Predicting BTC, ETH, and SOl Prices using a multivariate data set: A Machoine Learning Approach

Eren Muller

July 15, 2024

Contents

Introduction

Predicting stock prices remains one of the most challenging applications of time series analysis, primarily due to the Efficient Market Hypothesis (EMH) proposed by Eugene Fama. According to EMH, asset prices reflect all available information, making it inherently difficult to achieve consistent predictive accuracy (Fama, 1970). While there is an abundance of empirical research focused on stock price prediction (Kara et al., 2011), the literature specifically addressing cryptocurrency price predictions remains significantly less developed.

Cryptocurrencies, led by Bitcoin, have emerged as a significant force in the financial markets. Bitcoin, often referred to as the flagship of the cryptocurrency world, boasts a market capitalization close to one trillion dollars as of July 2024. Alongside Bitcoin, other notable cryptocurrencies such as Ethereum and Solana operate on proof-of-stake protocols, with market capitalizations of approximately 400 billion and 100 billion dollars, respectively. The digital asset market, encompassing both tangible and intangible assets, is projected to reach a staggering 24 trillion dollars by 2025 (Muddasir et al., 2020).

Given the substantial trading volume and increasing market relevance of cryptocurrencies, this study aims to explore whether their prices can be predicted despite the principles of EMH. By focusing on Bitcoin, Ethereum, and Solana, this thesis will employ a multifaceted feature dataset to examine the potential for accurate cryptocurrency price predictions. This investigation not only contributes to the academic discourse but also holds practical significance in enhancing our understanding of the cryptocurrency sector.

Literature Review

Predicting cryptocurrency prices using machine learning (ML) and deep learning (DL) methodologies is a relatively underexplored area in empirical research compared to stock market forecasting. To bridge this gap, our literature review will also consider past works and empirical studies on stock price prediction to glean insights applicable to cryptocurrency forecasting. We will delve into the theoretical foundations of stock price predictability, examining previous research efforts by analyzing their methodologies and the ML and DL techniques they employed. These studies utilized sophisticated feature sets, including various market variables, and we will review the literature to understand which feature sets were used and the rationale behind their selection. Moreover, this review aims to illuminate the impact of marketing-related variables on the predictive power of both stock and cryptocurrency markets. Specifically, we will explore the influence of social media sentiment, such as Reddit discussions, on cryptocurrency prices. By doing so, this comprehensive review seeks to provide a deeper understanding of the methodologies and variables that enhance the predictability of financial markets.

2.1 Efficient market hypothesis

Eugene Fama's Efficient Market Hypothesis (EMH) asserts that stock prices reflect all available information, implying that they always trade at their fair value (Fama, 1970). Fama argues that since all new information is immediately incorporated into stock prices, it is essentially impossible to consistently predict market movements or outperform the market through traditional stock-picking. This underpins the argument for passive index fund investing, which aims to match market returns rather than exceed them.

In a subsequent paper, 'Random Walk in Stock Market Prices,' Fama (1995) argues that technical analysis, or charting, is ineffective for predicting future stock prices in an efficient market, as the market almost instantly adjusts to new information. He provides empirical evidence through the evaluation of strategies such as the 5% filter rule. This strategy involves buying an asset if its closing price increases by 5% and holding it until the closing price decreases by 5%, at which point the investor sells and simultaneously shorts the asset. The short position is held until a subsequent 5% rise, followed by buying and covering. Fama demonstrates that such strategies do not yield higher returns compared to a traditional buy-and-hold portfolio; in fact, the latter often outperforms these filter strategies. This finding supports the notion that stock prices follow a random walk due to the competitive nature of the market.

Fama (1995) also acknowledges that while some analysts may predict the outcomes of new events and thus buy at lower prices anticipating future increases, the existence of many proficient analysts leads to instantaneous price adjustments. This collective efficiency means individual analysts cannot consistently outperform the market. He concludes that for technical analysts to vindicate their methods, they must demonstrate the ability to consistently make better-than-chance predictions.

2.2 Behavioral finance

Behavioral finance challenges Eugene Fama's Efficient Market Hypothesis by arguing that psychological factors and irrational behavior of investors can lead to market inefficiencies. Scholars like Daniel Kahneman and Amos Tversky, who develop prospect theory, demonstrate that cognitive biases such as overconfidence and loss aversion significantly influence investor decisions, often leading to predictable and systematic errors. Kahneman and Tversky (1979) mention that these biases cause stock prices to deviate from their true values, creating opportunities for superior returns through strategic trading, contrary to EMH's assertion that such opportunities are fleeting or non-existent. Behavioral finance thus provides a framework to understand why and how markets might not be entirely efficient.

In 2022, Ho-Jun Kang and his colleagues conduct research to investigate the presence of the Efficient Market Hypothesis (EMH) in the cryptocurrency market. Their study involves testing 893 cryptocurrencies, and the results reveal that only a small fraction of these currencies adhere to the EMH. Specifically, Kang et al. (2022) find that only 54 cryptocurrencies (6%) follow the weak-form EMH, and just 24 (3%) adhere to the semi-strong-form EMH. These findings suggest that the cryptocurrency market demonstrates limited efficiency in information processing. Moreover, the study concludes that most cryptocurrencies do not incorporate past prices or new information into their market prices.

In 2020, Vu Le Tran authors a paper examining the Efficient Market Hypothesis (EMH) within the cryptocurrency market. Tran (2020) concludes that market efficiency is highly variable over time, particularly noting significant inefficiencies before 2017. Tran observes that, over time, the cryptocurrency market is becoming increasingly efficient. Among the cryptocurrencies tested, Litecoin emerges as the most efficient, while Ripple is identified as the least efficient.

2.3 Past attempts at predicting the stock/crypto currency market

In the study conducted by Kara, an artificial neural network (ANN) and support vector machine (SVM) are employed to predict stock price movements on the Istanbul Stock Exchange. The independent variable in this research is a binary indicator reflecting whether the stock price will move up or down the following day. Kara (2011) finds that the SVM, particularly with a polynomial activation function, outperforms all other algorithms, including ANN and backpropagation network (BPN), achieving an accuracy of 71.5%. The feature set for this study comprises various technical analysis (TA) indicators such as the Moving Average Convergence Divergence (MACD), Moving Average (MA), and the stochastic oscillator %K (K%).

In another study focusing on trend deterministic data for stock price prediction, a classification model is used to forecast the up or down movement of stock prices. Patel (2015) mentions that this research incorporates a feature set consisting of binary variables indicating whether a technical indicator suggests an upward or downward trend. The highest performing model in this study is the random forest, which achieves an accuracy of 83.5%. However, a noted limitation of this approach is the binary nature of the technical indicators. The study suggests that incorporating additional levels to represent the degree of movement, such as 'slightly up', 'slightly down', and 'barely down', could enhance the model's accuracy.

In 2020, Chen conducts a study to predict Bitcoin prices using various machine learning methods, including logistic regression and long short-term memory (LSTM) networks. Chen (2020) finds that by utilizing 5-minute interval price data, the model achieves an accuracy of 66%, outperforming more complex neural network models. The feature set in this study is comprehensive, incorporating not only Bitcoin price data but also external factors such as gold spot prices, property and network data, as well as trading and market information.

In another study, Weng (2018) attempts to predict short-term stock prices using ensemble methods. The feature set in this research is diverse, comprising historical stock prices, well-known technical indicators, sentiment scores derived from published newspaper articles, trends in Google searches, and the number of visits to Wikipedia pages. The study demonstrates impressive results, predicting the next day's stock prices with a mean absolute percentage error (MAPE) of less than 1.5%. The best-performing algorithms in this research are boosted decision trees, including XGBoost and AdaBoost.

Usami et al. conduct a study to predict the Karachi Stock Exchange (KSE) using various machine learning algorithms. They employ a classification model to forecast whether the market will go up or down. The feature set for this study is extensive, including oil rates, gold and silver rates, interest rates, foreign exchange (FEX) rates, news and social media feeds, simple moving averages (SMA), and autoregressive integrated moving average (ARIMA) data. Usami et al. (Year) find that the best performing model is the multilayer perceptron (MLP), a type of artificial neural network (ANN), alongside support vector regression (SVR).

In 2022, Mailagaha Kumbure et al. conduct a comprehensive literature review on the application of machine learning and data used for stock market forecasting. This review examines a total of 138 articles related to machine learning in stock markets, providing a detailed overview of the models, markets, and feature sets used in these studies. Mailagaha Kumbure et al. (2022) highlight that the most used machine learning methods are neural networks, support vector machines/support vector regression (SVM/SVR), and fuzzy theories. Additionally, they note that most of these papers incorporate technical indicators in their feature sets.

2.4 Social Media and Sentiment Analysis in the Role of Predicting Stock/Crypto Prices

The influence of social media on Bitcoin prices has been a topic of significant interest in recent research. Feng Mai's 2018 study employs textual analysis and vector error corrections to demonstrate a clear link between social media sentiment and Bitcoin price movements. Mai (2018) shows that bullish posts on social media platforms are associated with higher future Bitcoin prices. This suggests that social media sentiment is a valuable predictor of Bitcoin price fluctuations, highlighting the impact of public opinion and social discourse on cryptocurrency markets.

In addition to social media, online search activity also correlates with Bitcoin price movements. Kristoufek (2013) analyzes Google Trends and Wikipedia page visits, finding strong correlations between these data points and Bitcoin prices. This research suggests that increased online searches and Wikipedia activity, reflecting public interest and awareness, can significantly influence Bitcoin market trends. Complementing these findings, Wesley S. Chan's 2003 study on stock market prediction through news sentiment reveals that positive newspaper headlines often lead to overvaluation of stocks, while negative headlines result in undervaluation. Chan (2003) further notes that this sentiment effect is more pronounced in smaller market capitalization stocks and that investors typically react slowly to sentiment changes. Together, these studies underscore the significant role of public sentiment, whether expressed through social media, search activity, or news headlines, in influencing financial markets.

The predictive power of social media sentiment on cryptocurrency prices has been further explored in recent studies. Olivier Kraaijeveld's 2020 research focuses on the influence of Twitter sentiment on the returns of major cryptocurrencies. Kraaijeveld (2020) concludes that Twitter sentiments indeed have predictive power over cryptocurrency prices, utilizing a lexicon-based sentiment analysis. The study highlights that news disseminated through Twitter can rapidly alter investor sentiments, leading to immediate and significant price movements. This finding emphasizes the crucial role of real-time sentiment analysis in anticipating market trends and price fluctuations in the volatile cryptocurrency market.

Similarly, news sentiment shows a notable impact on Bitcoin prices. Lavinia Rognone's 2020 study analyzes the effect of unscheduled news on Bitcoin compared to traditional currencies using intra-day data from January 2012 to November 2018. Rognone (2020) finds that Bitcoin often reacts positively to news, whether positive or negative, indicating a high level of enthusiasm among investors towards Bitcoin, unlike traditional stock markets. However, specific negative news, such as reports of fraud and cyber-attacks, have adverse effects on Bitcoin prices. The study utilizes RavenPack's real-time news data and employs a Vector Auto-Regressive Exogenous (VARX) model for the analysis. In parallel, Wasit Khan's 2020 research combines social media and news sentiment to predict stock market movements, using a dataset from Twitter and Yahoo Finance. Khan (2020) demonstrates that their predictive model achieves an accuracy of 80% after filtering out spam tweets, underscoring the significant impact of integrated sentiment analysis on market predictions. These studies collectively highlight the importance of sentiment analysis in understanding and forecasting market dynamics across various financial assets.

2.5 Conclusion

The literature review underscores the multifaceted and challenging nature of predicting stock and cryptocurrency prices, emphasizing the importance of robust predictions for effective trading strategies. Research on stock price prediction is extensive, utilizing various machine learning (ML) and deep learning (DL) methods. Studies such as those by Kara et al. (2011) and Patel (2015) highlight the effectiveness of SVM and random forest models, respectively, in forecasting stock prices using technical indicators. Similarly, Chen (2020) and Weng (2018) demonstrate the predictive power of logistic regression, LSTM networks, and ensemble methods for Bitcoin and stock prices, leveraging comprehensive feature sets that include market variables and sentiment scores. The review also notes the evolving efficiency of cryptocurrency markets, with studies like those by Kang et al. (2022) and Tran (2020) revealing limited adherence to the Efficient Market Hypothesis (EMH), indicating significant information processing inefficiencies. The impact of sentiment analysis on market predictions emerges as a critical theme. Research by Mai (2018) and Kraaijeveld (2020) establishes the predictive power of social media sentiment on cryptocurrency prices, while Kristoufek (2013) and Chan (2003) demonstrate similar effects for online search activity and news sentiment on Bitcoin and stock markets. Studies such as Rognone (2020) and Khan (2020) further validate the significant influence of real-time sentiment, integrating social media and news data to achieve high prediction accuracy. These findings collectively highlight the importance of incorporating diverse feature sets, including sentiment analysis, to enhance the predictability of financial markets and challenge traditional notions of market efficiency.

Methodology

3.6 Data Collection

The data collection process for this methodology involves several intricate steps, leveraging multiple data sources and employing various Python-based techniques to gather and process the required information. This section outlines the comprehensive approach taken to ensure the data is robust, reliable, and suitable for subsequent analysis and modeling.

3.6.1 Financial Market Data Acquisition

The first step in the data collection process involves obtaining financial market data from Yahoo Finance. This includes historical prices and volumes for a range of assets such as Bitcoin (BTC), Ethereum (ETH), Solana (SOL), gold, oil, Nvidia, the VIX, S&P 500, and Dow Jones Industrial Average. The Python yfinance library is employed to access the Yahoo Finance API. This library provides an efficient way to retrieve historical price data, allowing us to gather data spanning the past five years. The collected data includes:

- Daily closing prices
- Trading volumes

This data forms the foundation of the dataset, providing crucial information on market trends and volatility.

3.6.2 Google Trends and Wikipedia Data

To gauge public interest and sentiment, data from Google Trends and Wikipedia is incorporated. For Google Trends, the focus is on terms such as 'bitcoin', 'ethereum', 'solana', 'crypto', and 'blockchain'. This data is accessed using the pytrends library, which allows us to download historical search interest data for these terms. Similarly, Wikipedia page views for the same terms are collected using the Wikipedia API. The aggregation of this data involves summing the total page views and total search interest over the specified period. This provides a measure of the general public's engagement and interest in these topics.

3.6.3 Reddit Sentiment Analysis

Sentiment analysis is performed on Reddit posts related to Solana, sourced from three specific subreddits. The data is pulled using the Reddit API, focusing on the top 100 posts at the time of collection.

The information retrieved includes:

- Post titles
- Post descriptions (selftext)
- Number of upvotes
- Date posted

A sentiment analysis pipeline is implemented using the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool. The text data is preprocessed to remove noise, and sentiment scores are computed for each post. These scores are then aggregated by date, weighted by the number of upvotes, to calculate a mean weighted average sentiment for each day. This approach ensures that posts with higher engagement have a more significant impact on the sentiment measure.

3.6.4 Total Value Locked (TVL)

The total value locked (TVL) data for each cryptocurrency is obtained from Defi Lama. TVL represents the total capital held within a blockchain's DeFi ecosystem, providing insight into the level of trust and engagement from the community. This data is crucial for understanding the financial health and adoption of each cryptocurrency.

3.6.5 Technical Indicators

Technical indicators are derived from the historical price data obtained in the first step. A Python function is developed to calculate several key indicators, including:

- Moving Average (MA)
- Exponential Moving Average (EMA)
- Stochastic Oscillators
- Relative Strength Index (RSI)
- Commodity Channel Index (CCI)
- Moving Average Convergence Divergence (MACD)

These indicators are used to create trend deterministic columns, where a value of 1 indicates a bullish signal and 0 indicates a bearish signal. These technical indicators are essential for understanding market trends and potential future movements.

3.6.6 Data Merging and Preparation

The final step in the data collection process involves merging all the collected data on the date field. This step integrates the financial market data, Google Trends and Wikipedia data, Reddit sentiment scores, TVL data, and technical indicators into a single cohesive dataset. The merged dataset is then ready for feature selection and modeling, ensuring that all relevant information is available for comprehensive analysis.

3.7 Feature Engineering

3.7.1 Technical Indicators:

Calculation of moving averages, RSI, MACD, Bollinger Bands, etc., using historical price and volume data.

Indicator	**Formula**	**Trend Det. (1/0)**
10D MA	$MA = \frac{1}{10} \sum_{i=0}^{9} P_{t-i}$	1 if $P_t > MA, 0$ else
30D MA	$3MA = \frac{1}{30} \sum_{i=0}^{29} P_{t-i}$	1 if $P_t > 3MA, 0$ else
%K	$\frac{P_t - LL}{HH - LL} \times 100$, $LL = \min(P_{t-9}, \ldots, P_t)$, $HH = \max(P_{t-9}, \ldots, P_t)$	1 if $\%K_t > \%K_{t-1}, 0$ else
%D	$\frac{1}{3} \sum_{i=0}^{2} \%K_{t-i}$	1 if $\%D_t > \%D_{t-1}, 0$ else
RSI	$\Delta P_t = P_t - P_{t-1}$ $U_t = \frac{1}{14} \sum_{i=0}^{13} \max(\Delta P_{t-i}, 0)$ $D_t = \frac{1}{14} \sum_{i=0}^{13} \max(-\Delta P_{t-i}, 0)$ $RS = \frac{U_t}{D_t}$, $RSI = 100 - \frac{100}{1+RS}$	$\begin{cases} -1 & RSI \geq 70 \text{ \& hold} \\ 1 & RSI \leq 30 \\ 0 & \text{else} \end{cases}$ or 1 if $RSI \leq 30, 0$ else
Momentum	$P_t - P_{t-10}$	1 if $Mom > 1, 0$ else
MACD	$EMA_{12} = P_t \cdot \frac{2}{13} + EMA_{12,t-1} \cdot \frac{11}{13}$ $EMA_{26} = P_t \cdot \frac{2}{27} + EMA_{26,t-1} \cdot \frac{25}{27}$ $MACD = EMA_{12} - EMA_{26}$ $Signal = MACD_t \cdot \frac{2}{10} + Signal_{t-1} \cdot \frac{8}{10}$	1 if $MACD > MACD_{t-1}, 0$ else
CCI	$TP = \frac{H_t + L_t + P_t}{3}$ $SMA_{TP} = \frac{1}{20} \sum_{i=0}^{19} TP_{t-i}$ $MD = \frac{1}{20} \sum_{i=0}^{19} \lvert TP_{t-i} - SMA_{TP} \rvert$ $CCI = \frac{TP - SMA_{TP}}{0.015 \cdot MD}$	$\begin{cases} -1 & CCI \geq 100 \text{ \& hold} \\ 1 & CCI \leq -100 \\ 0 & \text{else} \end{cases}$ or 1 if $CCI \leq -100, 0$ else
Bollinger	$SMA_{20} = \frac{1}{20} \sum_{i=0}^{19} P_{t-i}$ $STD_{20} = \sqrt{\frac{1}{20} \sum_{i=0}^{19} (P_{t-i} - SMA_{20})^2}$ $UB = SMA_{20} + 2 \cdot STD_{20}$ $LB = SMA_{20} - 2 \cdot STD_{20}$	$\begin{cases} -1 & P_t > UB \text{ \& hold} \\ 1 & P_t < LB \\ 0 & \text{else} \end{cases}$ or 1 if $P_t < LB, 0$ else
ATR	$TR = \max(H_t, C_{t-1}) - \min(L_t, C_{t-1})$ $ATR = \frac{1}{14} \sum_{i=0}^{13} TR_{t-i}$	1 if $TR > ATR, 0$ else

3.7.2 Reddit Sentiment Analysis:

Top 100 posts from Reddit communities for Bitcoin, Ethereum, and Solana. Sentiment analysis using VADER lexicon to derive bullish or bearish sentiment scores.

Text Preprocessing

The process began with the collection of Reddit posts, including both the titles and the text of each post. To ensure a comprehensive sentiment analysis, the titles and texts were combined into a single column for each post. The combined text was then preprocessed using the Natural Language Toolkit (nltk) in Python. This preprocessing involved several steps to clean the text and prepare it for analysis. First, the text was converted to lowercase to ensure uniformity. Punctuation and special characters were removed, and the text was tokenized into individual words. Common stop words such as ”the,” ”and,” and ”is” were removed to reduce noise. Finally, lemmatization was performed to convert words to their base forms, such as converting ”running” to ”run.” This preprocessing step ensured that the text was clean and ready for sentiment analysis.

Sentiment Analysis with VADER

For sentiment analysis, we utilized VADER (Valence Aware Dictionary and sEntiment Reasoner), a lexicon and rule-based tool specifically designed to handle sentiments expressed in social media contexts. VADER employs a lexicon-based approach, where each word in its predefined list has an associated sentiment score indicating whether it is positive, negative, or neutral. Additionally, VADER applies rules and heuristics to handle punctuation, capitalization, degree modifiers, and conjunctions. For example, exclamation marks increase the intensity of the sentiment, uppercase words indicate stronger sentiment, and words like ”very” or ”kind of” modify the intensity of the sentiment.

The combined title and text of each Reddit post were passed through VADER for sentiment analysis. VADER first split the text into individual sentences, then matched each word in the sentences against its lexicon to retrieve sentiment scores. The sentiment scores of words in a sentence were aggregated, with adjustments made for the rules and heuristics VADER applies. The sentence-level sentiments were then combined to produce the final sentiment scores for the entire post.

VADER provided four key sentiment scores for each post: positive, neutral, negative, and compound. The positive score represented the proportion of positive words in the text, the neutral score represented the proportion of neutral words, and the negative score represented the proportion of negative words. The compound score, a normalized measure that sums the overall sentiment of the text, ranged from -1 (extremely negative) to +1 (extremely positive). This comprehensive approach allowed us to capture the nuanced sentiment expressed in Reddit posts effectively.

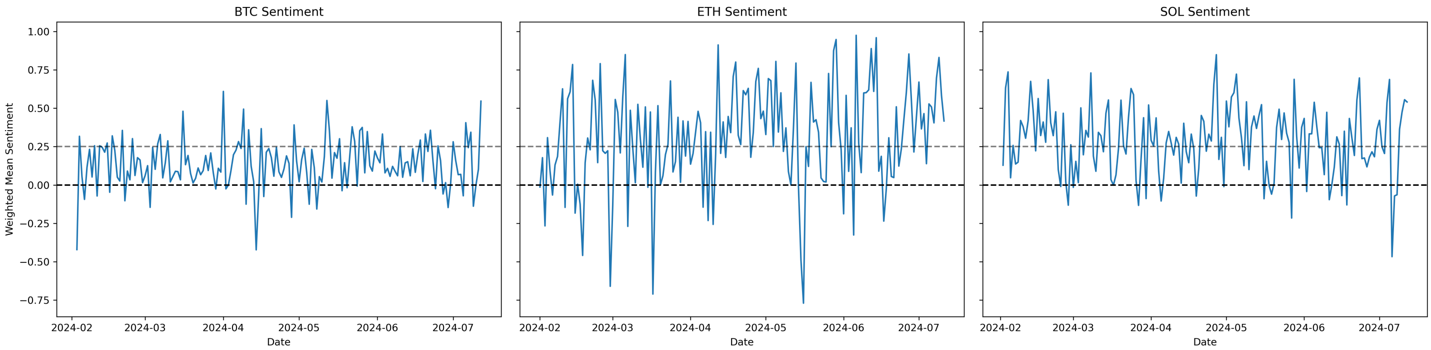

Figure 3.1: Diagram showing Each Subreddits Sentiment

3.7.3 Additional Features:

Inclusion of market indices data (S&P 500, NVIDIA) as predictors.

3.8 Machine Learning Models

3.8.1 Random Forest

Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees. The principle behind Random Forest is to reduce the risk of overfitting by averaging multiple deep decision trees, trained on different parts of the same training set.

Mathematically, for a given input $\mathbf{x}$, the prediction of the i-th tree $h_i(\mathbf{x})$ in a random forest is:

$$f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} h_i(\mathbf{x})$$

where N is the number of trees in the forest.

Key parameters include:

- Number of trees (N)
- Maximum depth of each tree

3.8.2 Support Vector Classification (SVC)

Support Vector Classification (SVC) aims to find the optimal hyperplane that maximizes the margin between the classes. This hyperplane is defined by the support vectors, which are the data points closest to the hyperplane.

The optimization problem for SVC can be written as:

$$\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \ \forall i$$

where $\mathbf{w}$ is the weight vector, b is the bias, $\mathbf{x}_i$ are the input vectors, and y_i are the class labels.

Key parameters include:

- Kernel type (linear, polynomial, radial basis function)
- Regularization parameter (C)

3.8.3 Gradient Boosting Classification

Gradient Boosting Classification is a sequential ensemble technique that builds models in a stage-wise manner. It optimizes for a loss function by adding weak learners to the model, typically decision trees.

The Prediction function for Gradient Boosting is:

$$\hat{y} = F_M(x) = F_0(x) + \eta \sum_{m=1}^{M} h_m(x)$$

Where:

- $\hat{y}$ is the predicted value.
- $F_0(x)$ is the initial prediction.
- M is the total number of iterations (trees).
- η is the learning rate.
- $h_m(x)$ is the prediction from the m-th weak learner (tree).

Key parameters include:

- Learning rate (η): The learning rate is a hyperparameter that controls the contribution of each weak learner (base model) to the final ensemble model. It scales the output of each weak learner before adding it to the accumulated model.
- Number of estimators: This parameter specifies the number of weak learners (usually decision trees) to be included in the ensemble. It defines how many iterations the boosting process will run.
- Maximum depth of each estimator: This parameter sets the maximum depth of the individual decision trees. It controls the complexity of the trees.

3.8.4 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems. This subsection provides a detailed methodology of LSTM networks, explaining their working with equations and their application in the context of cryptocurrency analysis.

LSTM Network Architecture

An LSTM network consists of a series of LSTM cells, each containing three main components: a cell state, and three gates (input gate, forget gate, and output gate).

Cell State The cell state C_t acts as a memory that carries information across the sequence steps, enabling the network to maintain long-term dependencies.

Gates The gates regulate the flow of information into and out of the cell state. They are defined as follows:

- **Forget Gate** (f_t): Decides what information to discard from the cell state.
- **Input Gate** (i_t): Decides which new information to add to the cell state.
- **Output Gate** (o_t): Decides what information to output based on the cell state.

Equations of LSTM

The mathematical formulation of the LSTM gates and cell state updates are given by the following equations:

Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cell State Update

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

Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

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

Where:

- x_t is the input at time step t,
- h_{t-1} is the hidden state from the previous time step,
- σ is the sigmoid activation function,
- W_f, W_i, W_C, W_o are the weight matrices,
- b_f, b_i, b_C, b_o are the bias vectors.

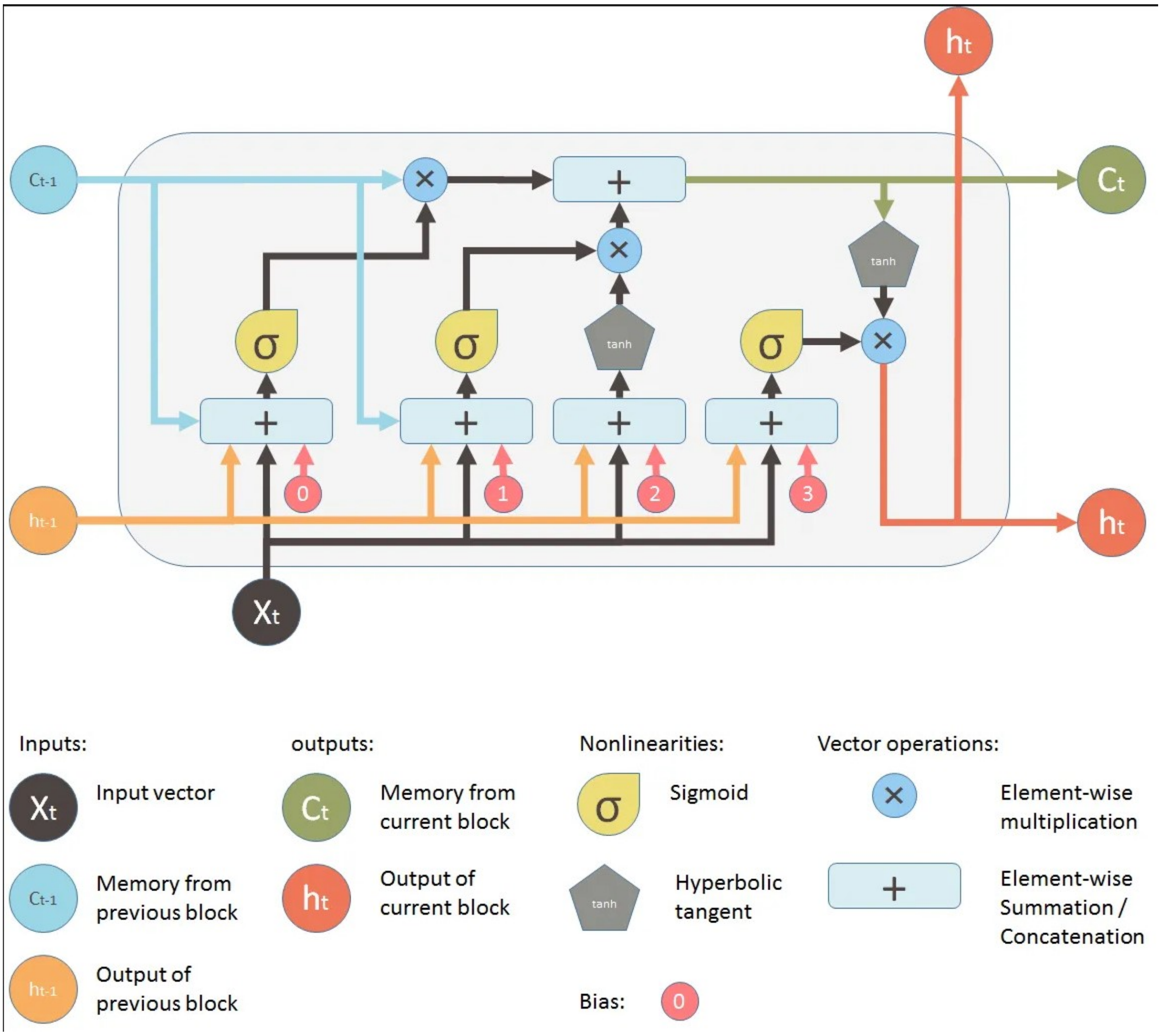

Figure 3.2: LSTM Diagram

3.8.5 Hidden Markov Model (HMM)

Hidden Markov Models (HMMs) are statistical models used for sequential data, where the system being modeled is assumed to be a Markov process with hidden states.

An HMM is defined by:

- Number of hidden states (N)
- Transition probability matrix $A = \{a_{ij}\}$, where $a_{ij} = P(s_{t+1} = j|s_t = i)$
- Emission probability matrix $B = \{b_j(o_t)\}$, where $b_j(o_t) = P(o_t|s_t = j)$

The model works by estimating the sequence of hidden states given the observed data, typically using algorithms such as the Forward-Backward algorithm or Viterbi algorithm.

$$P(O|\lambda) = \sum_{all\ paths} P(O, Q|\lambda)$$

where O is the sequence of observations, Q is the sequence of hidden states, and $\lambda = (A, B, \pi)$ represents the model parameters.

An illustrative image showing the state transitions and observation emissions would help explain HMMs more effectively.

3.9 Model Evaluation

- **Metrics:** Accuracy, precision, recall, F1-score, ROC-AUC for classification; MSE, MAE, R-squared for regression.
- **Cross-Validation:** K-fold cross-validation.
- **Model Comparison:** Performance metrics comparison to select the best model.

Results

4.10 Random Forest Classifier

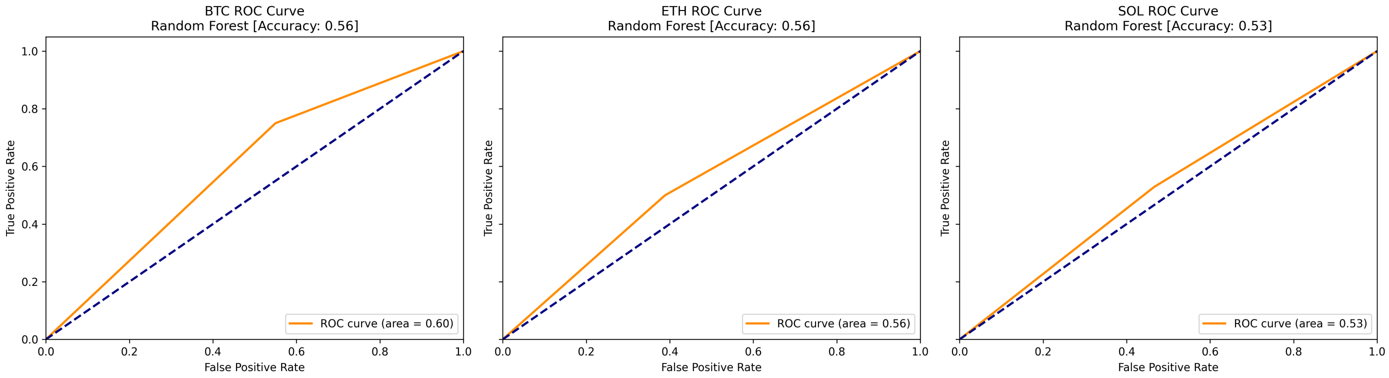

Figure 4.3: Random Forest ROC and Accuracy Results

text

4.11 Support Vector Machine

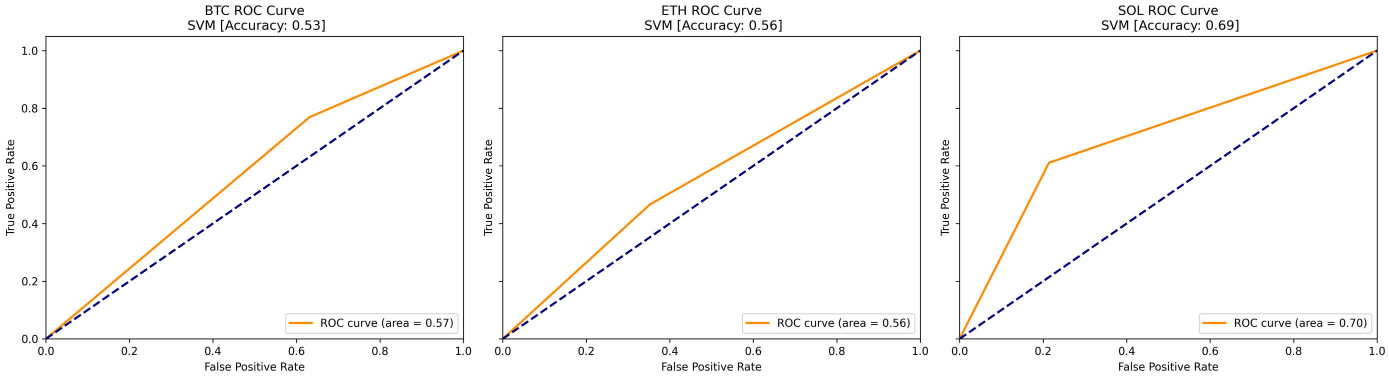

Figure 4.4: SVM ROC and Accuracy Results

text

4.12 SHAPLEY Values

text

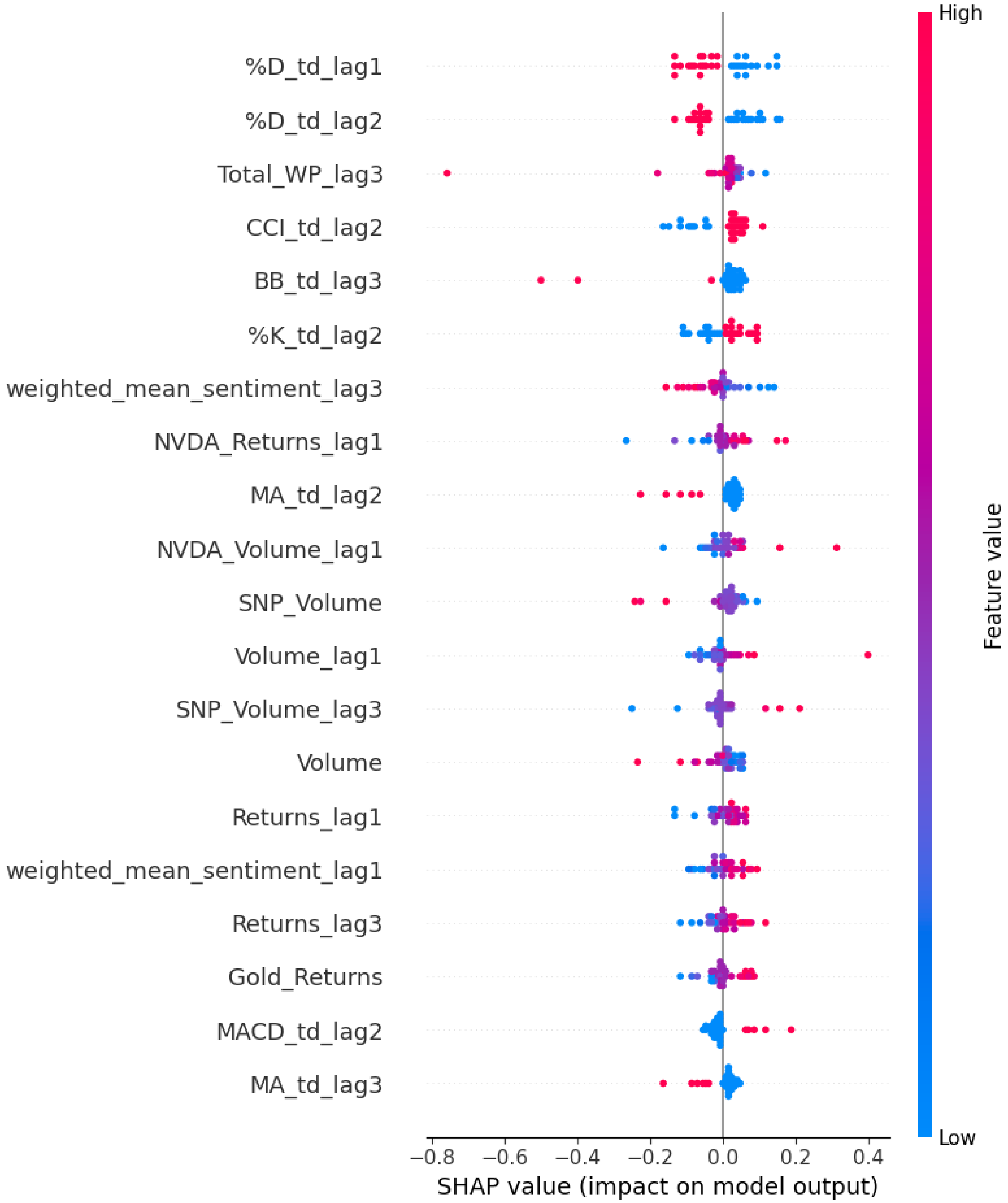

Figure 4.5: SHAP Beeplot SVM

4.13 XGBoost

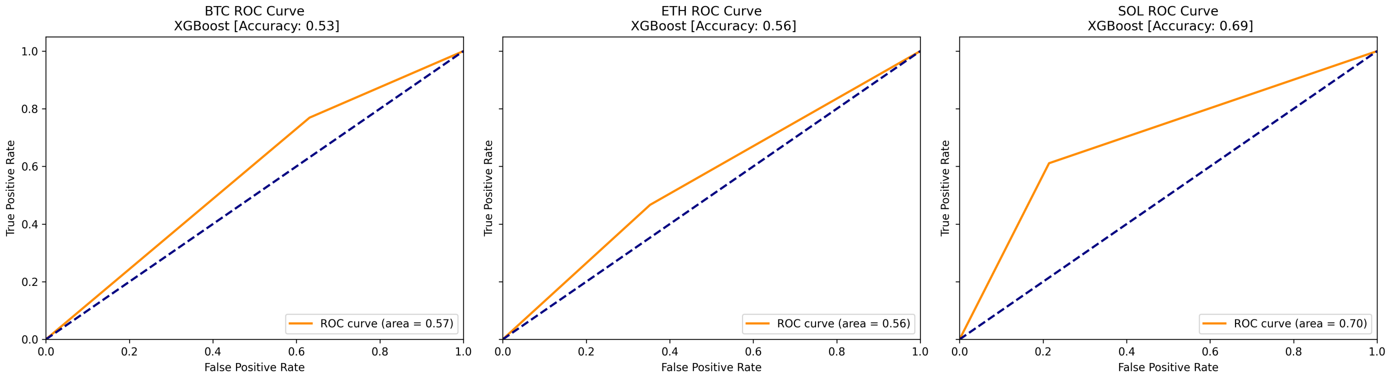

Figure 4.6: XGBoost ROC and Accuracy Results

text

4.14 LSTM

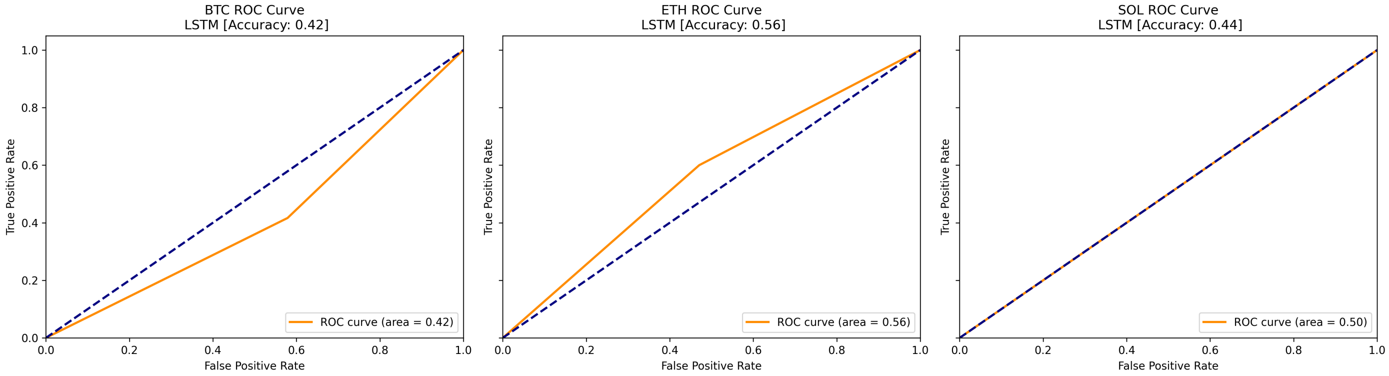

Figure 4.7: LSTM ROC and Accuracy Results