank Based
Crossover method	Crossover method used in the algorithm.	One-point
Elitism size	Number of individuals which will always be selected for recombination.	2
Crossover first point	First point used in the One Point and Two Point crossover method. If it is 0 the algorithm selects a random point for each pair of offspring's.	0
Crossover second point	Second point used in the Two Point crossover method. If it is 0 the algorithm selects a random point for each pair of offspring's.	0
Selective pressure	Rank Based fitness assignment allows values for the selective pressure greater than 0.	1.5
Mutation rate	This is a parameter of the mutation operator.	0.05
Selection loss goal	Goal value for the selection error.	0
Maximum Generations number	Maximum number of generations to perform the algorithm.	100
Maximum time	Maximum time for the inputs selection algorithm.	3600
Plot training error history	Plot a graph with the optimum training error of each generation.	TRUE
Plot selection error history	Plot a graph with the optimum selection error of each generation.	TRUE
Plot generation mean history	Plot a graph with the mean of the selection error of each generation.	TRUE

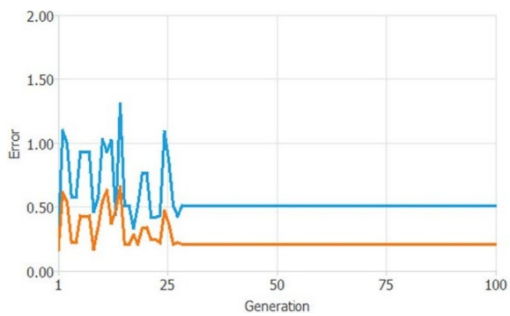

Fig. 13. GA error plot

Table 11
GA input selection result

Parameters	Value
Optimal number of inputs	12
Optimum training error	0.461958
Optimum selection error	0.170137
Generations number	100
Elapsed time	0:01

Fig. 14 illustrates the resulting deep architecture, which consists of a scaling layer, a neural network, and an un-scaling layer. In the Fig. 14, yellow circles represent scaling neurons, blue circles denote perceptron neurons, and red circles indicate un-scaling neurons. The architecture has 12 input variables and 1 output variable. The complexity of the architecture is indicated by the number of hidden neurons, which is 1. Additionally, Table 12 presents the error estimations for the training, selection, and testing phases. To obtain further details and additional information regarding the history of the mean of the selection error in each iteration during the GA input selection process, please refer to Figure IV in the Appendix section. The initial value of the GA error mean is 1.58226, and it converges to a final value of 0.497 after 100 generations.

Fig. 14. Final architecture

Table 12
Error table

Criteria	Training	Selection	Testing
Sum squared error	55.0845	8.14139	10.9464
Mean squared error	0.111058	0.076806	0.103268
Root mean squared error	0.333253	0.277138	0.321354
Normalized squared error	2.35655	3.03587	2.39714
Minkowski error	78.9245	13.4898	16.0615

11. Econometric models results

One of the primary considerations in data analysis is assessing the normality assumption. Considering the histogram distribution helps in selecting the right econometric model for analysis, thereby enhancing the accuracy of predictions or estimations. In the investigated time horizon, Fig. 15 displays the distribution of ETH prices.

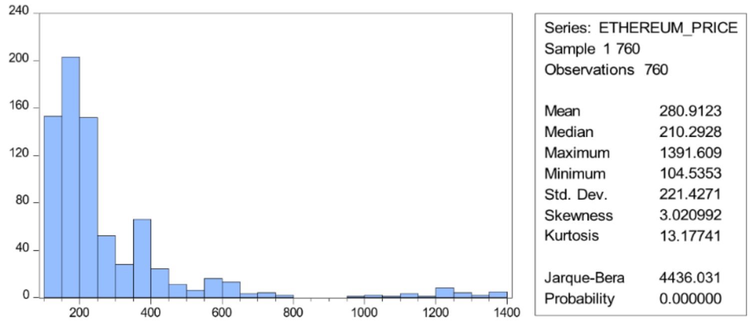

Fig 15. Ethereum price histogram

The Ethereum price distribution is clearly non-normal and exhibits skewness, indicating the need for caution when selecting an appropriate model. It is advisable to employ suitable models like Least Squares with Gauss-Newton/Marquardt steps. In the subsequent step, the stationarity of the process is assessed. Financial time series data typically do not exhibit stationarity. As depicted in Fig. 15, they possess attributes such as skewness and kurtosis with fat tails. Consequently, it is essential to examine and identify the stationarity of the time series. Therefore, as mentioned earlier, the Augmented Dickey-Fuller (ADF) test is employed as a unit root test to determine if the data is stationary.

Table 13
ADF Unit root test results
Null Hypothesis: ETHEREUM_PRICE has a unit root Exogenous: Constant
Lag Length: 10 (Automatic - based on SIC, maxlag=19)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		2.013394	0.9999
Test critical values:	1% level	-3.438854	
	5% level	-2.865183	
	10% level	-2.568766	

*MacKinnon (1996) one-sided p-values.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
ETHEREUM_PRICE (-1)	0.008667	0.004305	2.013394	0.0444
D (ETHEREUM_PRICE (-1))	-0.042978	0.035342	-1.216061	0.2244
D (ETHEREUM_PRICE (-2))	0.044748	0.035483	1.261110	0.2077
D (ETHEREUM_PRICE (-3))	-0.081373	0.034782	-2.339508	0.0196
D (ETHEREUM_PRICE (-4))	0.071553	0.034514	2.073149	0.0385
D (ETHEREUM_PRICE (-5))	-0.074857	0.034968	-2.140715	0.0326
D (ETHEREUM_PRICE (-6))	0.133395	0.035824	3.723615	0.0002
D (ETHEREUM_PRICE (-7))	-0.171371	0.036347	-4.714853	0.0000
D (ETHEREUM_PRICE (-8))	-0.196722	0.037460	-5.251583	0.0000
D (ETHEREUM_PRICE (-9))	-0.044568	0.044133	-1.009859	0.3129
D (ETHEREUM_PRICE (-10))	0.463700	0.043892	10.56456	0.0000
C	-0.997012	1.285778	-0.775415	0.4383

R-squared	0.257615	Mean dependent var	1.675533
Adjusted R-squared	0.246535	S.D. dependent var	22.33864
S.E. of regression	19.39047	Akaike info criterion	8.783333
Sum squared resid	277105.0	Schwarz criterion	8.857331
Log likelihood	-3277.358	Hannan-Quinn criter	8.811848
F-statistic	23.24966	Durbin-Watson stat	1.981681
Prob(F-statistic)	0.000000		

Based on the information provided in Tables 13, it can be observed that the t-statistic (i.e., 2.013394) exceeds the critical values at various significance levels (1%, 5%, and 10%). This indicates the presence of at least one-unit root, implying that the data is non-stationary. Consequently, to achieve stationarity, we apply first-level differencing. The resulting data after differencing is presented in Table 14.

Table 14

First-level differencing

Null Hypothesis: D(ETHEREUM_PRICE) has a unit root Exogenous: Constant

Lag Length: 9 (Automatic - based on SIC, maxlag=19)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-6.247577	0.0000
Test critical values	1% level	-3.438854	
	5% level	-2.865183	
	10% level	-2.568766	

*MacKinnon (1996) one-sided p-values.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
D (ETHEREUM_PRICE (-1))	-0.702183	0.112393	-6.247577	0.0000
D (ETHEREUM_PRICE (-1),2)	-0.319966	0.105506	-3.032681	0.0025
D (ETHEREUM_PRICE (-2),2)	-0.252906	0.097320	-2.598689	0.0095
D (ETHEREUM_PRICE (-3),2)	-0.314466	0.090563	-3.472340	0.0005
D (ETHEREUM_PRICE (-4),2)	-0.224458	0.084917	-2.643265	0.0084
D (ETHEREUM_PRICE (-5),2)	-0.278908	0.077484	-3.599554	0.0003
D (ETHEREUM_PRICE (-6),2)	-0.130771	0.070846	-1.845855	0.0653
D (ETHEREUM_PRICE (-7),2)	-0.287382	0.063306	-4.539545	0.0000
D (ETHEREUM_PRICE (-8),2)	-0.470131	0.053389	-8.805735	0.0000
D (ETHEREUM_PRICE (-9),2)	-0.490324	0.041939	-11.69124	0.0000
C	1.134833	0.730961	1.552522	0.1210

R-squared	0.639090	Mean dependent var	0.068234
Adjusted R-squared	0.634200	S.D. dependent var	32.12648
S.E. of regression	19.43055	Akaike info criterion	8.786148
Sum squared resid	278629.2	Schwarz criterion	8.853980
Log likelihood	-3279.412	Hannan-Quinn criter.	8.812286
F-statistic	130.6832	Durbin-Watson stat	1.993763
Prob(F-statistic)	0.000000		

Table 15

Model selection criteria Table

Model Selection Criteria Table Dependent Variable: D (ETH, 2) Sample: 1 760 Included

Model	LogL	AIC*	BIC	HQ
(3,4)	-3382.366070	8.948195	9.003177	8.969371
(4,4)	-3381.657401	8.948964	9.010055	8.972492
(4,3)	-3396.485978	8.985451	9.040433	9.006626
(1,4)	-3403.538291	8.998782	9.041545	9.015251
(2,3)	-3404.098971	9.000261	9.043025	9.016731
(2,4)	-3403.528849	9.001395	9.050268	9.020218
(3,2)	-3404.781705	9.002063	9.044826	9.018532
(3,3)	-3404.455589	9.003841	9.052713	9.022663
(4,2)	-3404.715235	9.004526	9.053398	9.023348
(1,3)	-3413.538817	9.022530	9.059184	9.036647
(1,2)	-3415.150520	9.024144	9.054689	9.035908
(4,1)	-3413.712596	9.025627	9.068390	9.042096
(3,1)	-3416.285516	9.029777	9.066432	9.043894
(0,4)	-3417.972995	9.034230	9.070884	9.048346
(2,2)	-3421.650850	9.043934	9.080588	9.058050
(0,2)	-3423.858877	9.044483	9.068919	9.053894
(1,1)	-3423.912214	9.044623	9.069060	9.054034
(0,1)	-3425.061348	9.045017	9.063344	9.052075
(0,3)	-3423.596016	9.046427	9.076973	9.058191
(2,1)	-3423.818635	9.047015	9.077560	9.058779
(4,0)	-3489.221507	9.222220	9.258875	9.236337
(3,0)	-3496.745811	9.239435	9.269980	9.251199
(2,0)	-3553.783216	9.387291	9.411727	9.396702
(1,0)	-3580.043286	9.453940	9.472267	9.460998
(0,0)	-3701.327482	9.771313	9.783531	9.776018

After performing the differencing on the data, we can employ the ARIMA model for prediction. In this analysis, the ARIMA model estimation is conducted using the automatic ARMA forecasting feature in Eviews10. Table 15 presents various model selections based on significant criteria and multiple evaluation metrics. Additionally, Table 16 summarizes the ARIMA equation. The results of the ARIMA forecasting model are summarized in Table 17. Based on the information provided in these tables, it can be concluded that the optimal model order for the ARMA model is AR (3) and MA (4) with an AIC value of 8.948195. Since we have applied one level of differencing, the final ARIMA model is denoted as (3,1,4). According to Table 15, we have used 25 lags to estimate ARMA model.

Table 16
Equation Output

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.015577	0.017391	0.895707	0.3707
AR (1)	-0.241086	0.038292	-6.295954	0.0000
AR (2)	-0.317776	0.043727	-7.267317	0.0000
AR (3)	-0.940496	0.040967	-22.95759	0.0000
MA (1)	-0.823171	0.048955	-16.81473	0.0000
MA (2)	0.143809	0.071321	2.016356	0.0441
MA (3)	0.467407	0.087170	5.362009	0.0000
MA (4)	-0.751360	0.053465	-14.05333	0.0000
SIGMASQ	436.1670	7.264702	60.03921	0.0000
R-squared	0.572642	Mean dependent va		0.047233
Adjusted R-squared	0.568077	S.D. dependent var		31.96813
S.E. of regression	21.00971	Akaike info criterio		8.948195
Sum squared resid	330614.6	Schwarz criterion		9.003177
Log likelihood	-3382.366	Hannan-Quinn crite		8.969371
F-statistic	125.4536	Durbin-Watson stat		1.940383
Prob(F-statistic)	0.000000			
Inverted AR Roots	.35+.93i	.35-.93i	-.95	
Inverted MA Roots	.98	.35+.88i	.35-.88i	-.85

Dependent Variable: D(ETH,2) Method: ARMA Maximum Likelihood (BFGS) Sample: 3 760 Included observations: 758 Convergence achieved after 89 iterations Coefficient covariance computed using outer product of gradients

Table 17
ARIMA forecasting summary

Automatic ARIMA Forecasting Selected dependent variable: D (ETH, 2) Sample: 1760 Included observations: 758
Number of estimated ARMA models: 25
Number of non-converged estimations: 0
Selected ARMA model: (3,4) AIC value: 8.94819543562

According to Fig. 16, it can be observed that the ARMA model with an order of (3, 4), representing AR (3) and MA (4), exhibits the lowest error based on the AIC criterion. It is important to note that if the difference in AIC values, also known as delta-AIC, is less than 2, there is no significant difference between the competing models. In Figure 16, all the delta-AIC values are less than 2 when comparing the models with each other. It is worth mentioning that the selection of ARMA (3, 4) was obtained using the automatic ARIMA forecasting feature in Eviews.

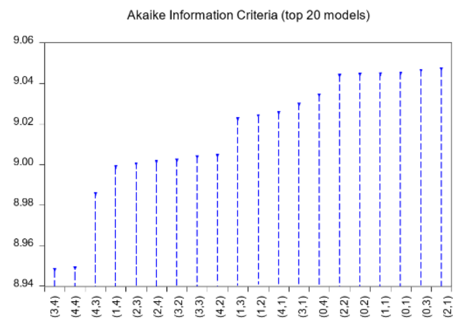

Fig. 16. Akaike information criteria (top 20 models)

The subsequent step involves conducting a regression analysis using Eviews10 as the analytical tool. The Least Squares method, specifically Gauss-Newton/Marquardt steps, is utilized with 500 iterations to estimate the regression and establish the relationship between the selected input variables obtained through GA and the target variable, ETH price mean. In Table 20, it is evident that variables with a probability value lower than 0.05 are deemed more significant in explaining the model.

Consequently, variables C (6), C (11), and C (12) are considered less important. The high R-squared value of approximately 0.95 indicates a strong fit and suggests a good regression model. Fig. 17 depicts the actual and predicted (fitted) values, along with the residuals. The red line represents the actual data, the green line represents the predicted data, and the blue line represents the residual. The close alignment between the red and green lines signifies a good forecast. For further insights into the prediction of ETH price, please refer to the appendix (Fig V).

Table 18

Ethereum regression using LSM

	Coefficient	Std. Error	t-Statistic	Prob.
C (1)	-266.8373	36.81040	-7.248964	0.0000
C (2)	0.021891	0.000649	33.72564	0.0000
C (3)	1.242690	0.102197	12.15980	0.0000
C (4)	0.978225	0.335595	2.914895	0.0037
C (5)	-86.67940	20.39444	-4.250149	0.0000
C (6)	0.001334	0.001104	1.208265	0.2273
C (7)	-0.058489	0.007593	-7.702550	0.0000
C (8)	-0.010569	0.002955	-3.576212	0.0004
C (9)	0.071957	0.011227	6.409238	0.0000
C (10)	0.031582	0.003478	9.081595	0.0000
C (11)	-0.000578	0.002230	-0.259207	0.7955
C (12)	0.021562	0.016379	1.316464	0.1884
C (13)	156.1531	21.02015	7.428733	0.0000
R-squared	0.946865	Mean dependent var		255.0865
Adjusted R-squared	0.945982	S.D. dependent var		155.0148
S.E. of regression	36.02808	Akaike info criterion		10.02400
Sum squared resid	937172.3	Schwarz criterion		10.10536
Log likelihood	-3670.821	Hannan-Quinn criter.		10.05538
F-statistic	1072.178	Durbin-Watson stat		0.139845
Prob(F-statistic)	0.000000			

Dependent Variable: ETHEREUM_PRICE Method: Least Squares (Gauss-Newton / Marquardt steps) Sample: 1740

Included observations: 735 after adjustments

ETHEREUM_PRICE=C (1) +C (2) *CBBTCUSD+C (3) *CBLTCUSD+C (4)*DCOILBRENTEU+C (5) *DGS2+C (6) *DJIA+C (7) *FTSE100+C (8) *GDAXI+C (9) *KS11+C (10) *NASDAQ+C (11) *NIFTY50+C (12) *SINGAPORE+C (13) *USD12MD156N

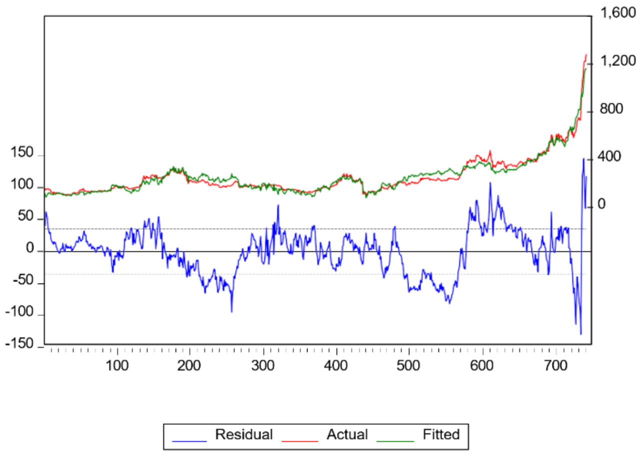

Fig 17. Actual vs. Predicted

12. Comparing the results

After conducting the experiments, it is necessary to compare the results and identify the superior model. We utilized three types of models for predicting the ETH price for the next day. One of the primary criteria for comparing these models is the R-squared value. The R-squared rate in the ANN model is 0.99766, while for the economic and econometric models, specifically regression analysis and ARMA, the rates are 0.946865 and 0.572642, respectively. Based on the R-squared values, the ANN model demonstrates the best performance. It should be noted that GA was used for feature selection and identifying the most important and relevant input variables, and it was not a predictive model itself. Furthermore, we need to compare the results based on the forecasting error. The error rates in the ANN, regression analysis, and ARMA models are 1.5571, 21.00971, and 36.02808, respectively. According to the error estimation, the ANN model also exhibits the best performance. Based on the obtained results, there are still some important points to consider. As mentioned earlier, AI- based models like ANN have certain advantages:

- Accelerated calculations
- Absence of strict pre-assumptions
- Compatibility with complex data structures

- User-friendly interfaces
- Strong fitting capabilities

However, these models also have some limitations:

- Prone to local optima (minima/maxima trap)
- Sensitive to parameter tuning
- Susceptible to early convergence/divergence during retraining
- Risk of overtraining

On the other hand, economic and econometric models have their own limitations as well. Prior to conducting regression analysis and forecasting models like ARMA, various assumptions such as linearity, normality, and stationarity should be evaluated. Failure to consider all the stages of the process may result in biased estimators rather than unbiased ones. Time series models primarily focus on historical data and may not effectively handle extreme jumps or drops, among other factors. Therefore, before performing any calculations, it is crucial to consider multiple factors such as data structure complexity, data size, parameter tuning, and other relevant conditions. By doing so, one can confidently select models or methods that align with their specific process and desired outcomes.

13. Conclusion

In this study, we aimed to predict the price of Ethereum using two types of models: I. AI-based methods and II. Econometric models. Economic and global indicators such as S&P500, NIFTY50, NASDAQ, etc., were employed as input variables. To identify the most important input variables, we utilized GA as a feature selection technique. While GA offers advantages such as adaptability to complex structures and lack of restrictive assumptions, it also has limitations including the potential removal of important indicators, time-consuming variable selection, early convergence, and other associated conditions. Furthermore, we employed ANN to find an optimal solution for ETH price forecasting in the next day. To ensure comparability and evaluate the performance of AI-based methods, we compared them with econometric models such as regression analysis and time series models. The results indicated that AI-based models, specifically ANN and GA, outperformed the econometric models, demonstrating higher predictability and offering advantages such as faster computation speed and adaptability to complex structures, among others. In summary, it is important to note that no single model or method can be deemed superior for forecasting purposes indefinitely. However, in the context of this study, the results suggest that AI-based models, namely ANN and GA, achieved better performance in predicting the ETH price.

Finally, we would like to highlight some important points and provide recommendations for future research:

One crucial and sensitive aspect of applying ANN is the tuning and parameter setting, including initial population, training rate, number of iterations, etc. These factors can significantly impact accuracy and predictability, and therefore, should be carefully considered.

As a recommendation for future research to obtain improved results, the utilization of other metaheuristic algorithms can be explored. These algorithms are powerful in both exploration and exploitation, potentially enhancing the search for optimal solutions and mitigating the risk of local optima/maxima traps.

References

Antonakakis, N., Chatziantoniou, I., & Gabauer, D. (2019). Cryptocurrency market contagion: market uncertainty, market complexity, and dynamic portfolios. *Journal of International Financial Markets, Institutions and Money*, *61*, 37-51.

Azadeh, A., & Farrokhi-Asl, H. (2019). The close–open mixed multi depot vehicle routing problem considering internal and external fleet of vehicles. *Transportation Letters*, *11*(2), 78-92.

Bai, A., & Vahedian, M. (2023). Beyond the Screen: Safeguarding Mental Health in the Digital Workplace Through Organizational Commitment and Ethical Environment. *Available at SSRN 4623055*.

Chang, V., Hall, K., Xu, Q. A., & Wang, Z. (2022). A social network analysis of two networks: Adolescent school network and Bitcoin trader network. *Decision Analytics Journal*, *3*, 100065.

Dapp, M. M. (2021). From Fiat to Crypto: The Present and Future of Money. In *Finance 4.0-Towards a Socio-Ecological Finance System* (pp. 1-25). Springer, Cham.

Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., & Li, L. (2022). Cryptocurrency trading: a comprehensive survey. *Financial Innovation*, *8*(1), 1-59.

Hamayel, M. J., & Owda, A. Y. (2021). A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. *AI*, *2*(4), 477-496.

Kasaie, A., & Rajendran, S. (2023). Integrating machine learning algorithms and explainable artificial intelligence approach for predicting patient unpunctuality in psychiatric clinics. Healthcare Analytics, *4*, 100242.

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091-8126.

Kim, H. M., Bock, G. W., & Lee, G. (2021). Predicting Ethereum prices with machine learning based on Blockchain information. *Expert Systems with Applications*, *184*, 115480.

Lai, C., Wang, Y., Fan, K., Cai, Q., Ye, Q., Pang, H., & Wu, X. (2022). An improved forecasting model of short-term electric load of papermaking enterprises for production line optimization. *Energy*, 123225.

Maleki, N., Nikoubin, A., Rabbani, M., & Zeinali, Y. (2020). Bitcoin price prediction based on other cryptocurrencies using machine learning and time series analysis. *Scientia Iranica*.

Naseri, H., Jahanbakhsh, H., Foomajd, A., Galustanian, N., Karimi, M. M., & D. Waygood, E. O. (2022). A newly developed hybrid method on pavement maintenance and rehabilitation optimization applying Whale Optimization Algorithm and random forest regression. *International Journal of Pavement Engineering,* 1-13.

Rathan, K., Sai, S. V., & Manikanta, T. S. (2019, April). Crypto-currency price prediction using decision tree and regression techniques. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 190-194). IEEE.

Shahvaroughi Farahani, M., & Razavi Hajiagha, S. H. (2021). Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. *Soft Computing, 25*(13), 8483-8513.

Shobana, G., & Umamaheswari, K. (2021, January). Forecasting by Machine Learning Techniques and Econometrics: A Review. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)* (pp. 1010-1016). IEEE.

Xu, X., & Zhang, Y. (2023). A high-frequency trading volume prediction model using neural networks. *Decision Analytics Journal*, *7*, 100235.

Appendix

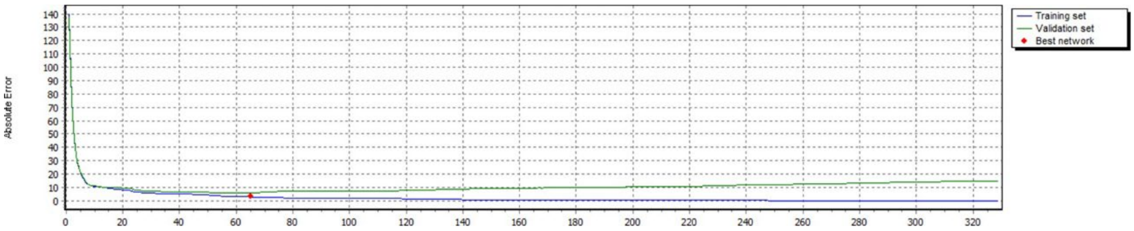

Fig. I. Training graph

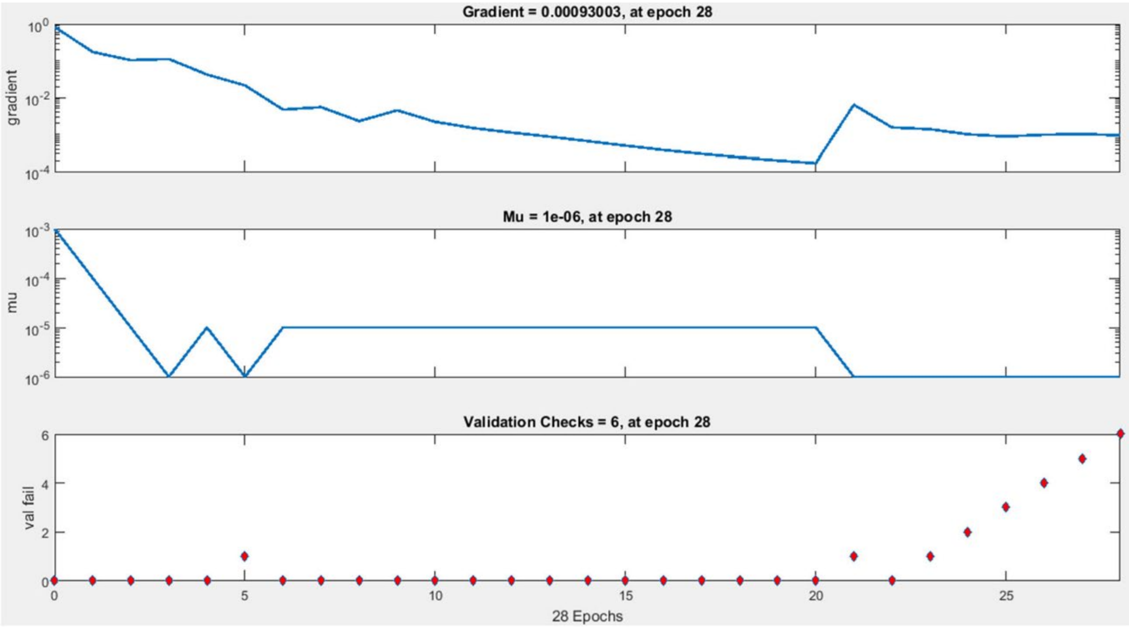

Fig. II. Training state

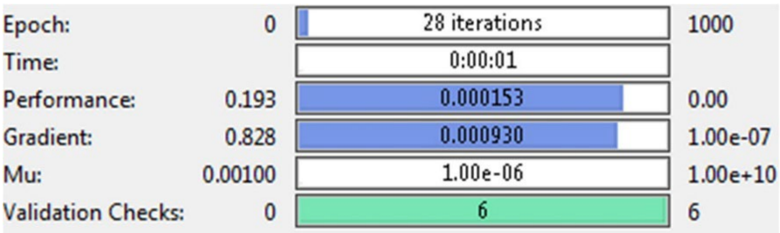

Fig. III. Progress

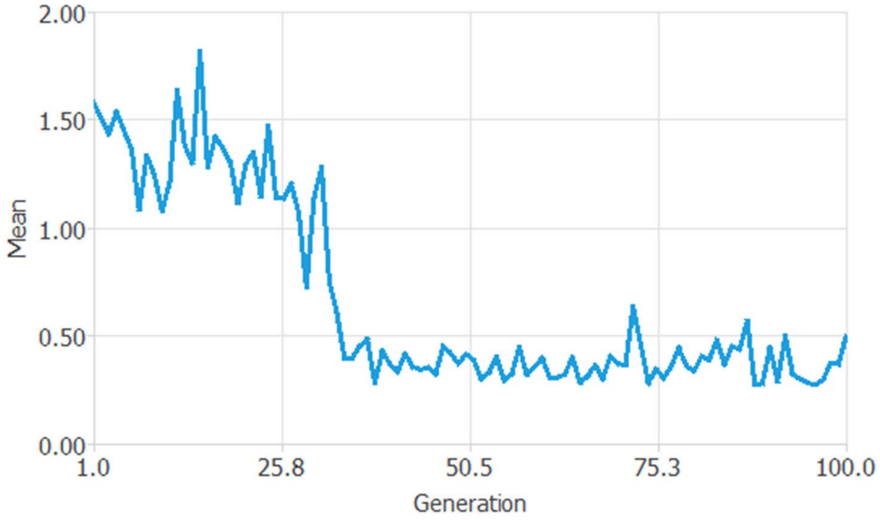

Fig. IV. GA error mean plot

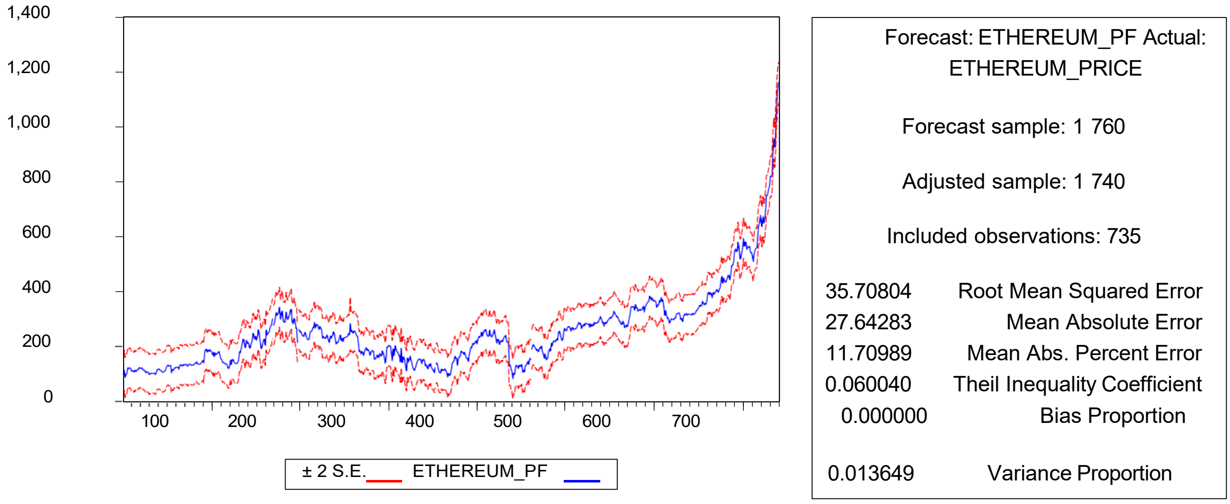

Fig. V. ETH-price forecasting