南京大学

FINAL REPORT

Topic: Ethereum Exchange Rate Prediction: Machine Learning Model Based on Economic and Technological Factors.

Subject: R Programming in Data Science

Professor: Professor Lele Kang (康乐乐)

Student Name: Do Minh Ngoc

Student ID: 502024145013

Table of Contents

List of Table

List of Figure

Abstract

Ethereum ($ETH) is more than just a cryptocurrency - it is the backbone of blockchain. Although Bitcoin paved the way for digital currencies, Ethereum introduced the concept of programmable blockchains through smart contracts, revolutionizing industries far beyond finance. This is why ETH stands out.

Ethereum significantly correlates with global financial assets such as crude oil, gold, and the US dollar. ETH and global financial assets have become more closely related, especially since the outbreak of the COVID-19 pandemic. This paper aims to evaluate ETH price prediction models with the support of global financial assets using deep learning models, such as Long Short Term Memory and Time Series Analysis, ARINA, and the SVR model.

Methodology/Approach: This paper proposes a more accurate prediction model for ETH trading by combining economic and technical factors models with the Long short-term memory model. This study develops a two-stage approach to investigate whether the information hidden in economic and technological determinants can accurately predict the ETH/USD exchange rate. By weighing the significance of technological and economic aspects, the first step employs Random Forest feature selection techniques to narrow down the pool of possible predictors.

In the second stage, the potential predictors are incorporated into a long short-term memory (LSTM) network to predict the ETH exchange rate, regardless of the previous exchange rate. Our results show LSTM can achieve better prediction performance using economic and technological determinants than SVR, ARINA, and LSTM methods using the previous exchange rate. Therefore, the information obtained from economic and technological determinants is more important for predicting the Ethereum exchange rate than the previous one.

1. Introduction

Ethereum, a decentralized, open-source blockchain technology, was the first network to include smart contracts. Unlike Bitcoin which was created as an alternative to national currencies, Ethereum was built as a platform that can facilitate programmatic smart contracts and applications using Ether.

In July 2015, Ethereum was introduced. Its value has increased dramatically in recent years as a result of the creation of numerous decentralized application (DApp) services. Ethereum is now the most popular blockchain network due to the rapid growth of decentralized finance (DeFi) and non-fungible tokens (NFTs).

Furthermore, Ether (ETH), its native currency, has steadily held onto its second-place ranking in the cryptocurrency industry regarding market capitalization and daily trading volume. Ethereum contributed 19.37% market value and ranked second in the crypto market, after Bitcoin (Statista, 2023)

Research on predicting Crypto exchange rates often focuses on statistical models (ARIMA, ...) (Azari, A. (2019)) but they can only handle linear problems and only focus on Bitcoin. Artificial intelligence (ANN, ...) (Aghashahi, M., & Bamdad, S. (2022). .) can learn complex patterns from big data but cannot exploit economic and technological factors that affect exchange rates. Previous research mainly focused on the impact of these factors but did not study their predictive ability. This study proposes combining economic and technological factors with LSTM networks to predict Ethereum exchange rates.

1.1. Research Question

Main Research Question:

- Can using economic and technological determinants improve the ability to predict Ethereum exchange rates compared to using only historical exchange rates?

More specific research:

- Which economic and technological factors significantly affect Ethereum exchange rates?
- Which machine learning method (e.g. LSTM, SVR, ARIMA) performs best in predicting Ethereum exchange rates?

1.2. Research Object and Scope

Research object: Ethereum price data from March 16, 2016 - March 14, 2023, and data on economic and technical factors affecting Ethereum price.

1.3. Research Method

This report proposes a two-stage approach:

- Random Forest feature selection to identify important economic and technical factors
- LSTM with these factors to predict Ethereum price and compare with traditional models (SVR, ARIMA, LSTM with previous price to predict).

The goal is to demonstrate that combining economic and technical factors improves the prediction ability compared to using only historical price data.

1.4. Research Structure

- Chapter 1: Introduction
- Chapter 2: Data Preparation
- Chapter 3: Methodology & Algorithms
- Chapter 4: Results
- Chapter 5: Conclusion, limitation and future work

2. Data Preparation

Table 1. List of economic and technological factors that can affect ETH price

	Description	Previous Research
Economic Factor	Macroeconomic indicators o Crude oil price o Gold price o NASDAQ	Gospodinov & Jamali, 2015
	Global currency ratio (USD) o EUR/USD o JPY/USD o CNY/USD	Dyhrberg, 2015; Jang & Lee, 2018; Kristjanpoller & Minutolo, 2018
Technology Factor	o Blocksize o Blocktime o Transaction Fee	W. Chen, H. Xu, L. Jia et al (2021)

	○ Transaction volume ○ Transaction Value ○ Market capitalization ○ Gas Limit ○ BTC price	
	○ Public Attention (Volume of Tweet on Twitter)	Dastgir, Demir, Downing, Gozgor, & Chi, 2018; Polasik et al., 2015; Zhang, Wang, Li, & Shen, 2018

Source: Synthesis of previous studies

2.1. Technical and Blockchain Data

The website bitinfocharts.com and etherscan.io provides detailed historical data on many aspects of the Ethereum blockchain network, including:

- **Price of Ethereum:** The ETH/USD price represents the value of one unit of Ethereum (ETH) expressed in US Dollars (USD). It shows how many US dollars it takes to buy one Ethereum or how many US dollars you will receive when you sell one Ethereum.
- **Ethereum Mention Tweets** is a metric that measures how often Ethereum is mentioned on the social media platform Twitter. It shows the number of tweets containing keywords related to Ethereum over a given period of time.
- **Ethereum's "transaction value"** refers to the total value of all transactions processed on the Ethereum network over a given period of time. It is usually expressed in a currency such as USD or in ETH itself.
- **Number of Transactions**: The number of transactions recorded on the blockchain over a given period.
- **Average Transaction Fee:** The average fee paid by users for each transaction.
- **Block Size**: The average size of blocks added to the blockchain.
- **Block Time:** The average time it takes to create a new block.
- **Market Cap**: Ethereum's market capitalization.
- Gas Limit refers to the maximum amount of computational work or resources that can be used for a transaction or contract execution on the Ethereum network. It determines how much “gas” (a measure of computational e!ort) is allowed for a transaction or smart

contract. A higher gas limit allows for more complex transactions or contracts but also increases transaction costs.

- BTC Price represents the market value of Bitcoin (BTC), the first and most well-known cryptocurrency. It is typically quoted in terms of USD or other fiat currencies and serves as a key indicator of Bitcoin's value in the broader cryptocurrency market. The BTC price fluctuates based on demand, market sentiment, and external economic factors.

Data from bitinfocharts.com and etherscan provides a better understanding of the health and performance of the Ethereum network, which can affect the value of Ether (ETH).

2.2. Economic Data

Investing.com is a financial platform that provides global market data, including:

- Exchange Rates: Exchange rates between currencies (e.g. CNY/USD, EUR/USD, JPY/USD).
- Gold Price: International gold prices.
- Oil Price: Crude oil prices.
- Stock Indices: For example, Nasdaq.

Data from investing.com reflects macroeconomic conditions and traditional market factors that can influence investor sentiment and, therefore, impact the value of digital assets such as Ether.

3. Methodology and Algorithms

3.1. Algorithms

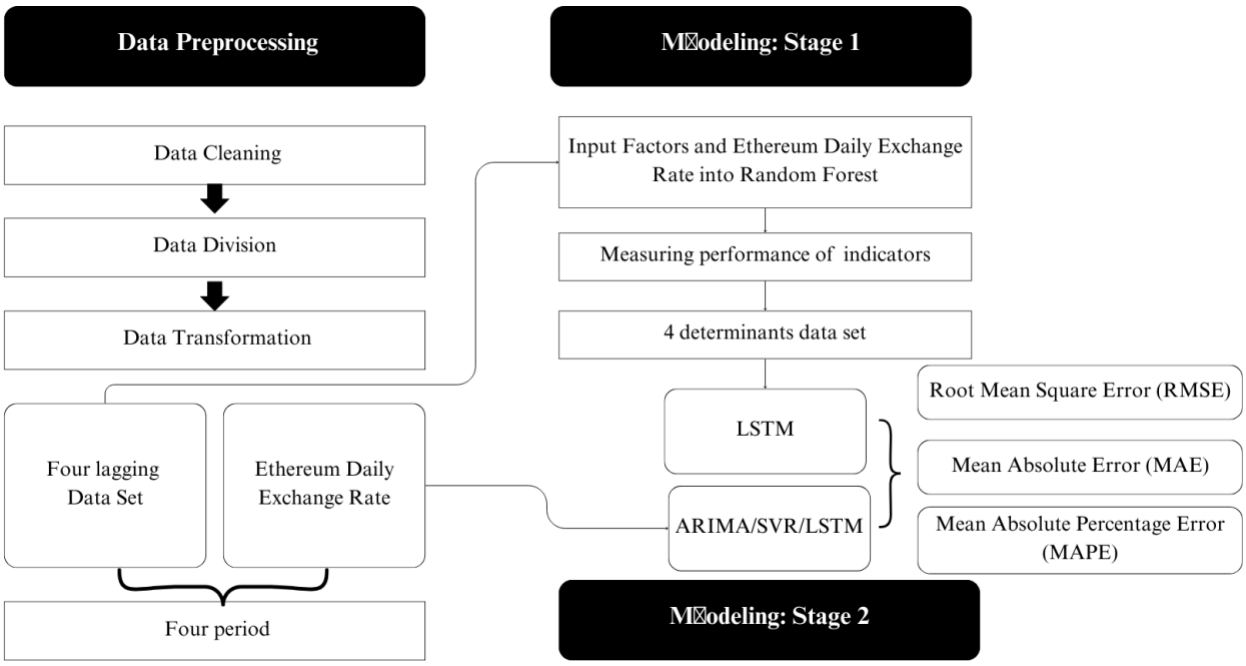

Figure 1. Algorithms employed in this report

Stage 1: Feature Selection with Random Forest

Goal: Identify the most important economic and technical factors influencing the Ethereum price. This helps reduce the number of input variables for the LSTM model in phase 2, avoid overfitting, and improve performance.

Method: Use **the Random Forest and ANN** algorithm.

Evaluating the importance of variables:

- **%IncMSE (Percentage Increase in Mean Squared Error):** The percentage increase in the Mean Squared Error when a variable is removed from the model. The higher the % of IncMSE, the more important the variable is.
- **IncNodePurity (Increase in Node Purity)**: The degree to which a variable contributes to reducing the impurity of nodes in the decision tree. The higher the IncNodePurity, the more effectively the variable separates the data.

- COE (Change in Output Error) measures the importance of an input factor in an ANN by evaluating the change in mean squared error (MSE) when that input is removed. It is defined as the difference between the MSE with and without the input.

Results: This stage will produce a list of economic and technical factors ranked by importance. Only the most important factors (based on %IncMSE and/or IncNodePurity with a certain threshold) will be included in stage 2.

Stage 2: Predicting Ethereum Price with LSTM and Comparing Performance

Objective: Build a Ethereum price prediction model using LSTM using the factors selected in stage 1 and compare its performance with other models.

Method:

- **LSTM with economic and technical factors:** This LSTM model will take as input the time series of economic and technical factors selected in stage 1.

Comparison models:

- LSTM using only historical prices: This LSTM model only uses the time series of historical Ethereum prices as input.
- ARIMA (Autoregressive Integrated Moving Average): Traditional statistical model commonly used for time series prediction.
- SVR (Support Vector Regression): Machine learning model that uses the support vector machine for regression problems.

Performance evaluation: Uses the following three metrics:

- RMSE (Root Mean Squared Error): The square root of the average of the squared errors. The lower the RMSE, the more accurate the model. Sensitive to outliers.
- MAE (Mean Absolute Error): The average of the absolute errors. The lower the MAE, the more accurate the model. Less sensitive to outliers than RMSE.
- MAPE (Mean Absolute Percentage Error): The average of the absolute percentage errors. The lower the MAPE, the more accurate the model. It is easy to compare different data sets.
- DA (Directional Accuracy) is a metric used to evaluate how well a model predicts the direction of changes in a variable (e.g., stock prices, temperature, etc.). It measures the percentage of time the model correctly predicts the direction of change (up or down) compared to the actual values.

3.2. Methodology

3.2.1. Data Cleaning

The data used in this study was collected through web scraping using Python, so some values will inevitably be missing. Due to the nature of time series data, I applied a linear interpolation method to handle these missing values. **Linear interpolation** estimates the missing value by creating a straight line between two adjacent data points with values. This method is beneficial for time series data because it preserves the trend and continuity of the data. Outliers were handled using the space symbol procedure proposed by Serneels, De, and Van Espen (2006), by removing the corresponding data points.

3.2.2. Data division

The data in this window is divided into two parts:

- Training set: Includes data from the start date of the window to the end date of training (end_train). The end date of training is the day before the end date of the window.
- Testing set: Includes data from the day after the end date of training (start_test) to the end date of the window (end_test). To ensure that the end date of the data is not exceeded, end_test is taken as the smallest value between the end date of the window and end_date_data.

After processing a window, the window will be slid forward by a period equal to step_size_years (1 year). This process is repeated until the sliding window exceeds end_date_data.

Using a rolling window allows us to evaluate the model's performance over different periods, thereby helping to detect problems such as overfitting or underfitting and assessing the stability of the model over time.

By training on past data and testing on future data, this method simulates how the model will perform in practice.

In summary, the rolling window method is used to create multiple pairs of training and testing sets, each pair representing a different period, providing a more comprehensive evaluation of the predictive model's performance.

3.2.3. Data Scaling

For the model to work well, the input data must be processed to avoid the situation where variables with large values "squeeze" variables with small values, as Rezakazemi et al. (2013) recommended. In other words, I must bring the variables to the same "scale" for equal roles in the model training process.

Therefore, this study used the Min-Max normalization method to "balance" the input variables. The Min-Max normalization formula is as follows:

$$x'_i = (x_i - \min(X)) / (\max(X) - \min(X))$$

This formula will "shrink" all values of a variable to the range from 0 to 1. Thanks to that, variables with different units of measurement and different values will have equal roles when entered into the model.

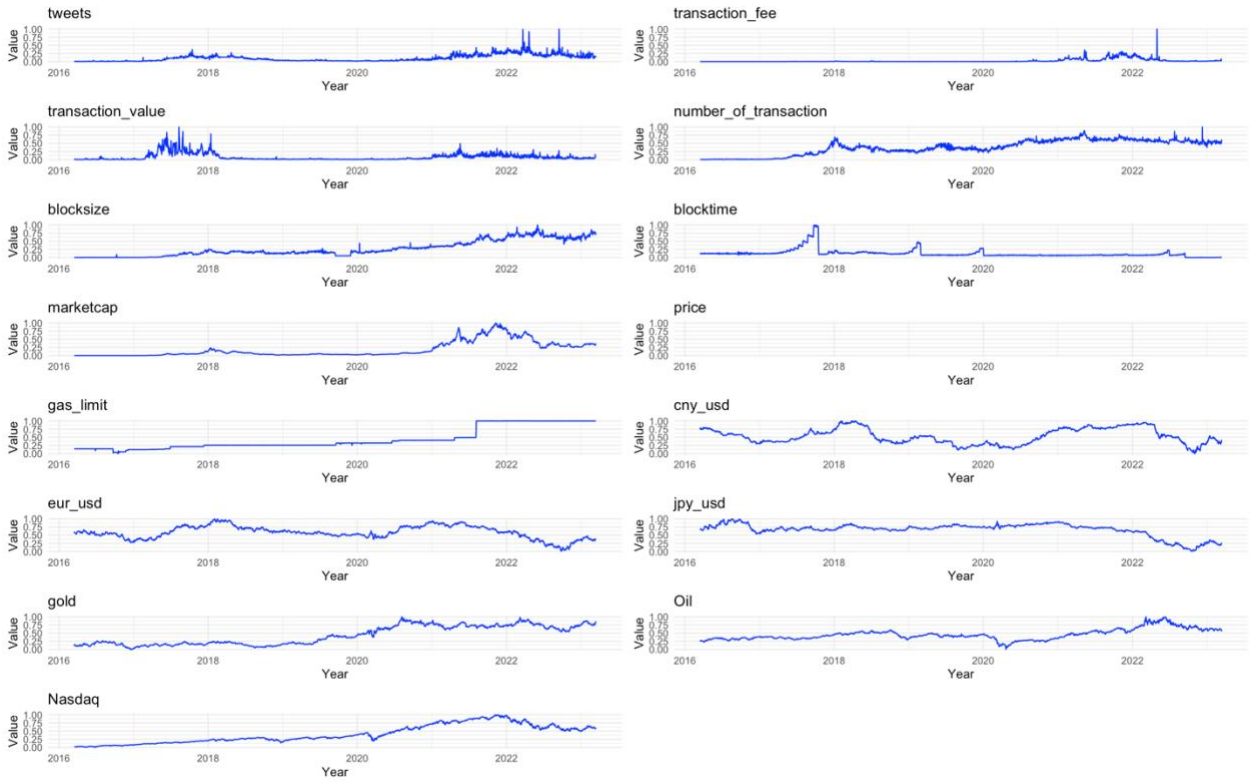

Figure 2. Line Chart of Scaled Data

(Source: Result from R studio)

3.2.4. Data Modeling Stage 1

The economic and technological factors selected in this study were mainly based on previous statistical analyses. However, their " machine " integration into Deep Learning models may lead to poor predictive performance.

Therefore, in Stage I, I used feature selection methods to identify the factors with the highest predictive value before removing the “redundant” factors. This first phase is divided into three sub-steps as follows:

Using two feature “filtering” models: We used two machine learning models, Random Forest (RF). Previous studies have shown that these two models have good feature “filtering” capabilities (Enke & Thawornwong, 2005; Tsai & Hsiao, 2011).

- **Factor Importance Assessment:** I used sensitivity analysis to measure the importance of each factor (Dag, Oztekin, Yucel, Bulur, & Megahed, 2016). The importance values of the variables measured by RF and ANN were then normalized to a scale of 0 to 1.

- **Combining Feature Selection Methods:** As Tsai and Hsiao (2011) pointed out, combining multiple feature selection methods often performs better than using only one method. Therefore, in this study, I used a "crossover strategy" to filter out unrepresentative variables and generate a final set of "candidate" features for stage II of the modelling process. The caret package in R's statistical language was used to build the RF models (Kuhn, 2015).

Random Forest

Random Forest is an ensemble machine-learning algorithm. It uses bootstrap and random node-splitting techniques to build multiple independent decision trees. The final prediction results are aggregated by "voting" from the member trees.

A Random Forest is more robust to noisy and missing data than a single regression tree. It can also analyse complex interacting categorical features (Breiman, 2001; Svetnik et al., 2003). In Random Forest, the importance of a factor x can be calculated as follows (Lahouar & Slama, 2017):

(i) Calculating the out-of-bag error (OOBE1): For each decision tree in Random Forest, a portion of the data that was not used to build the tree (called the "out-of-bag" data) will be used to calculate the prediction error. This error is called OOBE1. The formula for calculating OOBE is as follows:

$$OOBE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i),$$

Where:

- Yi is the observed value.
- $\hat{Y}i$ is the predicted value.
- n is the number of data samples.

(ii) Add noise and calculate OOBE2: Next, randomly add noise to the x factor in the out-of-bag data and calculate the OOBE again (this time called OOBE2).

IncMSE and IncNodePurity

As mentioned, Random Forest can assess the importance of each input variable. Here are two standard measures used for this purpose:

%IncMSE (Percentage Increase in Mean Squared Error): This measure represents the percentage increase in the mean squared error (MSE) when a particular variable is removed from the model. The higher the % of IncMSE, the more important that variable is.

How to calculate:

- Calculate the OOBE (out-of-bag error) for the original model (OOBE1).
- For each variable, shuffle the value of that variable in the OOB data and calculate the OOBE again (OOBE2).
- %IncMSE for that variable is calculated using the formula: ((OOBE2 - OOBE1) / OOBE1) * 100%

IncNodePurity (Increase in Node Purity): This measure represents how much a variable contributes to reducing the "impurity" of the nodes in the decision tree. The higher the IncNodePurity, the more important the variable is.

ANN: (Artificial Neural Network)

ANN can be used to measure the importance of an input factor xxx by assessing the change in mean squared error (COE) when the corresponding input is removed from the network (Xu, Wong, & Chin, 2013). COE is defined as:

$$COE = I(x) = MSE_2 - MSE_1$$

Where:

$$MSE = \frac{1}{P}\sum_{p=1}^{P}(Y_i - \hat{Y}_i)^2$$

MSE_1 and MSE_2 : represent the mean squared error with and without x x, respectively. For an introduction to ANNs, refer to the studies by Elhewy, Mesbahi, and Pu (2006) and Gomes and Awruch (2004). In this study, we used a multi-layer feed-forward neural network trained by back-propagation.

3.2.5. Data Modeling Stage 2

3.2.5.1. Long short term memory

LSTM (Long Short-Term Memory), proposed by Hochreiter and Schmidhuber (1997), is one of the most popular models in recurrent neural networks (RNNs). LSTM excels in time series

prediction because of its ability to handle multivariate or multi-input prediction problems, where classical linear methods struggle.

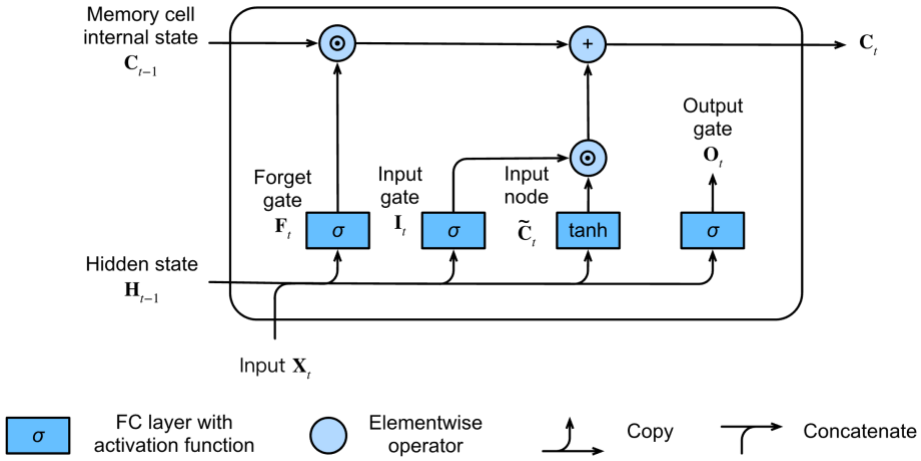

Figure 3. LSTM structure

At each time step t, each memory cell has three gates, which are responsible for maintaining and adjusting the cell state (*st)*

- Forget gate *(ft):* Decides what information will be "forgotten" from the previous cell state.
- Input gate *(it):* Controls what new information will be fed into the memory cell.
- Output gate *(ot):* Controls what information from the memory cell will be fed out.

Structure and operation of LSTM At each time step t, the cell state $s\ t$ is updated through the above gates, with the specific formula as follows:

Input Gate.

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i)$$

This gate uses a sigmoid function (σ) to determine how much information from the current input $x\ t$ and the previous hidden state $h\ t-1$ needs to be stored in the memory cell.

Forget Gate

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f)$$

This gate decides what percentage of information from the previous cell state $s\ t-1$ will be retained or discarded.

Output Gate

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o)$$

This gate determines what percentage of information from the current cell state $s\,t$ is used as output.

Due to the tiny sample size (around 2554), I employed a basic LSTM structure consisting of an input, hidden, and output layer. In the regression layer, I used dropout to lower the chance of overfitting. We also employed early stopping to automatically calculate the ideal number of training epochs for every study period. **Tensorflow and the R Keras package** were used to create the LSTM model.

3.2.5.2. Support Vector Regression (SVR)

Support Vector Regression (SVR) is a robust machine learning algorithm widely used in regression problems, including continuous value prediction such as Ethereum price. SVR is notable for its ability to handle non-linear data and operate efficiently in high-dimensional spaces, which is especially important when considering factors affecting Ethereum price.

- **Non-linear data handling ability:** SVR uses kernel functions to map data into a higher-dimensional space, where non-linear relationships can be represented linearly. This allows SVR to capture complex relationships between past and future Ethereum prices.
- **High-dimensional efficiency:** SVR can handle data with many features (e.g. Ethereum prices at different points in the past), allowing the model to consider many factors affecting Ethereum price.
- **Overfitting tolerance:** SVR uses a regression mechanism to reduce the risk of overfitting, helping the model generalize better and predict accurately on new data.

This study used the **e1071** package in **R** to build and train the SVR model. The e1071 package provides a powerful and flexible svm() function, which allows us to customize the parameters of the SVR model, including choosing the kernel function appropriate to the data's characteristics.

3.2.5.3. Autoregressive Integrated Moving Average (ARIMA)

The Autoregressive Integrated Moving Average (ARIMA) model is a powerful statistical tool widely used for time series prediction. ARIMA combines autoregressive (AR), differenced (I) and moving average (MA) components to model data and predict future values.

Components of ARIMA

- Autoregressive (AR): Uses past values of a time series to predict current values.
- Differenced (I): Corrects the stationarity of a time series by taking differences in the data.
- Moving average (MA): Uses past prediction errors to predict current values.

Application of ARIMA in Ethereum Price Prediction

In the Ethereum price prediction problem, ARIMA can be used to:

- **Modelling trends and seasonality:** ARIMA can capture the up-and-down trends and recurring cycles in Ethereum price data over time.
- **Predicting future Ethereum prices:** Based on the learned patterns, ARIMA can predict Ethereum prices in different periods (e.g., daily, weekly, monthly).

R packages used: In this study, I used the forecast package in R to build and train the ARIMA model. The forecast package provides powerful and convenient functions for analyzing, modelling, and predicting time series.

4. Result

4.1. Data Modeling Stage 1

Random Forest: Provides information on the importance of input variables. A high correlation (0.979) shows that the model fits the data well.

- **Market cap** is **the most important factor**, shown by the highest % of IncMSE and IncNodePurity. This shows that market cap has a significant influence on Ethereum price.
- **BTC Price** are also **important factors**, especially in terms of %IncMSE, showing that they significantly influence the accuracy of the model when removed.

Overall, technological (Marketcap, Gas Limit, BTC Price, Blocksize, Transaction Fee) and economic (Nasdaq, cny_usd, eur_usd) factors play a more important role than other factors in predicting Ethereum price.

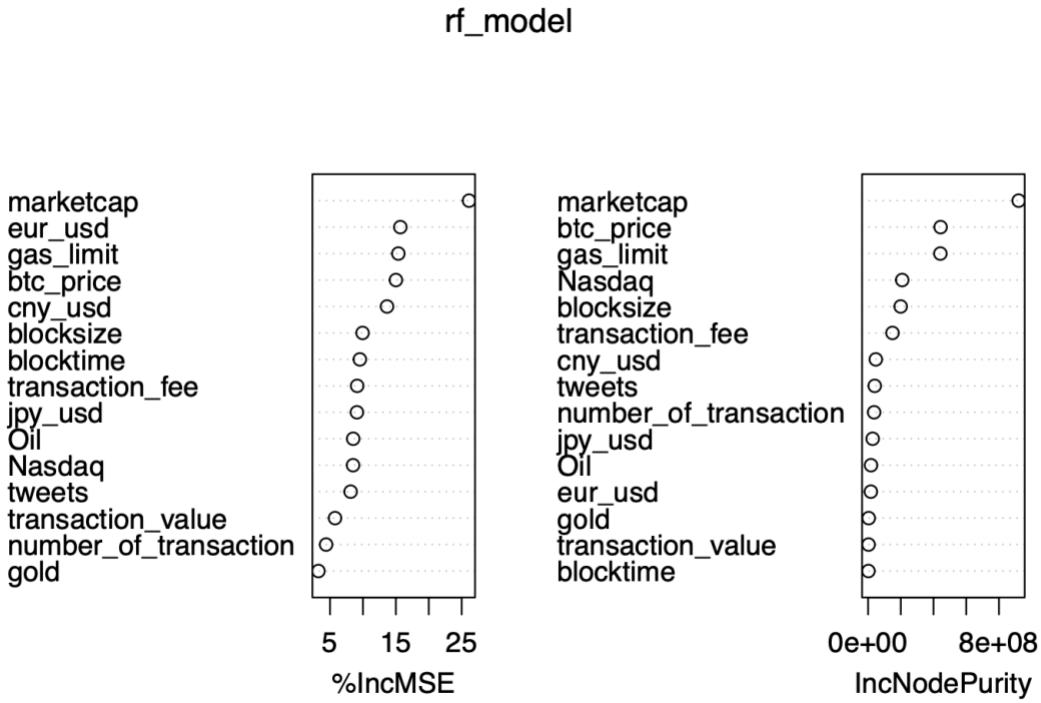

Figure 4. Importance of various factors on ETH price

Source: Rstudio & Excel

Table 2. COE from ANN model results

Input Feature	**COE (Delta MSE)**
tweets	-243815.579
number_of_transaction	865019.752
marketcap	-97623.163
eur_usd	865019.752
Oil	865019.753
transaction_fee	7623.929
transaction_value	-146228.011
blocksize	-49791.821
gas_limit	-87310.752
jpy_usd	-88946.822
blocktime	-275060.947

cny_usd	-255689.467
gold	-20095.421
btc_price	-352915.325
Nasdaq	-44763.326

4.2. Data Modeling Stage 2

The report used the following four metrics to evaluate the predictive ability of the models:

RMSE (Root Mean Squared Error)

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}$$

Measures the average deviation between the predicted value ($y \wedge t$) and the actual value ($y\ t$). The lower the RMSE, the better the model.

MAPE (Mean Absolute Percentage Error)

$$MAPE = \left(\frac{1}{n}\sum_{t=1}^{n}\left|\frac{\hat{y}_t - y_t}{y_t}\right|\right) \times 100$$

Measures the average percentage deviation between the predicted value and the actual value. The lower the MAPE, the better the model.

MAE (Mean Absolute Error)

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\hat{y}_t - y_t|$$

Measures the average absolute deviation between the predicted value and the actual value. The lower the MAE, the better the model.

4.2.1. Parameter for each model

This table shows the parameters used for each model (LSTM, SVR and ARIMA) during training and predicting the Ethereum price. Choosing and tuning these parameters is very important to ensure the best performance of the model.

Table 3. Parameters for different models

Model	Parameter	Value
LSTM	Input (Lags) Activation Function Learning Rate LSTM Nodes in Hidden Layer	30 Sigmoid (and tanh) 0.001 50
SVR	Input (Lags) G* C* ϵ*	Price_lag1 & Price_lag2 0.1 1 0.1
ARIMA	P* D* Q*	30 1 1

4.2.2. Performance of Each model

Performance in terms of RMSE

Table 4. Performance in terms of RMSE

Model	RMSE_mean	RMSE_sd	RMSE_min	RMSE_max
ARIMA	1343.0943	0.000000	1343.0943	1343.0943
SVR	158.4071	0.000000	158.4071	158.4071
LSTM (old_price)	1406.5838	0.748246	1405.6489	1407.7895
LSTM (selected_feature)	**148.7023**	4.786485	142.9106	157.0515

Source: Rstudio

The **LSTM (selected_feature)** model, however, has a much lower RMSE mean of **148.70**, which is better than **ARIMA** (1343.09) and **SVR** (158.41). This shows that LSTM can outperform traditional models like ARIMA and SVR when the correct features are selected for training.

Performance in terms of MAE

Table 5. Performance in terms of MAE

Model	MAE_mean	MAE_sd	MAE_min	MAE_max
ARIMA	1226.23994	0.000000	1226.23994	1226.23994
SVR	89.77589	0.000000	89.77589	89.77589
LSTM (old_price)	871.60192	1.036438	870.34502	873.24972
LSTM (selected_feature)	**84.47969**	2.807826	80.58696	89.50391

Source: Rstudio

LSTM (selected_feature) has the lowest **MAE**_mean of 84.47969, indicating that this model provides the best performance in terms of the average error..

Performance in terms of MAPE

Table 6. Performance in terms of MAPE

Model	MAPE_mean	MAPE_sd	MAPE_min	MAPE_max
ARIMA	2489.0273	0.000000	2489.0273	2489.02734
SVR	173.2521	0.000000	173.25207	173.25207
LSTM (old_price)	83.4101	0.361255	83.02235	83.97215
LSTM (selected_feature)	**20.5481**	2.626956	17.52631	26.37118

Source: Rstudio

LSTM (selected_feature) has the lowest **MAPE_mean** of **20.5481**, indicating the most accurate model for predicting percentage errors.

Performance in terms of DA

Table 7. Performance in terms of DA

Model	DA_mean	DA_sd	DA_min	DA_max
ARIMA	0.1972387	0.000000	0.1972387	0.1972387

SVR	53.4516765	0.000000	53.4516765	53.4516765
LSTM (old_price)	43.9489190	4.136689	37.5245580	49.7053050
LSTM (selected_feature)	**53.7708330**	2.828444	47.5000000	56.8750000

Source: Rstudio

LSTM (selected_feature) has the highest DA_mean of 53.7708330, showing that it has the best performance in terms of directional accuracy.

Summary

All four statistical indicators - minimum, maximum, mean, and standard deviation - performed better in the RMSE, MAPE, MAE, DA outcomes derived from the LSTM model employing economic and technological predictors. Put differently, when compared to models that solely used historical exchange rates, the LSTM model including predictors demonstrated better prediction performance in terms of accuracy, stability, and robustness.

5. Conclusion, limitation and future work

5.1. Conclusion

This study explored the possibility of predicting Ethereum exchange rates using economic and technological factors, compared to the traditional method based on historical exchange rate data alone. The results of the study answered three main questions:

- ***Does the use of economic and technological factors improve the ability to predict Ethereum exchange rates compared to using historical exchange rate data alone?***

The results clearly show that the LSTM model trained with economic and technological factors significantly outperforms the models using historical exchange rate data alone (LSTM, SVR, and ARIMA). This confirms the importance of combining information from different sources to predict exchange rates.

- ***Which economic and technological factors significantly affect Ethereum exchange rates?***

The study identified eight factors that significantly influence the Ethereum exchange rate: 'marketcap', 'gas_limit', 'eur_usd', 'btc_price', 'blocksize', 'transaction_fee', 'Nasdaq', and 'cny_usd',…These factors include both macroeconomic factors (e.g.,nasdaq, eur_usd, cny_usd) and blockchain technology-related factors (e.g., block size, transaction fees).

- ***Which machine learning method (e.g., LSTM, SVR, ARIMA) performs best in predicting the Ethereum exchange rate?***

The results show that the LSTM model achieves the best prediction performance compared to SVR and ARIMA. The ability of LSTM to handle time series data and learn complex relationships may be a key factor in its superiority.

5.2. Limitations

This study has several limitations that should be considered:

Model complexity: LSTM models can be complex and require a lot of data to train effectively. The choice of network architecture and hyperparameters can affect the performance of the model.

Overfitting: The risk of overfitting is a concern when training complex models such as LSTMs. Although the study used **regularization** techniques to minimize overfitting, it is still necessary to carefully evaluate the generalization ability of the model on new data.

Non-stationarity: Exchange rate data is often stationary, i.e. the mean and variance change over time. Handling the stationarity of the data can affect the prediction results.

Impact of external shocks: Unexpected events such as political statements or global economic fluctuations can cause shocks that affect the Ethereum exchange rate, and the model may not be able to predict these effects.

5.3. Future Research Prospects

This study can be extended in several directions:

- **Adding economic and technological variables**: The study can add other economic and technological variables to examine their impact on the Ethereum exchange rate. Collecting and processing data for these variables can be challenging, but can provide valuable information for the model.
- **Researching other models**: In addition to LSTM, SVR, and ARIMA, there are many other machine learning models that can be applied to the Ethereum exchange rate prediction problem, such as Transformer Networks or hybrid models. Comparing the performance of different models can help find the best method.
- **Considering market sentiment factors**: Market sentiment can play an important role in Ethereum price fluctuations. Incorporating market sentiment factors into the model can improve the prediction ability.
- **Developing short-term and long-term predictive models**: Research may focus on developing predictive models for different time periods, from short-term (e.g., daily, weekly) to long-term (e.g., monthly, yearly).

Reference

5. (2020). Deep Reinforcement Learning Based Optimal Route and Charging Station Selection. Energies, 13(23), 6255.
6. AI vs ANN: Understanding the Difference and Applications. https://aquariusai.ca/blog/a-comparison-of-artificial-intelligence-and-artificial-neural-networks-in-the-context-of-machine-learning-and-data-analysis
7. Asperti, A., Asperti, A., Asperti, A., Raciti, G., Raciti, G., Ronchieri, E., & Cesini, D. (2025). Machine Learning-Based Anomaly Prediction for Proactive Monitoring in Data Centers: A Case Study on INFN-CNAF. Applied Sciences, 15(2), 655.
8. Can you trade stocks using Investing.com?. https://app.fintrakk.com/article/can-you-trade-stocks-using-investing-com
9. Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, 37, 28-43. https://doi.org/10.1016/j.ijforecast.2020.02.008.
10. Collenteur, R., Haaf, E., Bakker, M., Liesch, T., Wunsch, A., Soonthornrangsan, J., White, J., Martin, N., Hugman, R., De Sousa, E., Fan, X., Peterson, T., Bikše, J., Wang, X., Yang, Z., Nölscher, M., Koch, J., Schneider, R., Massei, N., . . . Meysami, R. (2024). Data-driven modelling of hydraulic-head time series: Results and lessons learned from the 2022 Groundwater Time Series Modelling Challenge. Hydrology and Earth System Sciences, 28(23), 5193-5208.
11. Cui, X., Lu, J., & Han, Y. (2024). Remaining Useful Life Prediction for Two-Phase Nonlinear Degrading Systems with Three-Source Variability. Sensors, 24(1), 165.
12. Decentralized management of digital assets | by 3Commas Blog | Medium. https://3commastutorials.medium.com/decentralized-management-of-digital-assets-7aa08a7badf6
13. Dimitriadou, A., & Gregoriou, A. (2023). Predicting Bitcoin Prices Using Machine Learning. *Entropy*, 25. https://doi.org/10.3390/e25050777.
14. Ethereum's Smart Contracts: What Are They And How Do They Work?. https://chaindebrief.com/ethereum-smart-contracts/

15. Liu, M., Li, G., Li, J., Zhu, X., & Yao, Y. (2020). Forecasting the price of Bitcoin using deep learning. *Finance Research Letters*, 101755. https://doi.org/10.1016/j.frl.2020.101755.
16. Machine Learning Algorithms for Neural Networks - reason.town. https://reason.town/machine-learning-algorithms-neural-networks/
17. Mallqui, D., & Fernandes, R. (2019). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Appl. Soft Comput.*, 75, 596-606. https://doi.org/10.1016/j.asoc.2018.11.038.
18. Rebari, P. K., & Killi, B. R. (2023). Deep Learning Based Traffic Prediction for Resource Allocation in Multi-Tenant Virtualized 5G Networks. https://doi.org/10.1109/tencon58879.2023.10322446
19. Support Vector Machine: Unveiling the Power of Machine Learning Classification. https://www.thebestfashion.co/2023/08/support-vector-machine-unveiling-the-power-of-machine-learning-classification/
20. Tan, Y., Li, Y., Gu, Y., Liu, W., Fang, J., Pan, C., & Pan, C. (2024). Numerical Study on Heat Generation Characteristics of Charge and Discharge Cycle of the Lithium-Ion Battery. Energies, 17(1), 178.
21. Török, D., & Ageyeva, T. (2022). Machine Learning in Injection Molding: An Industry 4.0 Method of Quality Prediction. Sensors, 22(7), 2704.
22. Wang, K., Hou, W., Hou, W., Ma, H., & Hong, L. (2024). Eye-Tracking Characteristics: Unveiling Trust Calibration States in Automated Supervisory Control Tasks. Sensors, 24(24), 7946.
23. Wang, K., Hou, W., Hou, W., Ma, H., & Hong, L. (2024). Eye-Tracking Characteristics: Unveiling Trust Calibration States in Automated Supervisory Control Tasks. Sensors, 24(24), 7946.
24. Wiliani, N., Hesananda, R., Rahmawati, N., & Prianggara, E. (2022). APPLICATION OF MACHINE LEARNING FOR BITCOIN EXCHANGE RATE PREDICTION AGAINST US DOLLAR. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*. https://doi.org/10.33480/jitk.v7i2.2880.