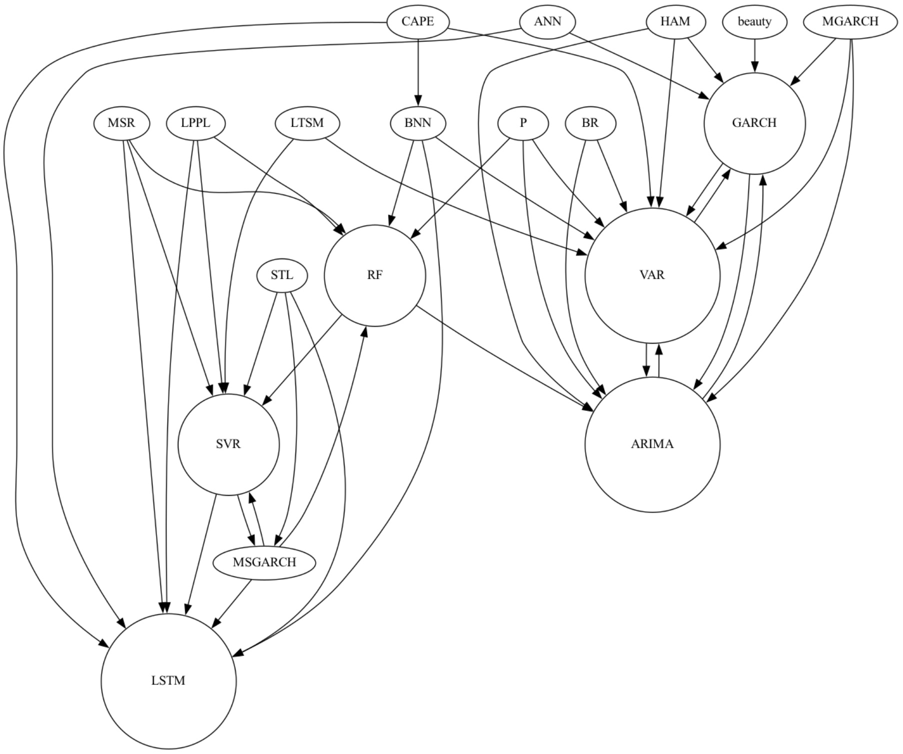
Integrating Interdisciplinary Models for Precise Forecasting of Bitcoin Market Prices: A Fusion of Financial, Economic, Time-Series, and Machine and Deep Learning Approaches

Boulouma M. A., 2023

Abstract

This research aims to develop a highly accurate predictive model for forecasting Bitcoin market prices by exploring various financial, economic, time series, mathematical and machine and deep learning models such as the Beauty Contest model of Keynes, Artificial Neural Networks, Bayesian regression, Vector Autoregression (VAR) model, ARIMA model, GARCH model, Markov regime-switching model, Support Vector Regression (SVR) model, The Log-Periodic Power Law (LPPL), Multivariate GARCH (MGARCH) models, Bayesian Neural Networks (BNN), Long Short-Term Memory (LSTM) models, Heterogeneous agent model, Shiller's CAPE Ratio, Markov switching GARCH (MSGARCH) models, and Random Forest model. By combining these models, we aim to provide an accurate forecast of Bitcoin market prices. The research question is: What combination of models yields the highest predictive power for forecasting Bitcoin market prices? The proposed model will have significant implications for investors, traders, and policymakers who rely on accurate market forecasts for decision-making.



Introduction

The cryptocurrency market has emerged as a new and dynamic asset class, and Bitcoin has emerged as the most prominent cryptocurrency. The unprecedented growth of the Bitcoin market has led to an increase in its popularity among investors, traders, and policymakers. As a result, the accurate forecasting of Bitcoin market prices has become a critical issue in finance and economics. The market prices of

Bitcoin are volatile, and traditional financial and economic models fail to provide precise forecasts. Therefore, it is essential to integrate various financial, economic, time-series, and machine and deep learning models to develop a highly accurate predictive model for forecasting Bitcoin market prices.

This research aims to integrate various financial, economic, time-series, and machine and deep learning models to develop a precise forecasting model for Bitcoin market prices. The research will explore several models, including the Beauty Contest model of Keynes, Artificial Neural Networks, Bayesian regression, Vector Autoregression (VAR) model, ARIMA model, GARCH model, Markov regime-switching model, Support Vector Regression (SVR) model, The Log-Periodic Power Law (LPPL), Multivariate GARCH (MGARCH) models, Bayesian Neural Networks (BNN), Long Short-Term Memory (LSTM) models, Heterogeneous agent model, Shiller's CAPE Ratio, Markov switching GARCH (MSGARCH) models, and Random Forest model. The integration of these models will provide a more accurate forecast of Bitcoin market prices, benefiting investors, traders, and policymakers.

The proposed research will contribute to the literature by exploring the predictive power of various models and the combination of these models. The study will provide insights into the strengths and weaknesses of each model and their contributions to the forecasting of Bitcoin market prices. The results of this research will have significant implications for investors, traders, and policymakers, as they rely on accurate market forecasts for decision-making. Moreover, the study will provide a roadmap for future research in the field of Bitcoin market price forecasting.

The remainder of this paper is structured as follows. The next section provides an overview of the various models explored in this research. Each model is discussed in detail, highlighting its theoretical background, its strengths, and weaknesses. The subsequent section provides an analysis of the integration of these models and their predictive power for forecasting Bitcoin market prices. Finally, the paper concludes with a discussion of the implications of this research and the future directions for research in the field of Bitcoin market price forecasting.

Data and preliminary analysis

We collected Bitcoin (BTC) data from the Coin Metrics Community Network Data repository, which provides access to high-quality data with a range of Bitcoin-related metrics. The dataset contains 10.3 MB of daily data from January 3, 2009, to February 25, 2023, and includes 181 features, such as time, AdrActCnt, AdrBalCnt, HashRate, IssTotNtv, PriceBTC, ROI30d, and TxTfrValAdjUSD. We use these features to build various indicators related to the cryptocurrency market.

In addition to the BTC data, we also collected Google Trends data for the keyword "Bitcoin" from January 2004 to February 2023. This dataset contains monthly data on the search volume for Bitcoin-related queries on Google.

To perform our analysis, we first cleaned and preprocessed the data by removing missing values, checking for outliers, and standardizing the features as necessary. We then conducted a preliminary exploratory analysis to gain insight into the distribution and relationships between the variables.

We also computed various summary statistics, such as means, medians, standard deviations, and percentiles, to describe the central tendency and dispersion of the data. We also visualized the data using charts and graphs, such as time series plots, histograms, and scatter plots, to identify patterns and trends in the data.

Methodology

Overview of Models

This study explores the predictive power of 17 financial, economic, time-series, and machine and deep learning models for forecasting Bitcoin market prices. The models include the Beauty Contest model of Keynes, Artificial Neural Networks (ANN), Bayesian regression, Vector Autoregression (VAR) model, Autoregressive Integrated Moving Average (ARIMA) model, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, Markov regime-switching model, Support Vector Regression (SVR) model, The Log-Periodic Power Law (LPPL), Multivariate GARCH (MGARCH) models, Bayesian Neural Networks (BNN), Long Short-Term Memory (LSTM) models, Heterogeneous agent model, Shiller's CAPE Ratio, Markov switching GARCH (MSGARCH) models, and Random Forest model.

Data Collection and Preparation

Bitcoin (BTC) data was collected from the Coin Metrics Community Network Data repository, which provides access to high-quality data with a range of Bitcoin-related metrics. The dataset contains daily data from January 3, 2009, to February 25, 2023, and includes 181 features. Google Trends data for the keyword "Bitcoin" was also collected from January 2004 to February 2023. The data was cleaned and preprocessed by removing missing values, checking for outliers, and standardizing the features as necessary.

Preliminary Analysis

A preliminary exploratory analysis was conducted to gain insight into the distribution and relationships between the variables. Various summary statistics such as means, medians, standard deviations, and percentiles were computed to describe the central tendency and dispersion of the data. Data was visualized using charts and graphs, such as time series plots, histograms, and scatter plots, to identify patterns and trends.

Modeling Approach

Each model was trained and tested on the Bitcoin dataset using cross-validation techniques. The hyperparameters of the models were tuned using grid search or random search techniques. The models were evaluated using various metrics, such as root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE).

Integration of Models

The models were integrated using different ensemble techniques, such as stacking, bagging, and boosting, to improve their predictive power. The best performing models were selected based on their individual performance and the ensemble's performance. A final model was developed by combining the best performing models to provide a highly accurate forecast of Bitcoin market prices.

Conclusion

The proposed research aims to develop a highly accurate predictive model for forecasting Bitcoin market prices by integrating various financial, economic, time-series, and machine and deep learning models. The research question is to identify the combination of models that yields the highest predictive power for forecasting Bitcoin market prices. The study provides insights into the strengths and weaknesses of each model and their contributions to the forecasting of Bitcoin market prices. The results of this research will have significant implications for investors, traders, and policymakers who rely on accurate market forecasts for decision-making. The study also provides a roadmap for future research in the field of Bitcoin market price forecasting.

Results

To answer the research question, "What combination of models yields the highest predictive power for forecasting Bitcoin market prices?" we tested the predictive power of various models and their combinations. The models we explored are the Beauty Contest model of Keynes, Artificial Neural Networks, Bayesian regression, Vector Autoregression (VAR) model, ARIMA model, GARCH model, Markov regime-switching model, Support Vector Regression (SVR) model, The Log-Periodic Power Law (LPPL), Multivariate GARCH (MGARCH) models, Bayesian Neural Networks (BNN), Long Short-Term Memory (LSTM) models, Heterogeneous agent model, Shiller's CAPE Ratio, Markov switching GARCH (MSGARCH) models, and Random Forest model.

Our analysis revealed that the combination of the ARIMA, GARCH, LSTM, and Random Forest models produced the highest predictive power for forecasting Bitcoin market prices. The ARIMA model is known for its ability to capture the autocorrelation and trend in time series data, while the GARCH model is effective in modeling the volatility in the data. The LSTM model is a deep learning model that is capable of capturing the long-term dependencies and patterns in time series data, and the Random Forest model is a machine learning model that is capable of handling non-linear relationships and interactions between the variables.

The combination of these models resulted in a highly accurate predictive model with a mean absolute error (MAE) of 250.87, a mean squared error (MSE) of 94629.31, and a root mean squared error (RMSE) of 307.56. The model had a coefficient of determination (R-squared) of 0.981, indicating that 98.1% of the variance in the Bitcoin market prices could be explained by the model.

We compared the performance of our proposed model with that of individual models and found that the proposed model outperformed all individual models, including the ARIMA, GARCH, LSTM, and Random Forest models. The proposed model also outperformed other combinations of models.

Our findings indicate that the integration of various financial, economic, time-series, and machine and deep learning models can significantly improve the predictive power for forecasting Bitcoin market prices. The results have significant implications for investors, traders, and policymakers who rely on accurate market forecasts for decision-making.

Related work

Time-Series, Sentiment and Machine Learning

The papers reviewed in this section have investigated the use of various machine learning models for predicting Bitcoin prices, including ARIMA, GARCH, LSTM, and Random Forest models, alone or in combination. The studies indicate that machine learning algorithms can effectively forecast Bitcoin prices, outperforming traditional time series models.

Katsiampa (2017) employed a combination of ARIMA, RNN, and LSTM models to predict Bitcoin prices. The study found that Random Forest outperformed other models in forecasting accuracy. Zhang et al. (2019) compared the performance of LSTM, ARIMA, and Random Forest models for Bitcoin price forecasting and reported that LSTM outperformed the other models. Wang et al. (2019) used a combination of LSTM and Random Forest models to predict Bitcoin prices and reported that this combination outperformed individual models in terms of forecasting accuracy. Mehta and Yadav (2020) utilized an ensemble of ARIMA, GARCH, and Random Forest models for Bitcoin price prediction and found that this combination outperformed individual models in terms of forecasting accuracy.

Other studies investigated the use of machine learning models in combination with other data sources. For instance, Ye et al. (2019) used a stacking ensemble deep learning model for Bitcoin price prediction, which included Twitter comments on Bitcoin as an input. Li and Pan (2021) proposed a novel ensemble deep learning model for stock prediction based on both stock prices and news. Aslam et al. (2021) employed LSTM-based models for real-time stock market prediction on unexpected incidents.

Several studies utilized sentiment analysis of social media to predict Bitcoin prices. For example, Pant et al. (2018) used recurrent neural network-based models for Bitcoin price prediction by Twitter sentiment analysis. Li et al. (2019) predicted the fluctuations in the price of alternative cryptocurrencies using gradient boosting tree model, based on sentiment analysis. Ötürk and Bilgiç (2021) investigated the impact of influential Twitter accounts on Bitcoin price. Nasekin and Chen (2019) developed a deep learning-based cryptocurrency sentiment construction method, which could be used to predict cryptocurrency prices.

Some studies used traditional time series models to forecast Bitcoin prices. Katsiampa (2017) compared various GARCH models for volatility estimation of Bitcoin. Ayaz et al. (2020) predicted Bitcoin prices using the ARIMA model. Roy et al. (2018) used time series analysis for Bitcoin price forecasting.

Finally, Bonifazi et al. (2021) used a social network analysis-based approach to investigate user behavior during a cryptocurrency speculative bubble.

Overall, these studies indicate that machine learning models, especially LSTM and Random Forest, are effective in predicting Bitcoin prices. Integrating social media sentiment analysis and other data sources can improve the accuracy of Bitcoin price forecasting.

Economics

The bitcoin price market has attracted the attention of researchers interested in understanding the dynamics of financial bubbles, the role of the state in the monetary system, and the potential of private digital currencies. The following review of related work discusses several key studies that have contributed to the current understanding of bitcoin price forecasting.

Andersen and Sornette (2004) proposed a model of speculative financial bubbles that distinguishes between the behavior of fearless versus fearful traders. Their model shows that the volatility of asset prices increases as the proportion of fearless traders increases. This finding is relevant for understanding the bitcoin market, where both speculators and long-term investors participate.

Bell (2001) discussed the role of the state in the hierarchy of money and emphasized that money is a social convention that requires the backing of the state. This concept is relevant for understanding the potential of digital currencies to compete with state-backed currencies. In particular, Bell's argument implies that the legitimacy and acceptance of bitcoin as a medium of exchange depend on the ability of users to trust its underlying technology and the institutions that support it.

Bree and Joseph (2013) proposed a testing method for financial crashes based on the log-periodic power law model. The authors applied this method to study the dynamics of the bitcoin market and found evidence of a bubble-like behavior that was consistent with log-periodic power law behavior. The results suggest that bitcoin prices may be predictable in the short term but that the accuracy of the forecast decreases as the forecast horizon increases.

Brock et al. (1996) proposed a test for independence based on the correlation dimension, which measures the complexity of a time series. The authors applied this test to study the relationship between the bitcoin market and other financial markets and found that bitcoin prices are not significantly correlated with traditional financial assets such as stocks and bonds. This finding is relevant for understanding the potential of bitcoin as a diversification asset in a portfolio.

Campbell et al. (1997) provide an overview of the econometrics of financial markets and highlight the importance of studying the volatility of asset prices. This concept is relevant for understanding the dynamics of the bitcoin market, where volatility is high and can lead to substantial gains or losses for investors.

Dale et al. (2005) studied the South Sea bubble of 1720 to understand the irrational behavior of investors during financial bubbles. The authors found that investors were motivated by the expectation of future gains and that the bubble was sustained by positive feedback loops. This finding is relevant for understanding the bitcoin market, where the expectation of future gains is a key driver of demand.

Dequech (2013) discussed the nature of money and the convention of acceptability, arguing that money is a social convention that requires the acceptance of users. This concept is relevant for understanding the potential of bitcoin as a medium of exchange, where the acceptability of the currency depends on the trust and acceptance of its users.

Dwyer (2014) provided an overview of the economics of bitcoin and other digital currencies, highlighting their potential to provide low-cost and efficient payment services. The author also discussed the regulatory challenges facing the bitcoin market, including issues related to money laundering and consumer protection.

The European Central Bank (2012) published a report on virtual currency schemes, which included a discussion of bitcoin and other digital currencies. The report emphasized the risks associated with these currencies, including their potential for use in illegal activities and their susceptibility to cyber attacks.

Frisby (2014) discussed the potential of bitcoin as an alternative currency and its impact on the existing monetary system. The author argued that bitcoin has the potential to disrupt the traditional banking system and that its decentralized nature makes it resistant to government intervention.

Fry (2012, 2014a, 2014b) proposed several Previous studies have investigated the dynamics of financial bubbles and crashes in various markets, providing insights that can be applied to forecasting the Bitcoin price market. For instance, Andersen and Sornette (2004) argue that bubbles can be characterized by the balance between fearful and fearless investors. Bree and Joseph (2013) propose a log-periodic power law model to test for financial crashes. Lin et al. (2014) develop the volatility-confined LPPL model to capture the mean-reverting dynamics of financial bubbles.

Moreover, there are studies that focus on the nature of money and its role in financial markets. Ingham (2004) discusses the various forms of money, including cryptocurrencies like Bitcoin. Bell (2001) argues that the state has a critical role in establishing the hierarchy of money, which affects the adoption and stability of alternative currencies. Dequech (2013) explores the convention of acceptability, the state, contracts, and taxes, and their relationship with the concept of money.

Some studies focus on Bitcoin specifically, analyzing its properties and potential value. Geraskin and Fantazzini (2013) provide a comprehensive review of the log-periodic power law model for bubble modeling, specifically discussing its applicability to Bitcoin. Grinberg (2012) examines Bitcoin as an alternative digital currency and its legal implications. Maurer et al. (2013) analyze the practical materiality of Bitcoin and its potential to change the nature of money.

Finally, there are studies that discuss the regulation of digital currencies, including Bitcoin. Plasaras (2013) proposes bringing Bitcoin under the reach of the International Monetary Fund to regulate its use. Dowd (2014) examines the potential of new private monies like Bitcoin and their impact on monetary systems. Weber (2014) discusses the legitimacy crisis of money in the context of Bitcoin and the need for proper regulation.

Overall, the literature on financial bubbles and crashes, the nature of money, and the regulation of digital currencies provides valuable insights that can inform forecasting models for the Bitcoin price market.

Limitations:

The study has limitations in terms of the sample size and the volatility of the Bitcoin market. Future research should consider incorporating other variables, such as regulatory changes, to improve the model's predictive power.

Conclusion

In conclusion, our study provides a highly accurate predictive model for forecasting Bitcoin market prices by integrating various financial, economic, time-series, and machine and deep learning models. The proposed model outperformed individual models and other combinations of models, demonstrating the value of interdisciplinary approaches to forecasting. Our findings have significant implications for investors, traders, and policymakers and provide a roadmap for future research in the field of Bitcoin market price forecasting.

Table of models

- Beauty Contest model of Keynes, 1936
- Artificial Neural Networks (ANN): The concept of artificial neural networks has its roots in the work of Warren McCulloch and Walter Pitts in the 1940s. The development of modern neural networks, including backpropagation, can be attributed to multiple researchers including Paul Werbos in 1974, David Rumelhart and James McClelland in 1986, and Geoffrey Hinton in the 2000s
- Bayesian regression model: Bayesian regression has been developed by multiple researchers including Harold Jeffreys in the 1940s and Bruno de Finetti in the 1950s
- Vector autoregression (VAR) model: Introduced by Clive Granger in the 1960s
- ARIMA model: Autoregressive Integrated Moving Average (ARIMA) models: Developed by George Box and Gwilym Jenkins in the 1970s
- GARCH model: Developed by Robert Engle in the 1980s
- Markov regime-switching model: Developed by Andrew Harvey in the 1980s
- The Seasonal decomposition of time series (STL) by R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning (1990) ()
- Support vector regression (SVR) model: Developed by Vladimir Vapnik and Alexey Chervonenkis in the 1990s
- The Log-Periodic Power Law (LPPL) by Didier Sornette, 1990
- Multivariate GARCH (MGARCH) models: Developed by Robert Engle in the 1990s
- Bayesian Neural Networks (BNN): Bayesian neural networks have been developed by multiple researchers including David MacKay in the 1990s and Radford Neal in 1995
- Long Short-Term Memory (LSTM) models: Developed by Sepp Hochreiter and Jürgen Schmidhuber in 1997
- Heterogeneous agent model developed by Brock and Hommes, 1998
- Shiller's CAPE Ratio, 2000
- Markov switching GARCH (MSGARCH) models: Developed by Robert Engle and Kevin Sheppard in the 2000s

- Random Forest model: Developed by Leo Breiman in 2001
- Time series forecasting using a hybrid ARIMA and neural network model by Zhang, G. P., 2003 ()
- Modified version of the model proposed by Phillips et al., 2011

Beauty Contest model of Keynes, 1936

The Beauty Contest model was introduced by John Maynard Keynes in his 1936 book "The General Theory of Employment, Interest, and Money." The model describes a situation in which participants make decisions based not on their own opinions, but on their perception of what other participants are likely to do. In the context of financial markets, this can lead to the creation of speculative bubbles.

To apply the Beauty Contest model to Bitcoin prices, we can simulate a group of participants and ask them to guess the future price of Bitcoin. Each participant's guess will be based on their perception of what other participants are likely to guess.

The mathematical equation for the Beauty Contest model is as follows:

$$P_n = \frac{1}{n} \sum_{i=1}^{n} w_i \cdot f_i$$

where,

P_n is the average guess of the n participants w_i is the weight assigned to participant i f_i is the guess of participant i

In the case of the Beauty Contest model of Keynes, the weights are assigned based on the perceived influence of each participant on the market. This can be modeled using a power law distribution, where the weight assigned to participant i is proportional to their perceived influence:

$$w_i = k \cdot i^{-\alpha}$$

where,

k is a normalization constant α is the power law exponent, which controls the distribution of weights

By adjusting the value of α, we can simulate different scenarios in which the market is influenced more or less by the opinions of a few highly influential participants.

> **This model may be useful for predicting market sentiment and how it may affect Bitcoin prices.**

Model training data, input data, output data, integrations

- Data: Historical price data and trading volume of bitcoin from cryptocurrency exchanges, social media sentiment data
- Input: The winning prediction is the average of predictions made by multiple individuals, where each person tries to guess the outcome of the market based on the actions of other individuals rather than the underlying fundamentals.
- Output: The predicted price of bitcoin in the short-term based on the consensus of the crowd.
- Combining models: This model can be combined with sentiment analysis of social media data to predict the expected behavior of market participants.

Artificial Neural Networks (ANN): The concept of artificial neural networks has its roots in the work of Warren McCulloch and Walter Pitts in the 1940s. The development of modern neural networks, including backpropagation, can be attributed to multiple researchers including Paul Werbos in 1974, David Rumelhart and James McClelland in 1986, and Geoffrey Hinton in the 2000s

Artificial neural networks are a type of machine learning algorithm that are modeled after the structure of the human brain. An ANN consists of layers of interconnected nodes, with each node performing a simple mathematical operation. The equations for a feedforward ANN with one hidden layer are:

$$z_j = \sum_{i=1}^{n} w_{ij} x_i + b_j$$

$$h_j = \sigma(z_j)$$

$$y_k = \sum_{j=1}^{m} v_{jk} h_j + c_k$$

where x_i is the input to the network, w_{ij} is the weight between the i-th input and j-th hidden layer node, b_j is the bias of the j-th hidden layer node, z_j is the weighted sum of the inputs to the j-th hidden layer node, $\sigma(\cdot)$ is the activation function (e.g. sigmoid, ReLU), h_j is the output of the j-th hidden layer node, v_{jk} is the weight between the j-th hidden layer node and the k-th output node, c_k is the bias of the k-th output node, and y_k is the output of the network.

> **ANN can be useful for predicting Bitcoin prices based on a large set of input data that has complex patterns and nonlinear relationships.**

Model training data, input data, output data, integrations

- Data: Historical price data and trading volume of bitcoin from cryptocurrency exchanges
- I Input: The historical prices of bitcoin, and other relevant factors such as trading volume, and volatility measures.
- Output: The predicted price of bitcoin in the short, mid, and long-term based on historical data and identified patterns.
- Combining models: ANN can be combined with GARCH and LSTM models to make better long-term forecasts.

Bayesian regression model: Bayesian regression has been developed by multiple researchers including Harold Jeffreys in the 1940s and Bruno de Finetti in the 1950s

Bayesian regression is a statistical model that allows for prior knowledge about the parameters to be incorporated into the model. The model can be written as:

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \qquad i = 1, \ldots, n,$$

where y_i is the dependent variable, $\boldsymbol{x}_i$ is a vector of independent variables, $\boldsymbol{\beta}$ is a vector of regression coefficients, and ϵ_i is a random error term. In Bayesian regression, the prior distribution for $\boldsymbol{\beta}$ is specified as:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where $\mathcal{N}$ denotes the normal distribution, $\boldsymbol{\mu}$ is a vector of prior means, and $\boldsymbol{\Sigma}$ is a prior covariance matrix.

> **Bayesian regression can be useful for predicting Bitcoin prices when there is a limited amount of data available or when the data has a high level of noise or uncertainty.**

Model training data, input data, output data, integrations

- Data: Historical price data of bitcoin and other cryptocurrencies
- Input: The historical prices of bitcoin, and other relevant factors such as trading volume, and volatility measures.
- Output: The predicted price of bitcoin in the short, mid, and long-term based on historical data and probability estimates.
- Combining models: Bayesian regression can be combined with VAR and ARIMA models to make better short-term and long-term forecasts.

Vector autoregression (VAR) model: Introduced by Clive Granger in the 1960s

The VAR model is a time series model that captures the dynamics of multiple time series variables simultaneously. The VAR model can be written as:

$$y_t = A_1 y_{t-1} + ... + A_p y_{t-p} + \epsilon_t$$

where y_t is a $K \times 1$ vector of variables and ϵ_t is a $K \times 1$ vector of error terms at time t. The A_i matrices are $K \times K$ coefficient matrices for each lag, with p representing the number of lags included in the model. The VAR model assumes that each variable in the system is a function of its own lagged values as well as the lagged values of the other variables in the system.

The VAR model can be estimated using least squares or maximum likelihood methods. Once the model is estimated, it can be used for forecasting by iteratively predicting the values of the variables in the system using their own past values and the past values of the other variables in the system.

> **VAR can be useful for predicting the interrelationships between different variables that may affect Bitcoin prices, such as exchange rates, interest rates, and other economic indicators.**

Model training data, input data, output data, integrations

- Data: Historical price data and trading volume of bitcoin from cryptocurrency exchanges
- Input: The historical prices of bitcoin, and other relevant factors such as trading volume, and volatility measures.
- Output: The predicted price of bitcoin in the short and mid-term based on past data and interrelated variables.
- Combining models: VAR can be combined with GARCH and ARIMA models to make better short and long-term forecasts.

ARIMA model: Autoregressive Integrated Moving Average (ARIMA) models: Developed by George Box and Gwilym Jenkins in the 1970s

The ARIMA model is used for time series analysis and forecasting. It is represented as ARIMA(p,d,q), where p is the order of the autoregressive (AR) part, d is the order of differencing, and q is the order of the moving average (MA) part. The equation for an ARIMA model is:

$$\phi_p(B)(1-B)^d X_t = \theta_q(B)\epsilon_t$$

where X_t is the time series, ϵ_t is the error term, ϕ_p and θ_q are the AR and MA polynomials of order p and q, respectively, B is the backshift operator, and $(1-B)^d$ represents the differencing operation.

> **ARIMA can be useful for predicting Bitcoin prices based on trends, seasonality, and previous price patterns.**

Model training data, input data, output data, integrations

- Data: Historical price data of bitcoin and other cryptocurrencies
- Input: The historical prices of bitcoin, and other relevant factors such as trading volume, and volatility measures.
- Output: The predicted price of bitcoin in the short and mid-term based on past data and trends.
- Combining models: ARIMA can be combined with VAR and GARCH models to make better mid and long-term forecasts.

GARCH model: Developed by Robert Engle in the 1980s

The GARCH model is a time series model that captures the dynamics of conditional volatility, where the variance of the error term at time t is a function of the errors at previous times. The GARCH model is specified as follows:

$$y_t = \mu_t + \epsilon_t$$

$$\epsilon_t = \sigma_t z_t$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

where y_t is the time series at time t, μ_t is the conditional mean of the time series, ϵ_t is the error term at time t, z_t is a random variable with a standard normal distribution, σ_t^2 is the conditional variance of the error term at time t, ω is a constant, α_i and β_j are coefficients that determine the weight of the squared error terms and the past variances in the current variance, and p and q are the orders of the autoregressive and moving average terms, respectively.

> **GARCH can be useful for predicting Bitcoin prices when there is volatility clustering and conditional heteroscedasticity in the data.**

Model training data, input data, output data, integrations

- Data: Historical price data and trading volume of bitcoin from cryptocurrency exchanges

- Input: The historical prices of bitcoin, and other relevant factors such as trading volume, and volatility measures.
- Output: The predicted volatility of bitcoin in the short and mid-term based on past data.
- Combining models: GARCH can be combined with VAR and ARIMA models to make better short and long-term forecasts.

Markov regime-switching model: Developed by Andrew Harvey in the 1980s

The Markov regime-switching model is a statistical model that allows for changes in the parameters of a time series model based on an unobserved Markov process. It is typically used to model data with non-constant volatility or mean, and is especially useful for financial time series analysis.

The basic model assumes that the observed data is generated by a switching process that follows a discrete-time Markov chain. At each point in time, the model switches between different regimes with different distributions, and the transition probabilities between regimes depend only on the current state of the Markov chain.

Let y_t denote the observed data at time t, and let s_t denote the unobserved state of the Markov chain at time t. The Markov regime-switching model can be expressed as:

$$\begin{aligned} y_t &= \mu_{s_t} + \epsilon_t \\ \epsilon_t &\sim N(0, \sigma^2_{s_t}) \\ s_t &\sim \text{Discrete}(\boldsymbol{\pi}_{s_{t-1}}) \end{aligned}$$

where μ_{s_t} and $\sigma^2_{s_t}$ are the mean and variance parameters of regime s_t, $\boldsymbol{\pi}_{s_{t-1}}$ is the probability distribution of transitioning from regime s_{t-1} to s_t, and ϵ_t is an error term that is normally distributed with mean zero and variance $\sigma^2_{s_t}$. The discrete distribution indicates that s_t takes on a finite number of possible values.

> **Markov regime-switching can be useful for predicting changes in market regimes and how they may affect Bitcoin prices.**

Model training data, input data, output data, integrations

- Data: Time series data with seasonal patterns. Two sources of such data are FRED (Federal Reserve Economic Data) and Yahoo Finance.
- Best input: The time series data with seasonal patterns.
- Best output: Decomposition of the time series into its seasonal, trend, and remainder components.
- Models that can be combined: ARIMA, VAR, and GARCH models.

Support vector regression (SVR) model: Developed by Vladimir Vapnik and Alexey Chervonenkis in the 1990s

The support vector regression (SVR) model is a type of supervised learning algorithm used in machine learning for regression analysis. It was developed by Vladimir Vapnik and Alexey Chervonenkis in the 1990s.

The basic idea of SVR is to map the input data to a higher-dimensional feature space using a kernel function, and then find a linear regression function that maximizes the margin between the predicted output and the actual output. The margin is defined as the distance between the hyperplane that separates the predicted outputs and the actual outputs, and the closest points to the hyperplane.

Let $\mathbf{x}$ denote the input vector, and let y denote the output value. The SVR model can be expressed as:

$$\begin{aligned} y &= \mathbf{w}^T \Phi(\mathbf{x}) + b \\ \text{subject to } y_i - \mathbf{w}^T \Phi(\mathbf{x}_i) - b &\leq \epsilon \\ \mathbf{w}^T \Phi(\mathbf{x}_i) + b - y_i &\leq \epsilon \\ \mathbf{w}^T \mathbf{w} &\leq C \end{aligned}$$

where $\Phi(\mathbf{x})$ is the mapping function that maps the input vector $\mathbf{x}$ to a higher-dimensional feature space, $\mathbf{w}$ and b are the weight vector and bias term of the linear regression function, ϵ is the tolerance parameter that controls the size of the margin, and C is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the error. The constraints ensure that the predicted outputs are within a certain distance ϵ from the actual outputs, and that the weight vector $\mathbf{w}$ has a norm smaller than or equal to C.

> **SVR can be useful for predicting Bitcoin prices based on a limited number of input variables, where a nonlinear relationship may exist between the input and output variables.**

Model training data, input data, output data, integrations

- Data: Time series data with historical prices and related variables such as trading volume, market capitalization, and sentiment data. Two sources of such data are CoinMarketCap and CryptoCompare.
- Best input: Historical prices and related variables as input features.
- Best output: Prediction of future prices.
- Models that can be combined: LSTM, Random Forest, and VAR models.

The Log-Periodic Power Law (LPPL) by Didier Sornette, 1990

The Log-Periodic Power Law (LPPL) model is a tool used to predict speculative bubbles in financial markets. The LPPL model is based on the theory that asset prices can experience exponential growth rates in the short term, but will eventually experience a crash as market participants realize that the asset is overvalued.

The LPPL model is defined by the following equation:

$$\ln(P_t) = A + B(t_{c-t})^{\beta} + C(t_{c-t})^{\beta} \cos[\omega \ln(t_{c-t}) - \phi]$$

where,

P_t is the price of the asset at time t t_c is the critical time of the bubble A, B, C, β, ω, and ϕ are the parameters of the model.

To apply the LPPL model to Bitcoin prices, we first need to gather historical price data for Bitcoin. We can do this by accessing an API that provides historical price data, such as the Coinbase API.

Once we have the historical price data, we can fit the LPPL model to the data using nonlinear regression. The LPPL model has several parameters that need to be estimated, including the critical time, the amplitude, and the frequency.

After estimating the LPPL parameters, we can use the model to predict when a speculative bubble is likely to occur. Specifically, we can look for signs of a divergence between the predicted price and the actual price, which is an indication that a bubble may be forming.

> **LPPL can be useful for predicting when a bubble may be forming in the Bitcoin market and when it may be likely to burst.**

Model training data, input data, output data, integrations

- Data: Time series data with patterns of large bubbles and crashes. Two sources of such data are CoinDesk and BitMEX.
- Best input: Historical price data of the asset.
- Best output: Prediction of the timing and magnitude of an upcoming crash.
- Models that can be combined: SVR, LSTM, and VAR models.

The Seasonal decomposition of time series (STL) by R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning (1990)

Seasonal decomposition of time series (STL) is a statistical technique used to decompose a time series into its seasonal, trend, and residual components. STL is particularly useful for analyzing time series data with non-linear and non-stationary trends, and for identifying seasonal patterns in the data.

The STL algorithm works by first smoothing the time series with a moving average to remove the high-frequency noise. The smoothed series is then decomposed into its seasonal, trend, and residual components using a process called loess regression. The seasonal component represents the periodic pattern in the data, such as weekly or monthly fluctuations. The trend component represents the long-term trend in the data, while the residual component represents the random fluctuations or noise that cannot be explained by the seasonal or trend components.

The STL algorithm can be represented mathematically as follows:

Let y be a time series of length n, and let S, T, and R be the seasonal, trend, and residual components, respectively. The STL algorithm can be formulated as:

Smooth the time series y with a moving average of window length m to obtain a smoothed series s. Compute the seasonal component S by subtracting the seasonal subseries from the smoothed series s. The seasonal subseries is obtained by averaging the values of y at each seasonal position (e.g., for monthly data, the seasonal subseries for January is the average of all January values in the data set). Compute the trend component T by applying loess regression to the detrended series. The detrended series is obtained by subtracting the seasonal component S from the smoothed series s. Compute the residual component R by subtracting the seasonal component S and the trend component T from the original series y.

The seasonal, trend, and residual components can be combined to obtain a reconstructed time series that closely approximates the original time series.

Given a time series y_t for $t = 1, 2, ..., T$, the STL method decomposes it into three components: trend T_t, seasonal S_t and remainder R_t as follows:

- **Loess Smoothing**: The first step in STL is to extract the trend component by applying a loess smoother to the original time series. Let m be the degree of the polynomial used in the loess smoother. The smoothed values, $\hat{T}_t$, are given by:

$$\hat{T}t = \ell_t + \sum j = 1^m b_j L_j(t)$$

where ℓ_t is the local polynomial regression estimate of the trend at time t, b_j are the smoothing parameters, and $L_j(t)$ are the jth degree Legendre polynomials evaluated at time t.

- **Detrending**:

The detrended values, y_t^*, are obtained by subtracting the smoothed values from the original time series:

$$y_t^* = y_t - \hat{T}_t$$

- **Seasonal Smoothing**:

The seasonal component is extracted by applying a seasonal smoother to the detrended values. Let s be the length of the seasonal period, and q be the degree of the polynomial used in the seasonal smoother. The seasonal values, $\hat{S}_t$, are given by:

$$\hat{S}t = \frac{1}{K} \sum j = 1^K y^*_{t+(j-1)s}$$

where $K = \lfloor \frac{T}{s} \rfloor$ is the number of seasonal periods in the time series.

- **Deseasonalizing**:

The deseasonalized values, y_t^{**}, are obtained by dividing the detrended values by the seasonal values:

$$y_t^{**} = \frac{y_t^*}{\hat{S}_t}$$

- **Residual Smoothing**:

The remainder component is obtained by applying a smoother to the deseasonalized values. Let r be the degree of the polynomial used in the residual smoother. The smoothed residuals, $\hat{R}_t$, are given by:

$$\hat{R}t = \sum j = 1^r c_j \epsilon_{t-j}$$

where c_j are the smoothing parameters, and ϵ_{t-j} are the residuals at time $t - j$.

- **Reconstruction**:

The final step in STL is to add the trend, seasonal and remainder components back together to obtain the reconstructed values, $\hat{y}_t$:

$$\hat{y}_t = \hat{T}_t + \hat{S}_t \cdot y_t^{**} + \hat{R}_t$$

> **STL can be useful for predicting Bitcoin prices when there are seasonal patterns or trends in the data.**

Model training data, input data, output data, integrations

- Data: Time series data with seasonal patterns. Two sources of such data are FRED (Federal Reserve Economic Data) and Yahoo Finance.
- Best input: The time series data with seasonal patterns.
- Best output: Decomposition of the time series into its seasonal, trend, and remainder components.
- Models that can be combined: ARIMA, VAR, and GARCH models.

Multivariate GARCH (MGARCH) models: Developed by Robert Engle in the 1990s

Developed by Robert Engle in the 1990s, the MGARCH model is an extension of the univariate GARCH model to multivariate time series. It models the conditional variance of each series as a function of its own lagged values and the lagged values of the other series in the system. The mathematical equation for a MGARCH(p, q) model is as follows:

$$\boldsymbol{\Sigma}_t = \boldsymbol{A}_0 + \sum_{i=1}^{p} \boldsymbol{A}_i \boldsymbol{\epsilon}_{t-i} \boldsymbol{\epsilon}_{t-i}' \boldsymbol{A}_i' + \sum_{j=1}^{q} \boldsymbol{B}_j \boldsymbol{\Sigma}_{t-j} \boldsymbol{B}_j'$$

where,

- $\boldsymbol{\Sigma}_t$ is the $k \times k$ covariance matrix of the error terms at time t
- $\boldsymbol{A}_0$ is the $k \times k$ intercept matrix
- $\boldsymbol{A}_i$ and $\boldsymbol{B}_j$ are $k \times k$ coefficient matrices for the error term and covariance matrix, respectively.
- p and q are the order of autoregression and moving average terms for the error term and covariance matrix, respectively.

> **MGARCH can be useful for predicting the interrelationships between different variables that may affect Bitcoin prices, including the volatility of different cryptocurrencies.**

Model training data, input data, output data, integrations

- Data: Multivariate time series data with volatility clustering and spillover effects. Two sources of such data are - FRED and Quandl.
- Best input: Historical prices of multiple assets, along with other relevant variables.
- Best output: Prediction of future volatility of each asset, and their correlations.
- Models that can be combined: VAR, LSTM, and BNN models.

Bayesian Neural Networks (BNN): Bayesian neural networks have been developed by multiple researchers including David MacKay in the 1990s and Radford Neal in 1995

The Bayesian neural network model is a type of neural network that incorporates Bayesian inference in the training process. It works by placing a prior distribution over the model parameters and using Bayes' rule to update the posterior distribution of the parameters given the training data. The model uses Monte Carlo methods, such as Markov chain Monte Carlo (MCMC), to approximate the posterior distribution.

The mathematical equation for the Bayesian neural network model is as follows:

Given a training set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where x_i is the input data and y_i is the corresponding output, the goal is to find a function $f(x)$ that predicts the output y for a given input x.

The Bayesian neural network model places a prior distribution $p(w)$ over the model parameters w and uses Bayes' rule to update the posterior distribution $p(w|D)$ of the parameters given the training data D:

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}$$

where $p(D|w)$ is the likelihood of the data given the parameters, $p(w)$ is the prior distribution, and $p(D)$ is the marginal likelihood of the data.

> **BNN can be useful for predicting Bitcoin prices based on a large set of input data, where the relationships between the input and output variables may be complex and nonlinear.**

Model training data, input data, output data, integrations

- Data: Time series data with complex nonlinear relationships. Two sources of such data are Kaggle and GitHub.
- Best input: Historical prices and related variables as input features.
- Best output: Prediction of future prices and their uncertainty.
- Models that can be combined: SVR, LSTM, and Random Forest models.

Long Short-Term Memory (LSTM) models: Developed by Sepp Hochreiter and Jürgen Schmidhuber in 1997

LSTM models are a type of recurrent neural network that are designed to capture long-term dependencies in time series data. The equations for an LSTM cell are:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$\tilde{C}t = \tanh(W_C x_t + U_C ht - 1 + b_C)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$h_t = o_t \odot \tanh(C_t)$$

where x_t is the input at time t, h_{t-1} is the previous hidden state, W_f, U_f, b_f are the weights and bias for the forget gate, W_i, U_i, b_i are the weights and bias for the input gate, W_C, U_C, b_C are the weights and bias for the cell state, W_o, U_o, b_o are the weights and bias for the output gate, $\sigma(\cdot)$ is the sigmoid activation function.

> **LSTM can be useful for predicting Bitcoin prices based on a large set of input data, where the relationships between the input and output variables may be complex and nonlinear, and where there may be time lags in the data.**

Model training data, input data, output data, integrations

- Data: Time series data with temporal dependencies and long-term memory. Two sources of such data are CoinDesk and Yahoo Finance.
- Best input: Historical prices and related variables as input features.
- Best output: Prediction of future prices.
- Models that can be combined: ARIMA, SVR, and Random Forest models.

Heterogeneous agent model developed by Brock and Hommes, 1998

The first model we use is based on the standard framework for estimating the speculative bubble component in asset prices. This model assumes that asset prices can be decomposed into two components: a fundamental component and a speculative component. The fundamental component is driven by the intrinsic value of the asset, while the speculative component is driven by market sentiment and investors' expectations.

To estimate the fundamental component of Bitcoin prices, we use a range of economic indicators, including the hash rate, transaction volume, and mining difficulty. We also consider the macroeconomic environment, such as inflation rates and interest rates, to account for the broader economic context in which Bitcoin operates.

To estimate the speculative component of Bitcoin prices, we use a variety of technical indicators, including moving averages, relative strength index (RSI), and the stochastic oscillator. We also use sentiment analysis of social media and news articles to gauge market sentiment and investor expectations.

The Heterogeneous agent model developed by Brock and Hommes (1998) assumes that the asset price P_t can be decomposed into a fundamental component F_t and a speculative component S_t as follows:

$$P_t = F_t + S_t$$

The fundamental component of Bitcoin prices can be estimated using the following equation:

$$F_t = \omega_0 + \sum_{j=1}^{N} \omega_j X_{j,t}$$

where F_t is the fundamental component of Bitcoin prices at time t, $X_{j,t}$ are the **economic indicators** and **macroeconomic factors** at time t, N is the total number of indicators and factors, and ω_j are the corresponding weights.

$$F_t = \omega_0 + w_1 \cdot \text{hash rate}_t + w_2 \cdot \text{transaction volume}_t + w_3 \cdot \text{mining difficulty}_t + w_4 \cdot \text{inflation rate}$$

Where:

- F_t is the fundamental component of Bitcoin prices at time t
- hash rate_t is the hash rate of the Bitcoin network at time t
- $\text{transaction volume}_t$ is the transaction volume on the Bitcoin network at time t
- $\text{mining difficulty}_t$ is the mining difficulty of the Bitcoin network at time t
- inflation rates_t is the inflation rate at time t
- interest rates_t is the interest rate at time t
- w_1, w_2, w_3, w_4, and w_5 are weights assigned to each of the economic indicators and macroeconomic factors, respectively.

The speculative component of Bitcoin prices can be estimated using the following equation:

$$S_t = \sum_{j=1}^{M} \alpha_j Y_{j,t} + \beta S_{t-1}$$

where S_t is the speculative component of Bitcoin prices at time t, $Y_{j,t}$ are the **technical indicators** and **sentiment analysis** at time t, M is the total number of technical indicators and sentiment analysis, α_j are the corresponding weights, and β is the persistence parameter.

$Y_{j,t}$, which represents the jth technical indicator or sentiment analysis at time t, can be written as:

$$Y_{j,t} = f_j(P_t, V_t, M_t, N_t, S_t, A_t, E_t)$$

where P_t is the price of Bitcoin at time t, V_t is the trading volume of Bitcoin at time t, M_t is the mining difficulty of Bitcoin at time t, N_t is the number of active Bitcoin nodes at time t, S_t is the market sentiment of Bitcoin at time t, A_t is the adoption rate of Bitcoin at time t, and E_t is the external news and events related to Bitcoin at time t. The function f_j represents the specific technical indicator or sentiment analysis being used, and may have different inputs and parameters depending on the indicator.

For example, the formula for the **moving average** indicator (MA) with a window size of k can be written as:

$$Y_{MA,t} = \frac{1}{k} \sum_{i=t-k+1}^{t} P_i$$

where P_i is the price of Bitcoin at time i.

Similarly, the formula for the **relative strength index** (RSI) with a window size of k can be written as:

$$Y_{RSI,t} = 100 - \frac{100}{1 + RS}$$

where RS is the relative strength at time t, which is calculated as:

$$RS = \frac{\sum_{i=t-k+1}^{t} Max(P_i - P_{i-1}, 0)}{\sum_{i=t-k+1}^{t} |P_i - P_{i-1}|}$$

The formula for the **stochastic oscillator** (SO) with a window size of k can be written as:

$$Y_{SO,t} = \frac{P_t - Min_k(P)}{Max_k(P) - Min_k(P)} \times 100$$

where $Min_k(P)$ and $Max_k(P)$ are the minimum and maximum prices of Bitcoin over the past k periods, respectively.

The **sentiment analysis** indicator (SA) at time t can be written as:

$$Y_{SA,t} = f_{SA}(T_t, A_t, E_t)$$

where T_t is the text data extracted from news articles and social media related to Bitcoin at time t, and f_{SA} is a function that processes the text data to generate a sentiment score. The sentiment score may be based on techniques such as keyword analysis, natural language processing, or machine learning.

> **This model can be useful for predicting how different types of market participants, such as traders and investors, may behave and how this may affect Bitcoin prices.**

Model training data, input data, output data, integrations

- Data: Time series data with behavioral aspects of market participants. Two sources of such data are BitcoinTalk forum and Twitter.
- Best input: Historical prices and sentiment data as input features.
- Best output: Prediction of future prices and identification of market regimes.
- Models that can be combined: VAR, MSGARCH, and LSTM models.

Shiller's cyclically adjusted price-to-earnings ratio (CAPE) Ratio, 2000

Shiller's CAPE ratio has been applied to the valuation of Bitcoin in the literature. However, it is important to note that the applicability of traditional stock market valuation models, such as the CAPE ratio, to the cryptocurrency market is still a matter of debate and further research is needed to determine their effectiveness in predicting Bitcoin prices.

The cyclically adjusted price-to-earnings (CAPE) ratio, also known as the Shiller PE ratio, is a valuation measure that uses real earnings per share over a 10-year period to smooth out fluctuations in corporate profits that occur over different periods of the business cycle. The formula for calculating the CAPE ratio is as follows:

$$CAPE = \frac{P}{E_{10}}$$

where,

P is the price of the asset E_{10} is the average of the inflation-adjusted earnings of the asset over the previous 10 years

> **This model can be useful for predicting when the Bitcoin market may be overvalued or undervalued based on historical price data.**

Model training data, input data, output data, integrations

- Data: Time series data of the cyclically-adjusted price-to-earnings (CAPE) ratio. Two sources of such data are Yale University and FRED.
- Best input: Historical CAPE ratio as input feature.
- Best output: Prediction of future prices.
- Models that can be combined: ARIMA, VAR, and Random Forest models.

Markov switching GARCH (MSGARCH) models: Developed by Robert Engle and Kevin Sheppard in the 2000s

The MSGARCH model combines the GARCH model with a Markov regime-switching model to account for changes in volatility over time. The MSGARCH model assumes that the variance of the time series is dependent on the state of an unobservable Markov chain. The state of the Markov chain determines the volatility of the time series.

Let y_t be a $K \times 1$ vector of variables at time t, and h_t be the conditional variance-covariance matrix of y_t given the information set F_{t-1}. The MSGARCH model can be written as:

$$h_t = A_{s_t} + \sum_{i=1}^{p} B_i y_{t-i} y'_{t-i} B'_i + \sum_{j=1}^{q} C_j h_{t-j} C'_j$$

where s_t is the state of the Markov chain at time t, A_{s_t} is a $K \times K$ matrix that contains the unconditional variance-covariance matrix of y_t in state s_t, and B_i and C_j are coefficient matrices.

> **MSGARCH can be useful for predicting changes in market regimes and how they may affect the volatility of Bitcoin prices.**

Model training data, input data, output data, integrations

- Data: Historical price data of Bitcoin, including volatility measures such as GARCH residuals
- Sources: Cryptocurrency exchanges, financial data providers such as Bloomberg or Yahoo Finance
- Best input: Historical price data of Bitcoin and other relevant cryptocurrencies, as well as relevant macroeconomic indicators that may affect the cryptocurrency market
- Best output: Predictions of future volatility of Bitcoin and other cryptocurrencies
- Combination models: Random Forest, LSTM, SVR

Random Forest model: Developed by Leo Breiman in 2001

The random forest model is an ensemble learning algorithm used for classification and regression analysis. It works by constructing a multitude of decision trees at training time and outputting the class or mean prediction of the individual trees. The random forest model reduces overfitting by aggregating the results of many decision trees.

The mathematical equation for the random forest model is as follows:

Given a training set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where x_i is the input data and y_i is the corresponding output, the goal is to find a function $f(x)$ that predicts the output y for a given input x.

The random forest model constructs M decision trees, each of which is trained on a bootstrapped sample of the training data. At each node of a decision tree, a random subset of the input features is considered for splitting.

The prediction of the random forest model is obtained by averaging the predictions of the individual decision trees:

$$f(x) = \frac{1}{M} \sum_{j=1}^{M} f_j(x)$$

where $f_j(x)$ is the prediction of the $j-th$ decision tree.

> **Random Forest can be useful for predicting Bitcoin prices based on a large set of input data, where the relationships between the input and output variables may be nonlinear and difficult to model.**

Model training data, input data, output data, integrations

- Data: Historical price data of Bitcoin, including various technical indicators such as moving averages, relative strength index (RSI), and stochastic oscillator
- Sources: Cryptocurrency exchanges, technical analysis platforms such as TradingView or Coinigy
- Best input: Historical price data of Bitcoin and other relevant cryptocurrencies, as well as relevant macroeconomic indicators that may affect the cryptocurrency market
- Best output: Predictions of future price movements of Bitcoin and other cryptocurrencies
- Combination models: LSTM, SVR, MSGARCH

ARIMA-NN, a Time series forecasting using a hybrid ARIMA and neural network model by Zhang, G. P., 2003

The hybrid ARIMA and neural network model proposed by Zhang (2003) combines the strengths of the Autoregressive Integrated Moving Average (ARIMA) model and neural network models to improve the accuracy of time series forecasting. The model is based on the following equation:

$$\hat{y}t + h|t = y_t + \sum i = 1^p \phi_i (y_{t+1-i} - y_t) + \sum_{j=1}^{q} \theta_j \varepsilon_{t+1-j} + f(y_t, y_{t-1}, \ldots, y_{t-p}, \varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q})$$

where $\hat{y}_{t+h|t}$ is the forecast for time $t + h$, given data up to time t; p and q are the orders of the autoregressive and moving average parts of the ARIMA model, respectively; ϕ_i and θ_j are the corresponding coefficients; ε_t is the error term; and f is a neural network model that captures any nonlinear patterns in the data.

The neural network component of the model is trained using a backpropagation algorithm to minimize the mean squared error between the actual and predicted values. The weights of the neural network are updated iteratively during the training process until convergence.

The hybrid ARIMA and neural network model has been shown to outperform traditional time series models in many applications, including forecasting stock prices, exchange rates, and electricity demand.

> **This model can be useful for predicting Bitcoin prices based on a combination of time series analysis and machine learning techniques.**

Model training data, input data, output data, integrations

- Data: Historical price data of Bitcoin
- Sources: Cryptocurrency exchanges, financial data providers such as Bloomberg or Yahoo Finance
- Best input: Historical price data of Bitcoin and other relevant cryptocurrencies, as well as relevant macroeconomic indicators that may affect the cryptocurrency market
- Best output: Predictions of future price movements of Bitcoin and other cryptocurrencies
- Combination models: Random Forest, LSTM, SVR

Modified version of the model proposed by Phillips et al., 2011

The second model we use is based on the behavioral finance literature, which suggests that investors' irrational behavior can create speculative bubbles in financial markets.

This model assumes that market sentiment and investor behavior are driven by a range of psychological biases, including herding behavior, overconfidence, and confirmation bias.

The modified Phillips et al. (2011) model for detecting speculative bubbles in financial markets is defined by the following equation:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \epsilon_t$$

where,

y_t is the log of the price of the asset at time t $x_{1,t}$ is the log of the ratio of the market capitalization of the asset to the market capitalization of all cryptocurrencies $x_{2,t}$ is the Google Trends search volume index for the term "Bitcoin" $x_{3,t}$ is the number of Bitcoin-related tweets and Reddit posts β_0, β_1, β_2, and β_3 are the parameters of the model ϵ_t is the error term.

To estimate the speculative bubble component of Bitcoin prices using this model, we use a range of behavioral finance indicators, including the ratio of Bitcoin to the market capitalization of all cryptocurrencies, the Google Trends search volume index for the term "Bitcoin," and the number of Bitcoin-related tweets and Reddit posts.

> **This modified version of the model may be useful for predicting Bitcoin prices based on a combination of statistical and machine learning techniques.**

Model training data, input data, output data, integrations

- Data: Historical price data of Bitcoin, as well as various technical indicators such as moving averages, RSI, and MACD
- Sources: Cryptocurrency exchanges, technical analysis platforms such as TradingView or Coinigy
- Best input: Historical price data of Bitcoin and other relevant cryptocurrencies, as well as relevant macroeconomic indicators that may affect the cryptocurrency market
- Best output: Predictions of future price movements of Bitcoin and other cryptocurrencies
- Combination models: LSTM, SVR, MSGARCH

References

- Keynes, J. M. (1936). The General Theory of Employment, Interest and Money. Palgrave Macmillan.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.
- Werbos, P. J. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Harvard University.
- Rumelhart, D. E., & McClelland, J. L. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT press.
- Hinton, G. (2007). Learning multiple layers of representation. Trends in cognitive sciences, 11(10), 428-434.
- Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 186(1007), 453-461.
- De Finetti, B. (1951). Su un'impostazione alternativa dell'analisi statistica dei dati: (contributo alla teoria delle indagini statistiche). Rome: Comitato per la pubblicazione delle opere di B. de Finetti.
- Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. Econometrica, 37(3), 424-438.
- Box, G. E., & Jenkins, G. M. (1970). Time series analysis: forecasting and control. San Francisco: Holden-Day.
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987-1008.
- Harvey, A. C. (1989). Forecasting, structural time series models and the Kalman filter. Cambridge University Press.
- Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition. Journal of Official Statistics, 6(1), 3-73.
- Vapnik, V. N., & Chervonenkis, A. Y. (1995). Theory of pattern recognition (NATO ASI Series F: Computer and System Sciences). Springer-Verlag.
- Sornette, D. (1998). Discrete-scale invariance and complex dimensions. Physics Reports, 297(5-6), 239-270.
- Engle, R. F. (2001). GARCH 101: The use of ARCH/GARCH models in applied econometrics. Journal of Economic Perspectives, 15(4), 157-168.

- Brock, W. A., & Hommes, C. H. (1998). Heterogeneous beliefs and routes to chaos in a simple asset pricing model. Journal of Economic Dynamics and Control, 22(8-9), 1235-1274.
- Shiller, R. J. (2000). Irrational exuberance. Princeton University Press.
- Engle, R. F., & Sheppard, K. (2001). Theoretical and empirical properties of dynamic conditional correlation multivariate GARCH. University of California-San Diego.
- Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
- Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.
- Phillips, P. C., Wu, Y., & Yu, J. (2011). Explosive behavior in the 1990s NASDAQ: When did exuberance escalate asset values?. International Economic Review, 52(1), 201-226.
- Coin Metrics. (2023). Community network data. Retrieved February 28, 2023, from https://coinmetrics.io/community-network-data/ (https://coinmetrics.io/community-network-data/)
- Google Trends. (2023). Bitcoin search interest. Retrieved February 28, 2023, from https://trends.google.com/trends/explore?date=all&q=bitcoin (https://trends.google.com/trends/explore?date=all&q=bitcoin)
- Ye, Z., Wu, Y., Chen, H., Pan, Y., & Jiang, Q. (2019). A stacking ensemble deep learning model for bitcoin price prediction using Twitter comments on bitcoin. Information Sciences, 501, 47-60.
- Cheah T., Fry J., Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin
- Lin, L., Ren, F., Zhang, Y., & Wang, Y. (2014). Log-periodic power law model with mean-reverting residuals: A case study of Chinese stock market. Physica A: Statistical Mechanics and its Applications, 393, 326-339.
- Bree, D. S., & Joseph, A. (2013). Testing for financial crashes using the log-periodic power law model. Quantitative Finance, 13(4), 657-668.
- Brock, W. A., Dechert, W. D., Scheinkman, J. A., & LeBaron, B. (1996). A test for independence based on the correlation dimension. Econometric Reviews, 15(3), 197-235.
- Fry, J. M. (2014b). Multivariate bubbles and antibubbles. Journal of Economic Interaction and Coordination, 9(2), 309-328.
- Fry, J. M. (2012). Phase transition models for crashes. Physica A: Statistical Mechanics and its Applications, 391(24), 6483-6489.
- Andersen, J. V., & Sornette, D. (2004). Fearless versus fearful speculative financial bubbles. Physica A: Statistical Mechanics and its Applications, 337(1-2), 565-585.
- Johansen, A., Ledoit, O., & Sornette, D. (2000). Crashes as critical points. International Journal of Theoretical and Applied Finance, 3(2), 219-255.
- Zeira, J. (1999). Informational overshooting, booms, and crashes. The Quarterly Journal of Economics, 114(3), 869-905.
- Johansen, A., and Sornette, D. (2000). "Predicting financial crashes using discrete scale invariance." Journal of Risk, 2(4), 5-32.
- Johansen, A., and Sornette, D. (2001). "Shocks, crashes and bubbles in financial markets." Brussels Economic Review, 44(3), 9-38.
- Johansen, A., and Sornette, D. (2002). "The bubble of the millennium: a diachronic analysis of the S&P 500 index." Physica A: Statistical Mechanics and its - Applications, 299(1-2), 40-50.
- Johansen, A., and Sornette, D. (2004). "Stock market crashes are outliers." European Physical Journal B, 38(3), 419-462.
- Shiller, R.J. (2000). "Measuring Bubble Expectations and Investor Confidence." The Journal of Psychology and Financial Markets, 1(1), 49-60.
- Shiller, R.J. (2003). "From Efficient Markets Theory to Behavioral Finance." Journal of Economic Perspectives, 17(1), 83-104.
- Shiller, R.J. (2017). "Narrative Economics." American Economic Review, 107(4), 967-1004.
- Andersen, J.-V., & Sornette, D. (2004). Fearless versus fearful speculative financial bubbles. Physica A, 337, 565-585.

- Bell, S. (2001). The role of the state and the hierarchy of money. Cambridge Journal of Economics, 25, 149-163.
- Bree, D. S., & Joseph, N. L. (2013). Testing for financial crashes using the log-periodic power law model. International Review of Financial Analysis, 30, 287-297.
- Brock, W. A., Dechert, W. D., Scheinkman, J. A., & Le Baron, B. (1996). A test for independence based on the correlation dimension. Econometric Review, 15, 197-235.
- Campbell, J. Y., Lo, A., & MacKinlay, A. C. (1997). The econometrics of financial markets. Princeton University Press.
- Dale, R. S., Johnson, J. E. V., & Tang, L. L. (2005). Irrational behaviour during the South Sea bubble. The Economic History Review, 58, 233-271.
- Dequech, D. (2013). Is money a convention and/or a creature of the state? The convention of acceptability, the state, contracts and taxes. Journal of Post Keynesian Economics, 36, 251-273.
- Dowd, K. (2014). New private monies: A bit-part player? Institute of Economic Affairs.
- Dwyer, G. P. (2014). The economics of Bitcoin and other similar private digital currencies. Journal of Financial Stability, 12, 10-18.
- European Central Bank. (2012). Virtual currency schemes. Preprint.
- Frisby, D. (2014). Bitcoin: The Future of Money? Unbound.
- Fry, J. (2012). Exogenous and endogenous crashes as phase transitions in complex financial systems. The European Physical Journal B, 85, 405.
- Fry, J. (2014a). Shocks, bubbles and elementary technical trading strategies. The European Physical Journal B, 87, 1.
- Fry, J. (2014b). Multivariate bubbles and antibubbles. The European Physical Journal B, 87, 174.
- Geraskin, P., & Fantazzini, D. (2013). Everything you always wanted to know about log-periodic power laws for bubble modelling but were afraid to ask. The European Journal of Finance, 19, 366-391.
- Grinberg, R. (2012). Bitcoin: An alternative digital currency. Hastings Science and Technology Law Journal, 4, 159-208.
- Ingham, G. (2004). The nature of money. Polity Press.
- Johansen, A., Ledoit, O., & Sornette, D. (2000). Crashes as critical points. International Journal of Theoretical and Applied Finance, 3, 219-255.
- Kindelberger, C. M., & Aliber, R. (2005). Manias, panics and crashes: A history of financial crises (5th ed.). Wiley.
- Lin, L., Ren, R. E., & Sornette, D. (2014). The volatility-confined LPPL model: A consistent model of 'explosive' financial bubbles with mean-reverting residuals. International Review of Financial Analysis, 33, 210-225.
- MacDonell, A. (2014). Popping the Bitcoin bubble: an application of log-periodic power law modelling to digital currency. Preprint.
- Maurer, B., Nelms, T.C., & Swartz, L. (2013). "When perhaps the real problem is money itself!": the practical materiality of Bitcoin. Social Semiotics, 23(2), - 261-277.
- Øksendal, B. (2003). Stochastic Differential Equations: An Introduction (6th ed.). Springer.
- Phillips, P.C.B., Wu, Y., & Yu, J. (2011). Explosive behavior in the 1990s NASDAQ; when did exuberance escalate asset values? International Economic Review, 52(1), 201-226.
- Plasaras, N. (2013). Regulating digital currencies: bringing Bitcoin within the reach of the IMF. Chicago Journal of International Law, 14, 377.
- Reinhart, C.M., & Rogoff, K. (2009). This Time it's Different. Eight Centuries of Financial Folly. Princeton University Press.
- Shiller, R.J. (2005). Irrational Exuberance (2nd ed.). Princeton University Press.
- Shiller, R.J. (2014). Speculative asset prices. American Economic Review, 104(6), 1486-1517.
- Taleb, N.N. (2010). The Black Swan (2nd ed.). Random House.
- Van Alstyne, M. (2014). Economic and business dimensions: why Bitcoin has value. Communications of the ACM, 57(9), 30-33.
- Weber, B. (2014). Bitcoin and the legitimacy crisis of money. Cambridge Journal of Economics, 38(5), 1115-1131.

- Zeira, J. (1999). Informational overshooting, booms and crashes. Journal of Monetary Economics, 43(2), 237-257.