Leveraging the Power of Hybrid Models: Combining ARIMA and LSTM for Accurate Bitcoin Price Forecasting

by

Maryam Gholipour

A thesis submitted to the
Department of Computer Science
in conformity with the requirements for
the degree of Master of Science

Bishop's University
Canada
August 2023

Abstract

This thesis conducts an extensive analysis of Bitcoin daily price forecasting, using ARIMA, LSTM and the new method of the ARIMA-LSTM paralleling by weight of regression coefficients. Feature engineering emerged as a key determinant by integrating technical indicators as supplementary features, including Simple Moving Average, Exponential Moving Average, RSI and MACD. Certain features, including Close, Returns, Upper Bollinger Band, and Lower Bollinger Band, emerged as influential contributors to enhance model performance. Research is structured around three distinct time periods, offering insights into market stability, heightened volatility, and a comprehensive overview of the entire timeframe. Among the models examined, LSTM model, with the evaluation metrics of MAPE = 1.37%, RMSE = 442.81, and R_2-score= 0.99. outperformed other models. While LSTM excelled, for the multivariate forecasting involving “Close” and “Returns” features, the ARIMA-LSTM improved overall performance as it reduced MAPE from 5.77% to 1.98% and increased R_2-score from 0.41 to 0.96.

Keywords: Time-Series Forecasting, Deep Learning, ARIMA, LSTM, Hybrid ARIMA-LSTM, Feature Engineering,

Acknowledgments

I would like to extend my heartfelt gratitude to my supervisor, Professor Madjid Allili for his invaluable guidance and support throughout this research journey.

I am also grateful to the esteemed faculty members of the Computer Science department and for their insightful guidance and support throughout my study.

Additionally, I would like to express my deepest appreciation to my family, who have stood by me with unwavering support. My heartfelt thanks go to my husband, Farzad, without your support, understanding, and constant motivation, navigating through the challenges of this journey would have been daunting. To my mother, whose encouragement have been a constant source of inspiration for me.

Lastly, I wish to honor and remember my late father, who passed away during the course of my master's thesis. His memory will forever serve as a guiding light of inspiration and strength for me.

Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Introduction

Bitcoin, the first and most important decentralized digital crypto currency, was developed and introduced by an anonymous person or group of people using the pseudonym Satoshi Nakamoto in 2009. [1] The Bitcoin network is rapidly expanding as a result of the rise in transaction volume. As of August 2023, its market capitalization is valued at more than 500 billion USD. [2]

As the Bitcoin market continues to mature, precise predictions of its prices can have significant implications for investors, traders, financial institutions, and policymakers. These predictions serve to optimize trading strategies, mitigate risks, and inform investment decisions within the dynamic and volatile cryptocurrency domain. In this thesis, two methodologies for Bitcoin price prediction are examined: the Autoregressive Integrated Moving Average (ARIMA) and the Long Short-Term Memory (LSTM) networks. Additionally, an innovative hybrid ARIMA-LSTM approach is introduced, combining the strengths of both models for enhanced forecasting accuracy.

Numerous studies have already explored predicting Bitcoin and cryptocurrency prices, but understanding this market is still one of the most discussed subjects and difficult to analyze. This study introduces a new method to improve predictions by using data sources like historical data and technical indicators. These findings could enhance prediction models, helping investors make smarter decisions in the cryptocurrency world.

This study is driven by three key objectives:

1. Predicting Bitcoin prices using three distinct models: ARIMA, LSTM, and a hybrid ARIMA-LSTM approach.
2. Comparing the performance of these models in terms of accuracy and reliability in predicting Bitcoin prices.
3. Evaluating the impact of incorporating technical indicators as additional features on the prediction outcomes of these models.

With these objectives driving the research, this study strives to contribute insights to the field of cryptocurrency forecasting, equipping stakeholders with informed decision-making tools in the ever-evolving world of digital finance.

1.2 Thesis Outline

Chapter 2 provides an in-depth exploration of the existing research in the field of Bitcoin price prediction. Previous works that have employed ARIMA, LSTM, and hybrid ARIMA-LSTM methods are reviewed. In addition, it provides the fundamental concepts and principles of the research. Related works that have contributed to the understanding of time series analysis and deep learning techniques are investigated.

Chapter 3 describes the dataset used and the analysis performed on it before it was used. Then, describes the time series analysis techniques that were utilized to make operational decisions on the applied algorithms. Finally, the algorithms are proposed, along with the metrics and statistical techniques that have been used to evaluate their performance. This research focuses on quantitative prediction, as it relies on Bitcoin's historical data. Regression functions using supervised learning strategies are used to reach this goal. The prediction models are coded in Python 3.9.16, utilizing libraries such as Keras, Tensor Flow, Scikit-learn, Statsmodels, Numpy, Pandas, Seaborn, matplotlib, and yfinance. The models are developed on the Google Colab Free version, which provides a virtual RAM space of 12.7 GB for running the code efficiently.

Chapter 4 provides the results obtained from the experimentation of each model. The outcomes of ARIMA, LSTM, and hybrid ARIMA-LSTM models are showcased. Comprehensive evaluation metrics, prediction plots, and insightful analyses for each time frame are featured.

In Chapter 5, the cumulative findings of the study are summarized.

Chapter 2: Literature Review

2.1 Review of Previous Studies

The use of advanced machine learning methods has led to a fascinating change in the way financial predictions are made in recent years. Two significant approaches in this field are the traditional Autoregressive Integrated Moving Average (ARIMA) and the deep learning Long Short-Term Memory (LSTM) models. The ARIMA model is proficient at capturing past patterns by utilizing its autoregressive and moving average components. On the other hand, the LSTM model, excels in grasping sequential data dynamics, particularly complex temporal dependencies, using its gated memory cells.

In 2020, [3] presented a comprehensive exploration of time-series forecasting for Bitcoin prices, employing high-dimensional features and machine learning models. The study stands as the pioneering study to consider price indicators up until December 31, 2019, and to provide accurate forecasts for end-of-day, short-term, and mid-term BTC prices utilizing four machine learning techniques including ANN, SANN, SVM, and LSTM. The dataset utilized in this study was sourced from bitinfocharts.com. the data was partitioned into three distinct intervals. The first interval covered the time period from April 1, 2013, to July 19, 2016. The second interval spanned from April 1, 2013, to April 1, 2017. Lastly, the third interval, which was previously unexplored in existing literature, consisted of data from April 1, 2013, to December 31, 2019, making it the most extensive interval under consideration. The developed models all showcased satisfactory performance, with LSTM showing the best overall results. For daily price predictions, the Mean Absolute Percentage Error (MAPE) was found to be as low as 1.44%. However, in longer timeframes, from seven to ninety days, the MAPE varied between 2.88% and 4.10%.

In [4], the authors provided a comprehensive analysis about the performance of LSTM and ARIMA models in forecasting short-term Bitcoin values. This study utilizes a dataset of Bitcoin (BTC) historical Closing Prices, covering a period of one year from December 21, 2020, 0:00 to December 21, 2021, 16:00. The data points were collected at intervals of 10 minutes. The dataset has been partitioned into training and test subsets, with a distribution of 99.5% for training and 0.5% for testing. The results of the study provide interesting observations regarding each model’s behaviors. The ARIMA model better performs in capturing the increasing trend of Bitcoin, which is consistent with the characteristics of the ARIMA model. It is important to acknowledge that the performance of ARIMA may fluctuate when confronted with a downward trend. On the other hand, LSTM stands out as the top performer, achieving prediction accuracy of 99.73%. In this context, 'accuracy' serves as the key evaluation metric.

In [5], researchers integrated machine learning methodologies with sentiment analysis to provide a multidimensional perspective on cryptocurrency price movements. The research utilized two datasets. The historical data has been sourced from cryptocurrency exchanges, including Coinmarketcap, Bitstamp, Coinbase, and Blockchain Info. The second dataset comprised tweets sourced from Twitter and posts extracted from Reddit. The incorporation of this combination introduced a social and emotional dimension to the investigation. Dataset is divided into train/test with 70% split ratio. The comparative evaluation of ARIMA and LSTM model is conducted by the Root Mean Square Error (RMSE) metric. The LSTM model demonstrated superior performance in terms of RMSE values, with 198.448 for single feature and 197.515 for multi-feature, in contrast to the ARIMA model's RMSE value of 209.263. The (RMSE) of the LSTM model is minimized due to the variability in the data, ranging from 0 to 10000 USD, and the substantial fluctuations observed in the closing prices.

Authors in [6] were the first to combine dimension engineering on Bitcoin price granularity with advanced machine learning methodologies, resulting in improved predictive precision. The study used two datasets. One, obtained from CoinMarketCap.com, containing daily Bitcoin prices as well as a variety of variables such as network and property data, trading and market data, media and investor influence and gold spot price from February 2, 2017, to February 1, 2019. The second dataset was collected from Binance with a 5-minute interval at a high frequency spanning from February 2, 2017, to February 1, 2019. The statistical approach (LR and LD)) has shown proficiency in forecasting daily data, achieving an average accuracy rate of 65.0%, surpassing the performance of machine learning models which achieved an accuracy rate of 55.3%. In contrast, the machine learning models showed better results with 5-minute interval dataset. The LSTM model achieved a result of 67.2% in accuracy.

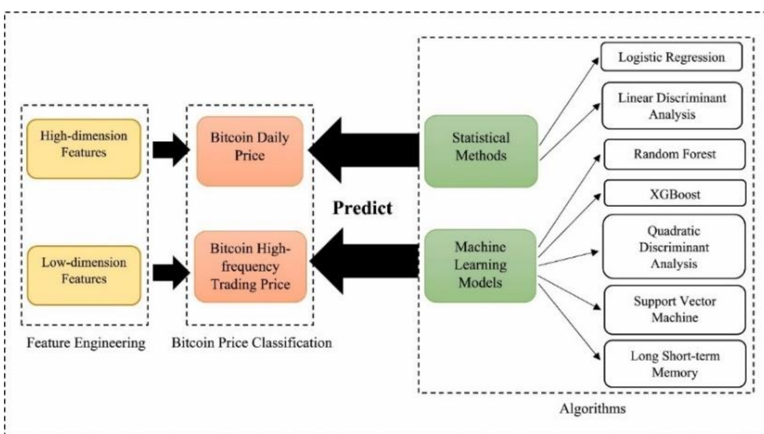


Figure 2.1. Overview of the research framework. Image from [6]

In the context of financial prediction, both ARIMA and LSTM exhibit distinct strengths and weaknesses. Surprisingly, despite the inherent strengths of ARIMA and LSTM models, the financial domain lacks research that delves into their combined application. Although there have been separate investigations into both models in the field of financial prediction, the combination of their capacities in this context has not been fully investigated. This study gap highlights the necessity to conduct an inquiry into the potential of combining ARIMA and LSTM approaches to achieve improved predictive accuracy in financial forecasting.

Interestingly, [7] introduced a hybrid ARIMA-LSTM model, but within a different context: the prediction of COVID-19 cases during pandemic. The dataset utilized in this study was obtained from confirmed COVID-19 cases in China, retrieved from Hopkins University's epidemic website. The data covers the period from 1 January 2021 to 10 October 2022. This model achieved the following metrics: mean squared error (MSE) of 4049.913, root mean squared error (RMSE) of 63.639, mean absolute percentage error (MAPE) of 0.205, coefficient of determination (R2-score) of 0.837, and mean absolute error (MAE) of 44.320. The study found that the ARIMA-LSTM combined regression prediction model had superior performance compared to both the ARIMA and LSTM models individually as well as the SVR model. To validate the model's effectiveness, the analysis was extended to include epidemic data from India. And the results indicated that the hybrid ARIMA-LSTM model demonstrated better alignment with actual test sample values compared to the SVR prediction model.

2.2 Fundamentals and Concepts

2.2.1 ARIMA model and its Components

In this section, the relevant literature on time series analysis and the key concepts necessary for understanding the ARIMA model will be reviewed, including white noise and random walk theory, stationarity, unit root and Augmented Dickey Fuller (ADF) test, transformation to stationary, Autocorrelation Function and Partial Autocorrelation Function analysis.

2.2.1.1 Time Series

A time series consists of sequential observations recorded at regular intervals. Time series data can be broken down into components such as trend, which refers to the long-term pattern and the upward or downward changes in the data ; seasonality, which is the pattern that happens at consistent intervals in a series; cyclical variation, which is the repeated fluctuations in a time series and is not related to seasonality; and a non-systematic component called irregular variation or remainder, which refers to the random noise in a time series. As demonstrated in Figure 2.2 [8] depending on the kind of trend and seasonality, time series can be defined as either an additive model or a multiplicative model. [9]–[12]

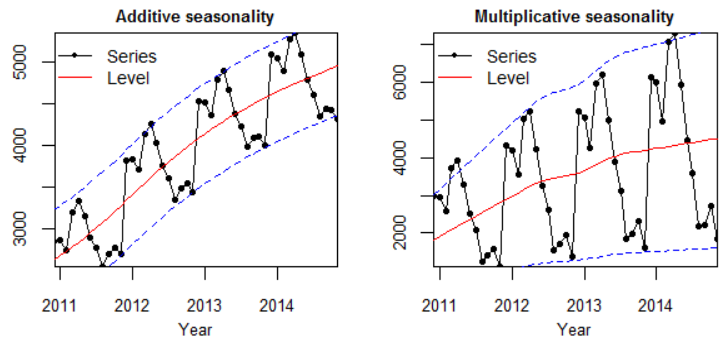


Figure 2.2. Additive seasonality versus Multiplicative seasonality. Image from [8]

2.2.1.2 White Noise and random walk theory

A time series is said to be "white noise" if its variables are random and uncorrelated, with constant mean and variance, where no algorithm could possibly forecast its behavior. This will serve as a benchmark against which the predictive power of a model is evaluated.

Random walk theory has implications for financial markets, including the Bitcoin market. It suggests that price movements are random and independent of the previous prices, which means that future behavior of the series is unpredictable. This model is also a useful benchmark for analyzing the behavior of bitcoin prices. In 2016, authors in [14] found evidence against the random walk hypothesis using the ADF test, indicating that Bitcoin prices exhibit serial correlation and are not independent of their past prices. However, another study found support for the random walk hypothesis using the KPSS test, indicating that Bitcoin prices are stationary and follow a random walk model.

2.2.1.3 Unit Root and ADF test

A time series is said to be stationary if its statistical properties, including mean and variance, do not fluctuate over time and the future behavior of the series can be predicted based on its past behavior. To determine stationarity, the Augmented Dickey-Fuller (ADF) test is commonly used which examines the presence of a unit root by involving null and alternative hypotheses in a time series [15]. The Null hypothesis (H0) states the presence of a unit root, indicating time series is non-stationary. While the alternative hypothesis (H1) suggests that the time series is stationary. The ADF test equation is expressed as follows:

$$yt = c + \beta_t + \alpha y_{t-1} + \phi_1 \Delta Y_{t-1} + \phi_2 \Delta Y_{t-2} + \ldots + \phi_p \Delta Y_{t-p} + e_t \tag{2.1}$$

where, y_{t-1} = lag 1 of time series and ΔY_{t-1} = first difference of the series at time (t-1) The significance level, typically set at 1%, 5%, or 10% is a threshold or fixed level of probability used to evaluate whether the null hypothesis should be rejected. If the calculated p-value is less than the significance level, this means that the observed result was significantly different from what would have been expected by chance, then H0 is not true, and the time series is said to be stationary. Otherwise, it is non-stationary. [16]

2.2.1.4 Transforming to stationary

Techniques such as differencing, logarithmic transformation, are commonly applied to remove trends, seasonality, stabilize the mean and variance of the series, and finally transform a non-stationary time series into a stationary one. Logarithmic transformation takes the logarithm of the observations, which helps stabilize the variance and enables series with exponential growth or large fluctuations for prediction. In Differencing method, the first difference is obtained by subtracting the value at time (t-1) from the value at time t. By employing the first-difference method, we can effectively achieve stationarity in the Bitcoin price series, enabling us to apply various modeling techniques for accurate predictions. [17]–[19]

$$y'_t = y_t - y_{t-1} \tag{2.2}$$

In terms of comparing the efficacy of these techniques, researchers in [20] have compared the differencing and logarithmic transformation in removing non-stationarity from Bitcoin prices and discovered that both techniques can be beneficial. They also discovered that first-order differencing is very useful for reducing seasonality and trend. In addition, [21] demonstrated that the first difference in the logarithmic returns of Bitcoin prices is stationary. In this research, the first-difference method is used for transforming the price series into stationary models.

2.2.1.5 ACF and PACF

The autocorrelation function (ACF) helps identify the presence of the correlation coefficient between a time series and its lagged version at each lag. whereas the partial autocorrelation function (PACF) calculates the correlation between two variables after eliminating the impact of the intermediate variables. The results obtained from analyzing ACF and PACF plots can be used to identify the appropriate lag order of moving average (MA) and autoregressive (AR) components, respectively. These lag orders play an important role in fitting the ARIMA model accurately. [15], [22]

2.2.1.6 ARIMA Model

The ARIMA model is widely used since it is simply a linear regression for forecasting time series. It is best suited for data with high and consistent correlation, as it does regression based on its own past data. ARIMA models are composed of three components: autoregression (AR), integration (I), and moving average (MA). The AR component forecasts future values of the time series based on past values, the MA component predicts future values based on residuals of past observations, and the I component is used to remove seasonality or trends from time series data. The ARIMA (p,d,q) model is defined by the following equation:

$$y'(t) = c + \phi 1 * y'(t-1) + \cdots + \phi p * y'(t-p) + \theta 1 * \varepsilon(t-1) + \cdots + \theta q * \varepsilon(t-q) + \varepsilon t \tag{2.3}$$

Here, y'(t) is the value of the differenced time series at time t, c is a constant, $\phi(1)$ through $\phi(p)$ are the AR coefficients, $\varepsilon(t)$ is the error term at time t, $\theta(1)$ through $\theta(q)$ are the MA coefficients, and d is the degree of differencing (I component). The parameters p, d, and q correspond to the order of the autoregressive, degree of differencing, and order of the moving average components, respectively. [23], [15] ARIMA is a powerful model that can give better results than Deep Learning (DL) models in the stock market. However, when it comes to cryptocurrency, where seasonality is extreme, ARIMA cannot handle trend and seasonality very well. This may require careful parameter adjustment and an in-depth understanding of the data.

Another limitation is that it may not notice the non-linear pattern of the price movements, leading to inaccurate predictions. In [24], study showed that DL models performed better in terms of accuracy than ARIMA, which had the highest error rates among all models, with an MAE of 529.4. On the other hand, [25] compared ARIMA, LSTM, and GRU on bitcoin price and found that ARIMA outperformed other models for monthly series. However, in terms of RMSE and MAPE metrics, the GRU model outperformed both the ARIMA and LSTM models for daily time series. Also [26] have combined ARIMA with LSTM and proposed a new model called "LSTM with AR(2)". Their results show that this model outperforms other models, including LSTM and ARIMA.

2.3 Deep Learning and Time Series Prediction

Deep learning, a subset of machine learning, employs a representation like that of the human brain with the help of multiple layers of neural networks. In the field of time series prediction, recent research suggests that DL models have the potential to outperform traditional machine learning (ML) models. In a comparative analysis by Mudassir et al. [3] various models including LSTM, SVM, SANN, and ANN were employed to predict future bitcoin prices for the daily, weekly, monthly, and 90 days intervals. LSTM was the most successful among other models as it can better handle the complex pattern and the high volatility of the prices when it comes to bitcoin price prediction. Furthermore, DL models outperform Machine Learning models in terms of accuracy for large datasets due to their ability to learn features from the data itself without the need for manual feature engineering. Figure 2.3 [27] indicates that the accuracy of ML models falls while that of DL models improves as data volume increases.

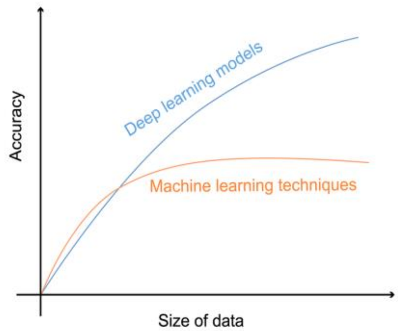


Figure 2.3: The accuracy of ML approaches compared to the accuracy of DL models with respect to data size. Image from [27]

2.3.1 Recurrent Neural Networks (RNN)

Recurrent neural networks (RNNs) are essentially recurrent "for" loops that use data from earlier iterations. After each input is evaluated by a single RNN cell, the cell's output is forwarded to the next cell in the network, and so on until the last cell is reached. [28]

$$
\begin{aligned}
a(t) &= tanh\,(W_{ax}x(t) + W_{aa}a(t-1) + b_a) \\
\hat{y}(t) &= softmax(W_{ya}a(t) + b_y)
\end{aligned}
\tag{2.4}
$$

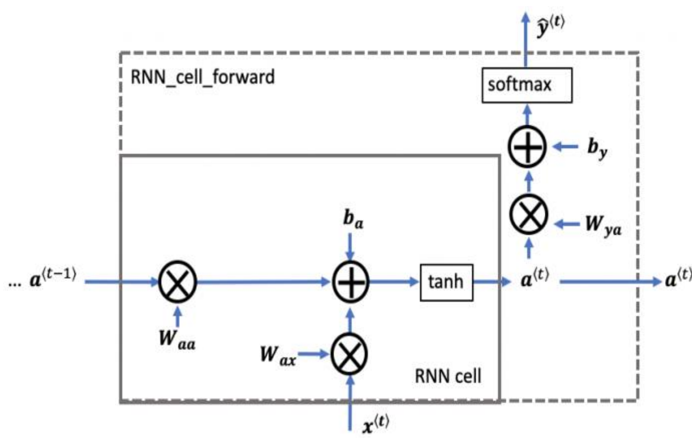


Figure 2.4. RNN network

One main downside of a simple RNN is its inability to effectively capture long-term dependencies. As more layers are added to the network, it becomes untrainable, a phenomenon known as the "vanishing gradient problem.". [29] This occurs when the backpropagation algorithm cycles back through each neuron in a neural network to adjust their weights. This shift is calculated using multiplicative mathematics. As a result, the gradient computed at the network's depths tend to diminish exponentially as it is propagated backwards.

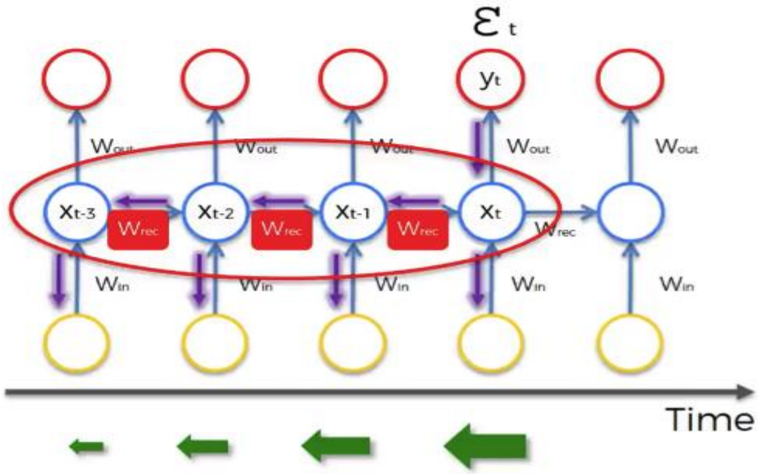


Figure 2.5. Vanishing Gradient Problem. Image from [29]

$$\frac{\partial X_t}{\partial X_k} = \prod_{t \geq i > k} \frac{\partial X_i}{\partial X_{i-1}} = \prod_{t \geq i > k} {W_{rec}}^T diag\left(\sigma'(x_{i-1})\right)$$
$$\frac{\partial \varepsilon}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\partial \varepsilon_t}{\partial \theta} \tag{2.5}$$

2.3.2 Long Short-Term Memory (LSTM)

LSTM (long short-term memory), a powerful variant of the Simple RNN that was created in 1997 by Sepp Hochreiter and Juergen Schmidhuber to overcome the vanishing gradient problem. [30] The key characteristic of LSTM models lies in their ability to retain and use information over longer time intervals. As demonstrated in Figure 2.6, the input gate determines the importance of incoming information.[31] The memory cell can remember its previous state because of the non-linear gating units that regulate the flow of new data into the cell, allowing the network to capture long-term dependencies and complex patterns in data. The forget gate manages the retention or deletion of existing information in memory cells, and the output gate generates the final output based on input and the current state of the memory cells. This makes LSTM well-suited for Bitcoin price prediction, as it can handle the inherent volatility and non-linear dynamics of cryptocurrency markets.

$$
\begin{aligned}
a(t) &= \Gamma_o{}^t * tanh(c(t)) \\
\Gamma_f{}^t &= \sigma(W_f[a\langle t-1\rangle, x\langle t\rangle] + b_f) \\
\Gamma_u{}^t &= \sigma(W_u[a\langle t-1\rangle, x\langle t\rangle] + b_u) \\
c(t) &= \Gamma_f{}^t * c(t-1) + \Gamma_u{}^t * c^{\sim}(t) \\
\Gamma_o{}^t &= \sigma(W_o[a\langle t-1\rangle, x\langle t\rangle] + b_o)
\end{aligned}
\tag{2.6}
$$

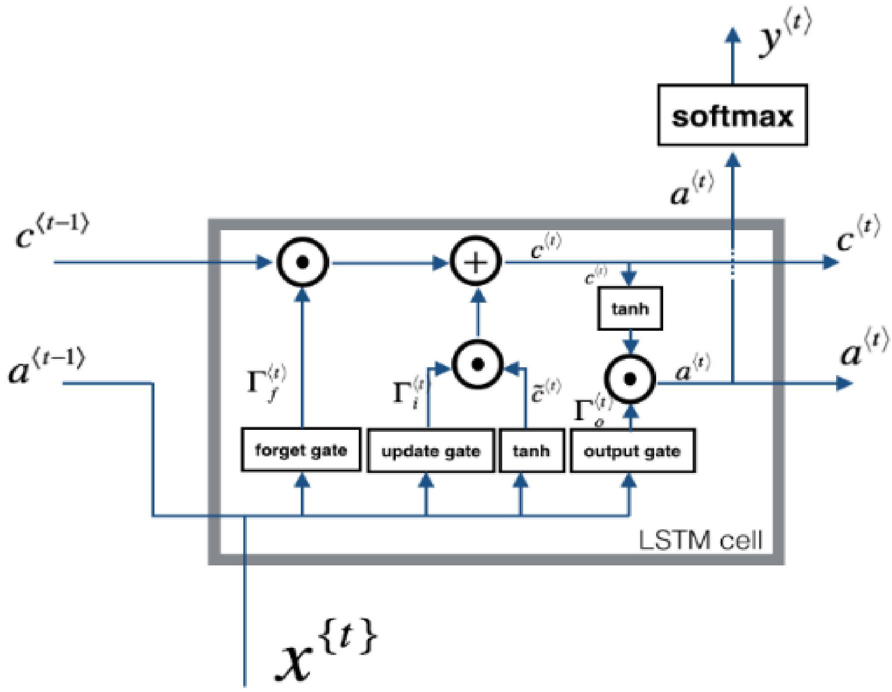


Figure 2.6. LSTM network. Image from [31]

Despite its strengths, LSTM model also comes with challenges. The complexity of LSTM networks requires a larger amount of training data and longer training times compared to simpler models. The risk of overfitting, which occurs when the model becomes too complex and starts to memorize the training data instead of learning patterns. Additionally, selecting appropriate hyperparameters and network architectures for LSTM models can significantly impact their performance.

2.4 Hybrid ARIMA-LSTM Model

In 2022, [7] introduced a hybrid approach, combining the strengths of both ARIMA, and LSTM. The study presents two main approaches for effectively combining ARIMA and LSTM: the series-based strategy and the parallel strategy.

1. In the series-based approach, LSTM predicts ARIMA model residuals and adjusts the ARIMA predictions accordingly.
2. In the parallel approach, specific weights are assigned to the ARIMA and LSTM predictions using various methods such as equal weight averaging and weighted average based on error variance. The study also introduced a novel method of using regression coefficients as model weights, offering an alternative way to achieve effective hybrid predictions.

$$y = \beta_0 + \beta_1 x_{LSTM} + \beta_2 x_{ARIMA} \tag{2.7}$$

Where x_{LSTM} signifies the predicted value from the LSTM model, x_{ARIMA} represents the predicted value from the ARIMA model. β_1 denotes the weight attributed to the LSTM model. Similarly, β_2 signifies the weight assigned to the ARIMA model. β_0 is the constant term of the regression model. By adopting this method, the parallel ARIMA-LSTM model strives to strike a balance between the predictive contributions of both models, enhancing the accuracy of the final prediction. [7]

Chapter 3: Methods and Implementation

3.1 Dataset

The historical OHLCV daily data on Bitcoin prices used in this study was extracted from Yahoo Finance's API [32] using the “yfinance library”. [33] The data collection process spanned a period of nine years, from September 17, 2014, to the end of the March 2023. Daily bitcoin price data includes open, high, low, closing prices, trading volume, dividends, and stock splits. all denominated in USD.

3.1.1 Data pre-processing

After the data collection process, the raw data was cleaned and preprocessed for further analysis. First, the stock split and dividends columns were removed from the dataset since Bitcoin does not have stock splits or dividends. In addition, a thorough check was conducted to ensure the dataset was complete and free of any missing or null values.

Table 3.1 shows the description of the dataset:

	open	high	low	close	volume
count	**3248.000000**	**3248.000000**	**3248.000000**	**3248.000000**	**3248**
mean	**13761.799017**	**14094.033270**	**13398.411521**	**13769.294013**	**19,436,270,000**
std	**16016.188302**	**16413.201606**	**15563.214615**	**16013.610699**	**19,436,270,000**
min	**176.897003**	**211.731003**	**171.509995**	**178.102997**	**5,914,570**
max	**67549.734375**	**68789.625000**	**66382.062500**	**67566.828125**	**350967900000**

Table 3.1. BTC data statistical description

3.1.2 Exploratory data analysis (EDA)

The close price movements of bitcoin from the start point of the dataset until March 2023, ranges from 178 to 67,566 USD. The price fluctuations during the earlier years (2014-2016) were relatively stable, ranging between 200-500 USD. Given the limited variability of prices during this period, it was considered insignificant and removed from further analysis. Consequently, the remaining dataset from 2016 onward was used for training the data in this study as demonstrated in Figure 3.1.The daily close price of bitcoin and its mean on a weekly, monthly, yearly, and seasonal basis were plotted in Figure 3.2 to gain a better understanding of the price movement. The candlestick view of the bitcoin prices is also provided in Figure 3.3.

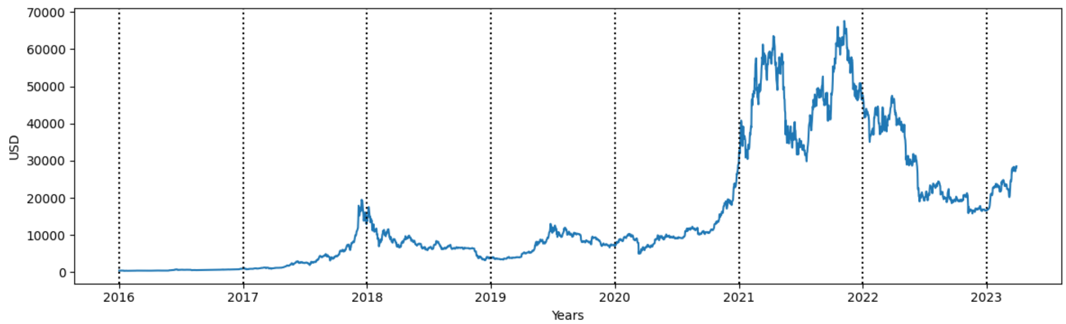


Figure 3.1. BTC close prices from 2016-2023

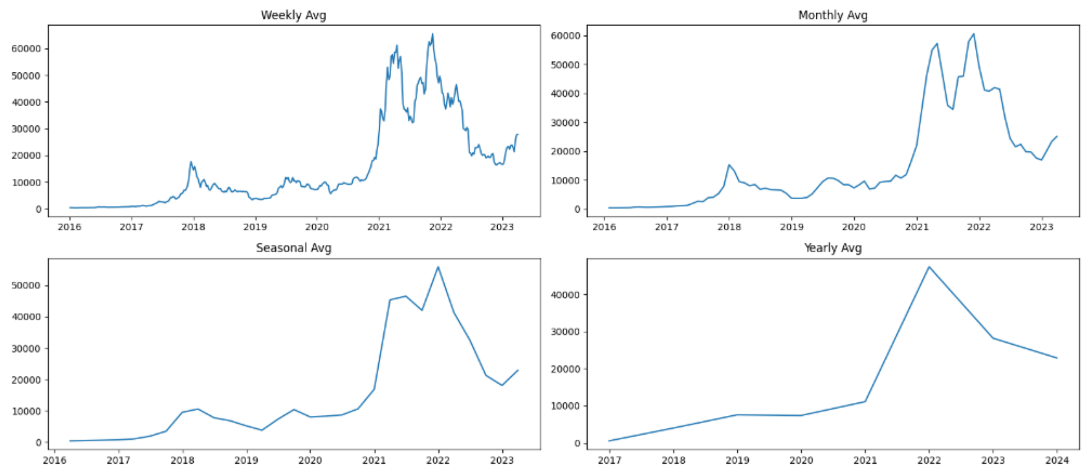


Figure 3.2. BTC close prices on weekly, monthly, seasonal, and yearly basis

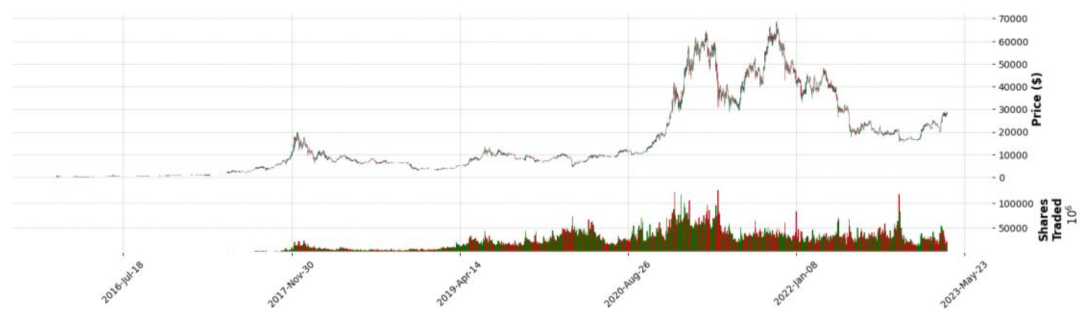


Figure 3.3. BTC close price candlestick view

As demonstrated, after a prolonged period of stability from 2014 to 2017, bitcoin rose in the middle of 2017. The sharp drop that followed the peak in the middle of 2018 continued the upward trend into 2020. Then, during the first wave of the COVID-19 epidemic, it fell from almost \$8,000 to \$5,000 in a single day in March 2020. Yet by November of 2020, it had risen to about \$20,000. After reaching a high of about \$64,000 in the middle of April 2021, Bitcoin quickly lost over half its value due to regulatory pressure from China's crackdown on the industry. For a while it hovered around $40,000, then in the middle of 2022 it dropped to about $16,000 before rebounding in early 2023.

3.1.3 Feature Engineering

A range of technical indicators has been mathematically derived from the historical price data to develop the quality of the data and provide insights into potential market trends and price movements. These selected technical indicators, as outlined in Table 3.2, play a crucial role in enhancing the predictive power of the models. Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Upper Bollinger band and Lower Bollinger band are the indicators which are used as additional features to train the system.

Correlation heatmap in Figure 3.5 has been created to illustrate the relationships among these technical indicators. The heatmap visually presents how strongly these indicators are correlated with each other. This allows us to identify potential patterns and dependencies between the indicators.

Feature	Description
Open	The opening price of BTC at the beginning of the trading period.
High	The highest price reached by BTC during the trading period.
Low	The lowest price reached by BTC during the trading period.
Close	The closing price of BTC at the end of the trading period.
Volume	The total number of units of BTC traded during the trading period.
Return	The return of BTC, calculated as the percentage change in its price from the previous trading period.
SMA220	Simple Moving Average over 220 days
SMA48	Simple Moving Average over 48 days
SMA26	Simple Moving Average over 26 days
SMA20	Simple Moving Average over 20 days
SMA12	Simple Moving Average over 12 days
EMA12	Exponential Moving Average with a 12-day smoothing factor
EMA26	Exponential Moving Average with a 26-day smoothing factor
EMA48	Exponential Moving Average with a 48-day smoothing factor
EMA220	Exponential Moving Average with a 220-day smoothing factor
Bollinger_Upper	A volatility-based indicator calculated from the standard deviation of the price.
Bollinger_Lower	represents potential oversold conditions when prices move below this band.
RSI	Relative Strength Index, a momentum oscillator that measures the speed and change of price movements.

MACD	Moving Average Convergence Divergence, shows the relationship between two moving averages of the price, helping to identify potential changes in trend direction.

Table 3.2. Technical Indicators Description

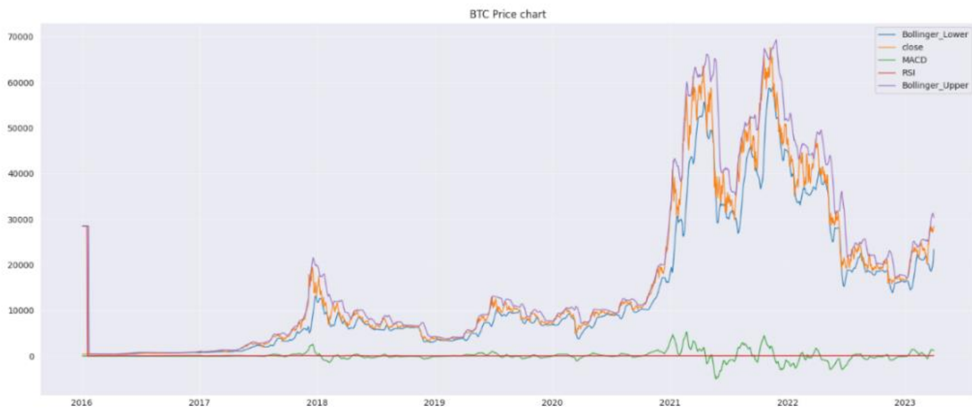


Figure 3.4. BTC price chart with Technical Indicators

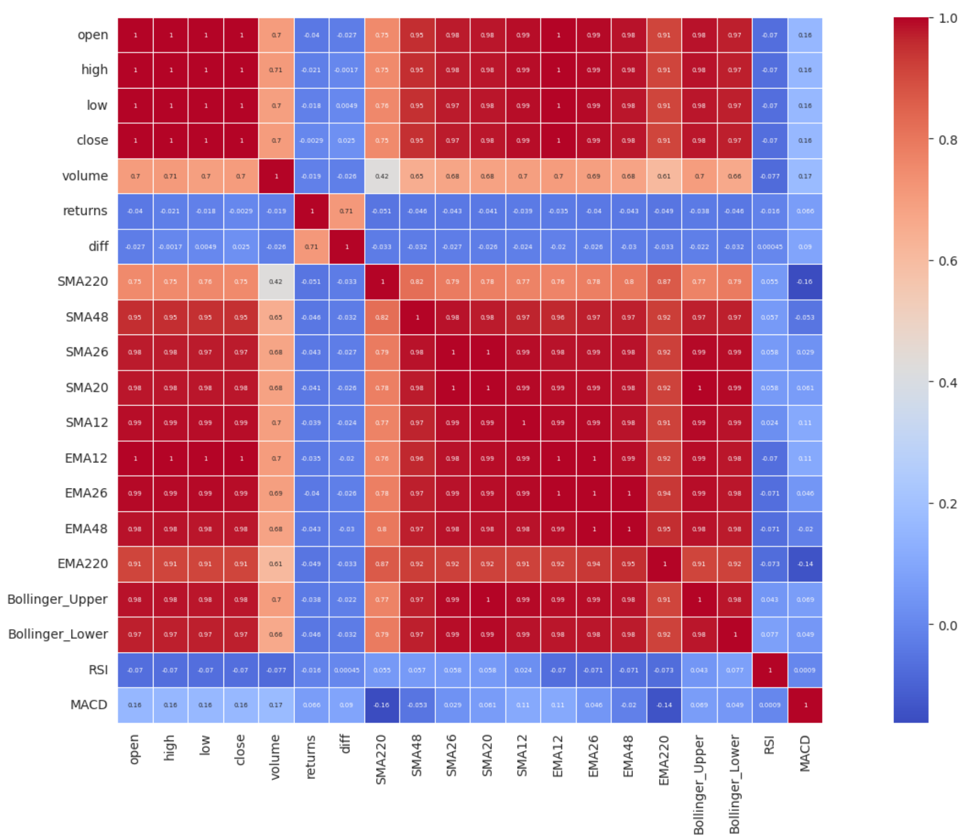


Figure 3.5. Correlation Heatmap of Technical Indicators with BTC price

3.2 Statistical Tests and Data Transformation

Before implementing ARIMA model, the ADF test was performed on the data to confirm that the input data is stationary. [34] The calculated p-value of 0.456 indicates that we cannot reject the null hypothesis, and therefore, the dataset is non-stationary. The first order differencing method was used for transforming data to stationery. An ADF test with a p-value of 9.933399758673523e-14 was also performed to confirm the stationarity of the data at a confidence level of 0.05. The Ljung-Box statistic test was also run to test for white noise.[35] For lags greater than 9, the results showed a p-value less than the threshold for statistical significance (0.000034< 0.05). Thus, the series is not a white-noise process.

3.3 Dataset Partitioning

To gain insights into how models perform under different market conditions, the dataset was divided into three unique time intervals:

Period 1: Spanning from '2016-01-01' to '2019-08-31'.

Period 2: containing the timeline from '2019-09-01' to '2023-03-31'.

Period 3: Spanning the entire dataset from '2016-01-01' to '2023-03-31'.

The selection of these periods was intentional to show different market behaviors. The first period represents a phase of stable market conditions with lower volatility, while the second period indicates a phase characterized by high market volatility. In addition, the third phase allows for an examination of the performance of the models throughout the entire study timeframe.

Figure 3.6. Partitioned dataset

3.4 ARIMA Model Implementation

The input data for the ARIMA model has been divided into three sets: an 80% training set, a 10% validation set, and a 10% test set. This approach is also utilized throughout the study to ensure robust model evaluation.

Figure 3.7. Training, Validation and Test dataset for Period1

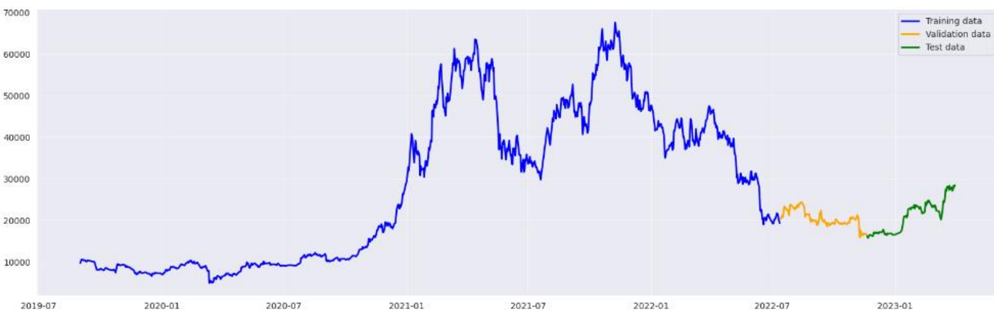


Figure 3.8. Training, Validation and Test dataset for Period2

Figure 3.9. Training, Validation and Test dataset for Period3

Autocorrelation was computed at various lags to examine the potential presence of temporal patterns within the data. Among these lags, the highest correlation was observed at lags 7, 8, and 9, indicating a potential weekly pattern within the data. The autocorrelation function (ACF) and partial autocorrelation function (PACF) plots in Figure 3.10 and Figure 3.11 visually illustrate that these lags could be good candidates for the optimal p and q values.

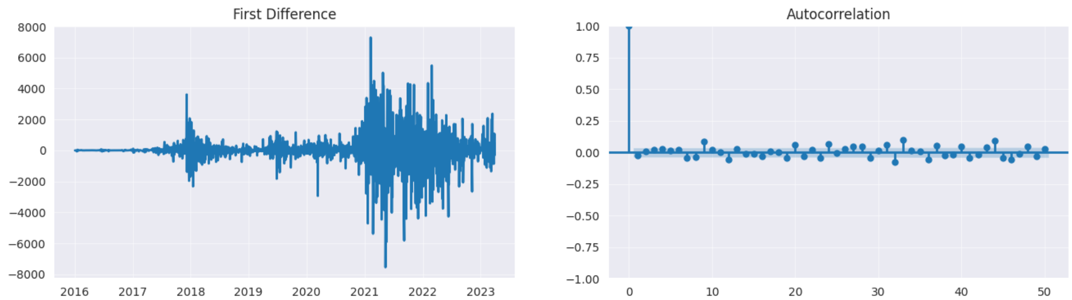


Figure 3.10. First difference and ACF plot

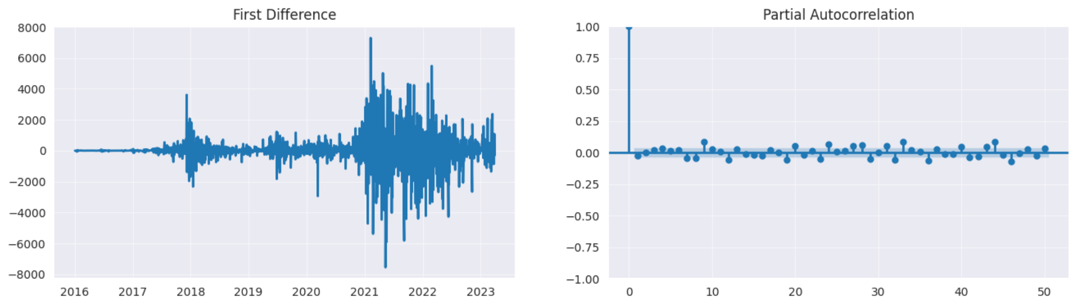


Figure 3.11. First difference and PACF plot

For each period, the ARIMA model was built using the first-order differencing (d=1) to achieve stationarity. An exploration of the model was conducted by testing a range of p and q values from 0 to 10. The objective was to identify a model configuration that would result in a lower Akaike Information Criterion (AIC) score. The AIC score helps balance how well the model fits the data with its complexity, helping us choose the best model. The evaluation results for the best performing ARIMA model within each period are presented below:

Time Frame	**Top 3 Model Parameters**	**AIC**
	(9,1,7)	15268.992
Period 1	(9,1,8)	15271.964
	(7,1,9)	15272.409
	(9, 1, 6)	17909.306
Period 2	(9, 1, 7)	17910.008
	(7, 1, 9)	17910.700
	(8,1,8)	34100.544
Period 3	(8,1,7)	34103.528
	(7,1,8)	34103.844

Table 3.3. Top three fitted models for each period

Time Frame	Best Model	Log Likelihood	AIC	BIC	Ljung-Box (L1) (Q)	Jarque-Bera (JB)
Period 1	(9, 1, 7)	-7617.496	15268.992	15353.542	0.00	19250.29
Period 2	(9, 1, 6)	-8938.653	17909.306	17988.535	0.18	948.02
Period 3	(8, 1, 8)	-17033.272	34100.544	34196.710	0.01	26871.21

Table 3.4. Diagnostic tests for each period

In addition to model configuration and AIC score, the analysis of residuals provides valuable insights into the model's performance. Residuals are the differences between the observed values and the predictions made by the ARIMA model. A thorough examination of residuals including residual plots and diagnostics, will be discussed in the next chapter alongside the predictions obtained from the ARIMA model.

3.5 LSTM Model Implementation

The implementation strategy was guided by the observed autocorrelation plots, which indicated a recurring weekly pattern within the data. Therefore, all aspects of model implementation and evaluations were aligned with a 7-day lag and one-step prediction approach.

3.5.1 Architecture and Hyperparameters

When building LSTM models, three individual model types tested as follows:

Model Type	Hidden Layers	Drop-Out layer	Activation Function	Kernel Regulizer	Optimizer	Learning Rate	Loss Function	Batch size
Simple LSTM	0	None	None	None	Adam	0.001	MSE	32
Stacked LSTM 1	1	P=0.2	ReLU	L2(0.001)	Adam	0.001	MSE	32
Stacked LSTM 2	2	P=0.2	ReLU	L2(0.001)	Adam	0.001	MSE	32

Table 3.5. LSTM models Architecture and Hyperparameters

Another crucial hyperparameter is the number of epochs which represents the number of passes required to learn the training data. Larger epoch values often result in more accurate predictions. However, excessively high epoch values can lead to overfitting issues, where the model fits noise in the training data rather than the underlying patterns. Therefore, the number of epochs was individually adjusted for each model type and each period, taking into consideration the loss function results and the potential for overfitting.

3.5.2 Univariate and Multivariate Forecasting

All three variations of LSTM models were implemented and trained for both univariate (using close price) and multivariate forecasting. This was achieved by using the calculated features to capture diverse market behaviors. This approach was applied iteratively for each of the time periods. The following sets of features were used for multivariate forecasting:

1. Close, Returns
2. Close, Exponential Moving Averages (EMA12, EMA26, EMA48, EMA220)
3. Close, Simple Moving Averages (SMA12, SMA26, SMA20, SMA48, SMA220)
4. Close, Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD)
5. Close, Bollinger Bands (Upper and Lower)
6. Close, Exponential Moving Averages (EMA12, EMA26, EMA48, EMA220), Simple Moving Averages (SMA12, SMA26, SMA20, SMA48, SMA220), Bollinger Bands (Upper and Lower)
7. Close, Exponential Moving Averages (EMA12, EMA26, EMA48, EMA220), Simple Moving Averages (SMA12, SMA26, SMA20, SMA48, SMA220), Bollinger Bands (Upper and Lower), RSI, MACD

3.6 Hybrid ARIMA-LSTM Implementation

The parallel approach was chosen due to its ability to generate more accurate predictions for bitcoin prices. For each period, LSTM models were implemented for both univariate and multivariate forecasting. The best-performing model was selected for each step and combined with the ARIMA results to determine the values of intercept, coefficient for LSTM forecasts and coefficient for ARIMA forecasts. This process aimed to enhance the final prediction results through a passive integration of the chosen models.

3.7 Evaluation Metrics

To assess the performance of the implemented models, several evaluation metrics were employed with the equations defined as follows:

Root Mean Squared Error (RMSE): quantifies the average magnitude of the prediction errors. It measures the square root of the average of squared differences between predicted values and actual values.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \widehat{Y_i})^2} \tag{3.1}$$

Mean Squared Error (MSE): MSE computes the average of squared differences between predicted values and actual values, providing a measure of the overall prediction error.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \widehat{Y_i})^2 \tag{3.2}$$

Mean Absolute Error (MAE): MAE calculates the average absolute differences between predicted values and actual values:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left| Y_i - \widehat{Y_i}\right| \tag{3.3}$$

Mean Absolute Percentage Error (MAPE): MAPE computes the average percentage difference between predicted values and actual values:

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{Y_i - \widehat{Y_i}}{Y_i}\right| \tag{3.4}$$

Coefficient of Determination (R2 score): It measures the proportion of the variance in the dependent variable that is explained by the independent variables. It indicates the goodness of fit of the model to the data:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{3.5}$$

Where Y_i is the actual value, $\widehat{Y_i}$ is the predicted value, RSS is the sum of squares of residuals and TSS is the total sum of squares.

Chapter 4: Experimental Results

4.1 ARIMA Results

In this section, the results of the ARIMA models are presented and analyzed. The focus is on examining the residuals, diagnostic tests, predictions, and evaluation metrics for each period.

4.1.1 Residual Analysis

In Figure 4.1. Diagnostics of Residuals in Period1, it is evident that the residuals demonstrate no correlation and lack evident seasonal patterns. Furthermore, the residuals exhibit a normal distribution centered around zero mean. The QQ-plot illustrates that the sequence of residuals aligns closely with the linear trend of samples extracted from a standard normal distribution (N(0, 1)). This alignment provides strong evidence supporting the hypothesis that the residuals adhere to a normal distribution. The same observations and conclusions hold true for the residuals of periods 2 and 3, as illustrated in Figure 4.2.

The Ljung-Box test was employed to assess autocorrelation within the residuals. The p-value of 0.980402 substantiates the lack of significant evidence to reject the null hypothesis (residuals are distributed independently), thus confirming that the residuals are white noise. The same confirmation holds true for periods 2 and 3 as well.

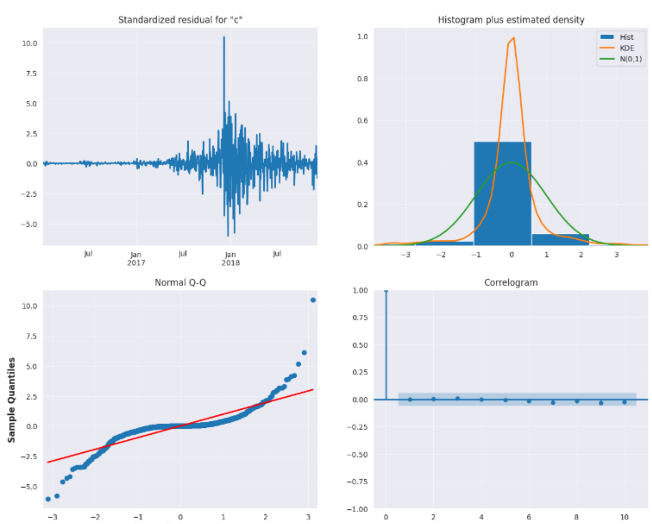


Figure 4.1. Diagnostics of Residuals in Period1

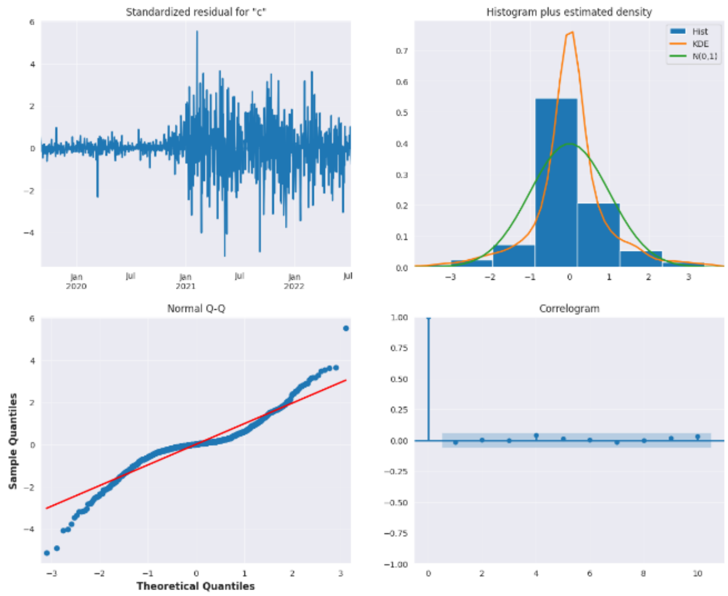


Figure 4.2. Diagnostics of Residuals in Period2

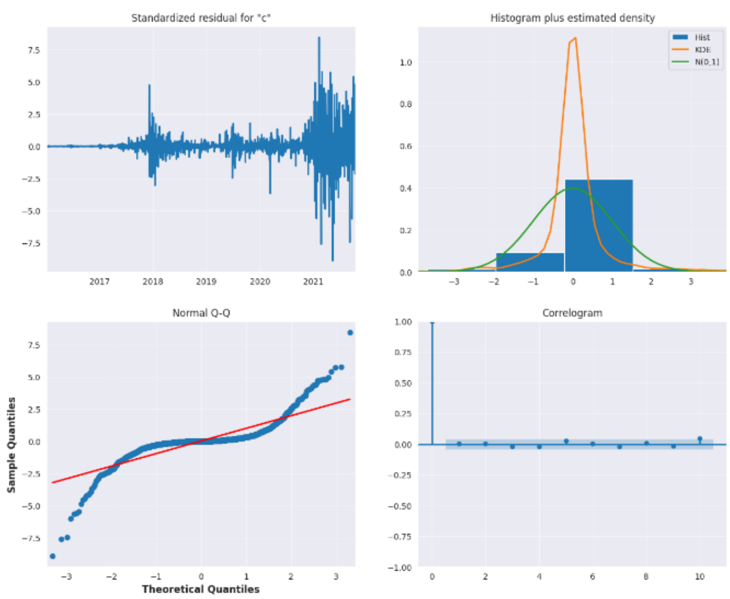


Figure 4.3. Diagnostics of Residuals in Period3

4.1.2 ARIMA Predictions and Evaluations

The ARIMA model's predictions are presented through both in-sample and out-of-sample forecasts for each period. The model's performance is assessed on both the validation and test datasets to provide a comprehensive evaluation of its accuracy.

4.1.2.7 In-Sample forecast results:

The following plots illustrate predictions within validation data for each period:

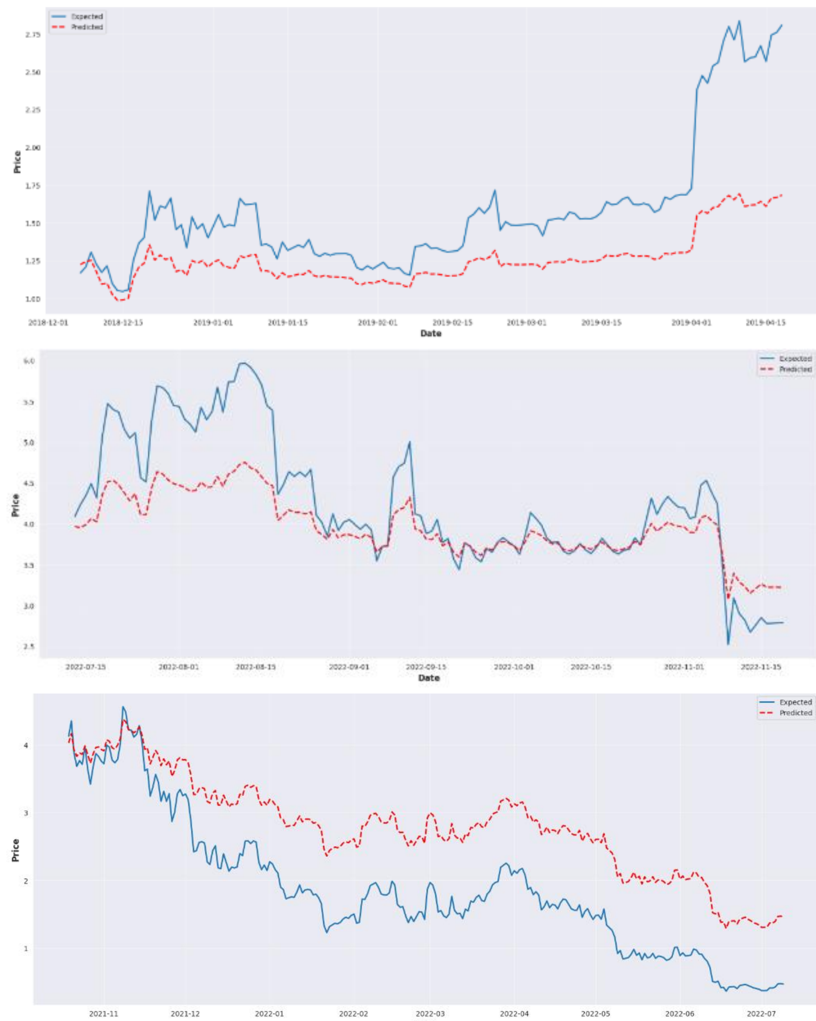


Figure 4.4. ARIMA I.S predictions for Period 1(top), Period2 (middle) and Period 3 (bottom)

4.1.2.8 Out-of-Sample forecast results:

The following plots in Figure 4.5 showcase each model’s ability to capture the underlying trends within test data for each period.

Figure 4.5. ARIMA O.S predictions for Period 1(top), Period2 (middle) and Period 3 (bottom)

The performance of the ARIMA model's predictions across all the time frames is summarized in the table below:

Metric	Period	In-Sample	Out-of-Sample
MSE	Period 1	842136.13	39790958.50
	Period 2	5368029.55	17159955.54
	Period 3	802058479.20	2217567316.88
RMSE	Period 1	917.68	6308.01
	Period 2	2316.90	4142.46
	Period 3	28320.64	47091.05
MAE	Period 1	766.36	5962.51
	Period 2	1735.80	3615.06
	Period 3	25389.94	46993.73
MAPE	Period 1	18.28	63.02%
	Period 2	8.11	16.53%
	Period 3	77.26	233.61%

Table 4.1. Evaluation metrics for each period

4.2 LSTM Model Results

Table 4.2. presents a comprehensive analysis of the LSTM models' performance across different feature combinations and time periods. Its noteworthy to mention that significant differences in MAPE observed among models in Period 3 highlights the difficulty of accurately predicting bitcoin prices due to heightened volatility during this phase.

Features	Period	Model	RMSE	MSE	MAE	MAPE	R2
Close	1	Vanilla	486.71	236882.64	348.60	3.60%	0.94
	2	Vanilla	599.68	359613.40	410.13	1.85%	0.98
	3	Stacked 1	669.66	448448.03	468.61	2.23%	0.95
Close, Returns	1	Vanilla	311.86	97259.64	168.18	4.71%	0.24
	2	Vanilla	422.57	178568.53	222.36	1.22%	0.36
	3	Stacked 1	476.69	227240.08	255.75	5.77%	0.41
Close, EMA12, EMA26, EMA48, EMA220	1	Vanilla	544.90	296917.83	390.80	4.96%	0.85
	2	Vanilla	1646.76	2711828.28	1329.48	6.31%	-0.45
	3	Vanilla	2281.26	5204161.95	1838.70	8.84%	0.29
Close, SMA12, SMA26, SMA20, SMA48, SMA220	1	Stacked 1	543.39	295272.43	410.08	4.93%	0.88
	2	Stacked 1	961.48	924458.53	702.048	3.48%	0.51
	3	Vanilla	2706.58	7325567.71	2284.35	11.04%	-0.16
Close, RSI, MACD	1	Vanilla	380.75	144968.09	169.78	38.48%	0.54
	2	Vanilla	386.77	149589.06	206.21	73.75%	0.12
	3	Vanilla	557.15	310418.95	310.14	103.40 %	-0.12
Close, Upper Bollinger Band, Lower Bollinger Band	1	Vanilla	410.59	168584.89	282.95	3.01%	0.96
	2	Vanilla	442.81	196079.93	289.67	1.37%	0.99
	3	Stacked 2	591.57	349961.04	439.79	2.13%	0.96
Close, EMA12, EMA26, EMA48, EMA220, SMA12, SMA26, SMA20, SMA48, SMA220, Upper and Lower Bands	1	Stacked 1	603.60	364341.09	467.20	5.98%	0.84
	2	Vanilla	1961.86	3848892.27	1236.54	6.04%	-2.08
	3	Stacked 1	2111.59	4458837.60	1622.53	7.29%	0.46

Close, EMA12, EMA26, EMA48, EMA220, SMA12, SMA26, SMA20, SMA48, SMA220, Upper Bollinger Band, Lower Bollinger Band, RSI, MACD	1	Stacked 1	762.22	580991.33	552.129	90.15%	-19.38
	2	Stacked 1	1858.12	3452617.50	1419.33	50.88%	-2.03
	3	Stacked 1	2078.96	4322099.06	1618.80	152.7%	-24.43

Table 4.2. Performance of LSTM models in each Time frame

4.2.1 Best-Performed model analysis

Overall, the feature combination of Close, Upper Bollinger Band, and Lower Bollinger Band consistently showed the most favorable outcomes across the evaluated features. Following this, the Close price itself and the feature combinations of Close, Return, and Close, RSI, MACD also exhibited good performance. However, after careful consideration, **Close, Upper Bollinger Band, and Lower Bollinger Band** selected as the optimal feature selection for the LSTM model across all timeframes.

For Period 1, Figure 4.6 indicates the comparison of MAE and loss values and between Validation and Training data during the training process, showing that the model perfectly fitted the training data. In addition, Bitcoin's Close Price Prediction in all periods are plotted in Figure 4.7, Figure 4.8 and Figure 4.9 respectively:

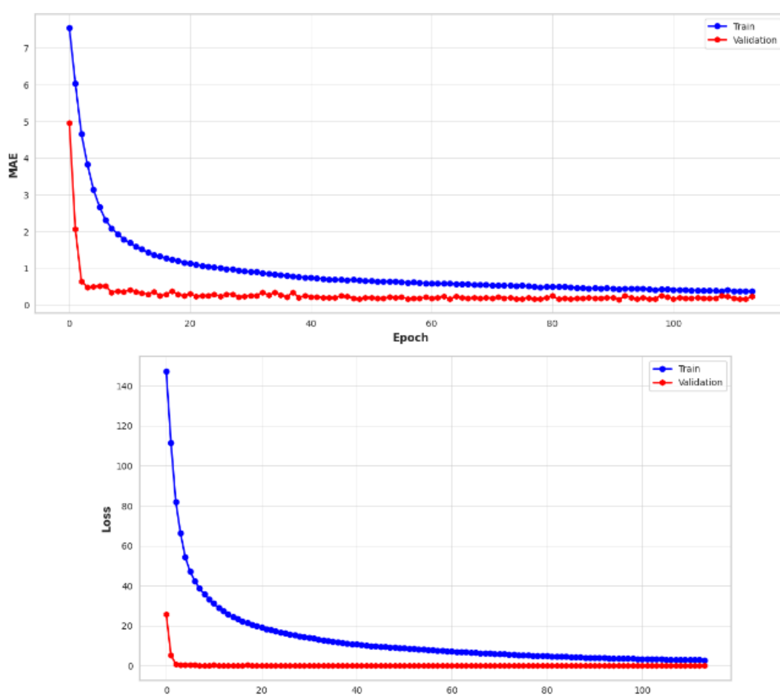


Figure 4.6. MAE comparison (Top) and Loss comparison (Bottom) between Validation and Train Data in Period1

Figure 4.7. Bitcoin Close price prediction Versus Test data in Period1

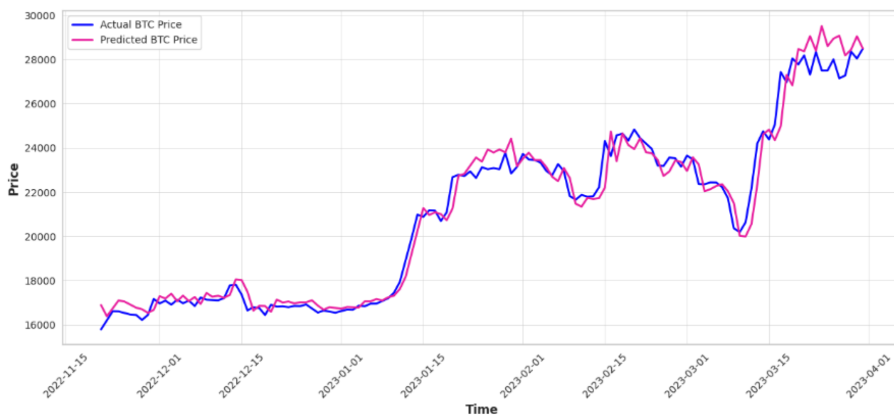


Figure 4.8. Bitcoin Close price prediction Versus Test data using Vanilla LSTM model for Period2.

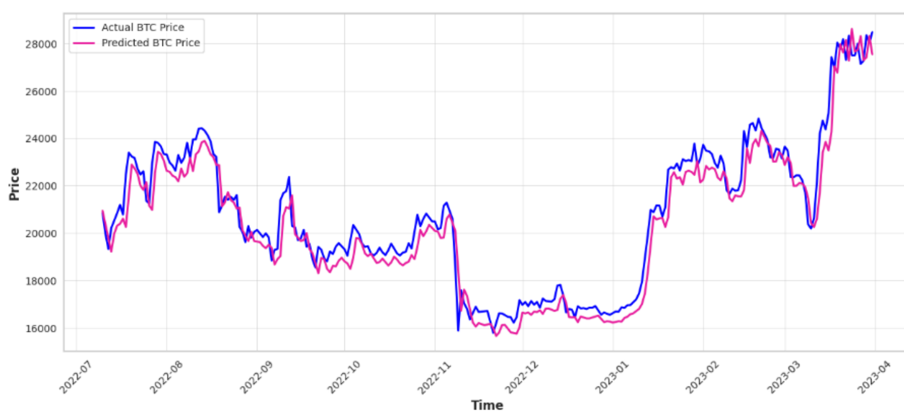


Figure 4.9. Bitcoin Close price prediction Versus Test data using Stacked LSTM (2 hidden layer) for Period3

4.3 LSTM on Differenced Data

For a comprehensive comparison of LSTM model performance, experiments were also tested on differenced data instead of original close price data on all time periods.

Figure 4.10. Predicted values and their reverse transformations into original prices in Period1

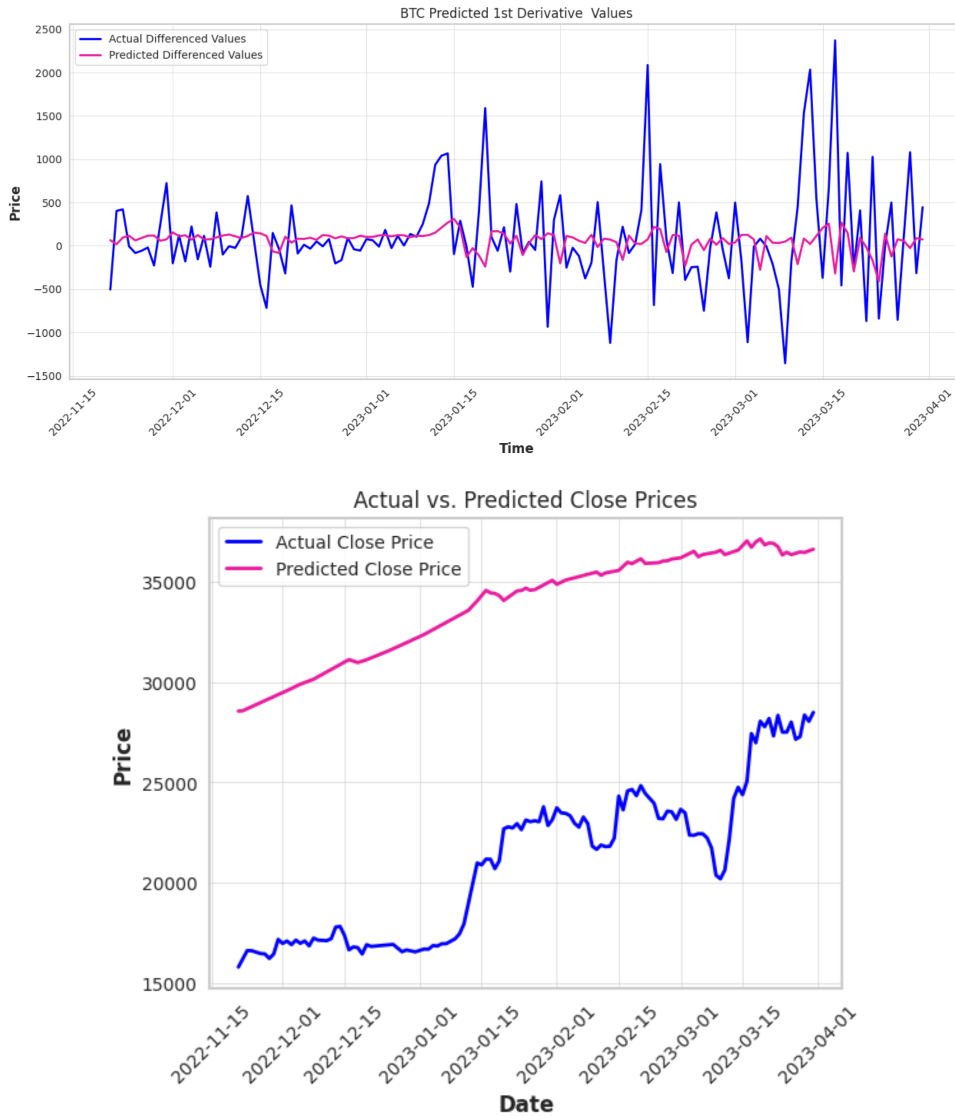


Figure 4.11. Predicted values and their reverse transformations into original prices in Period2

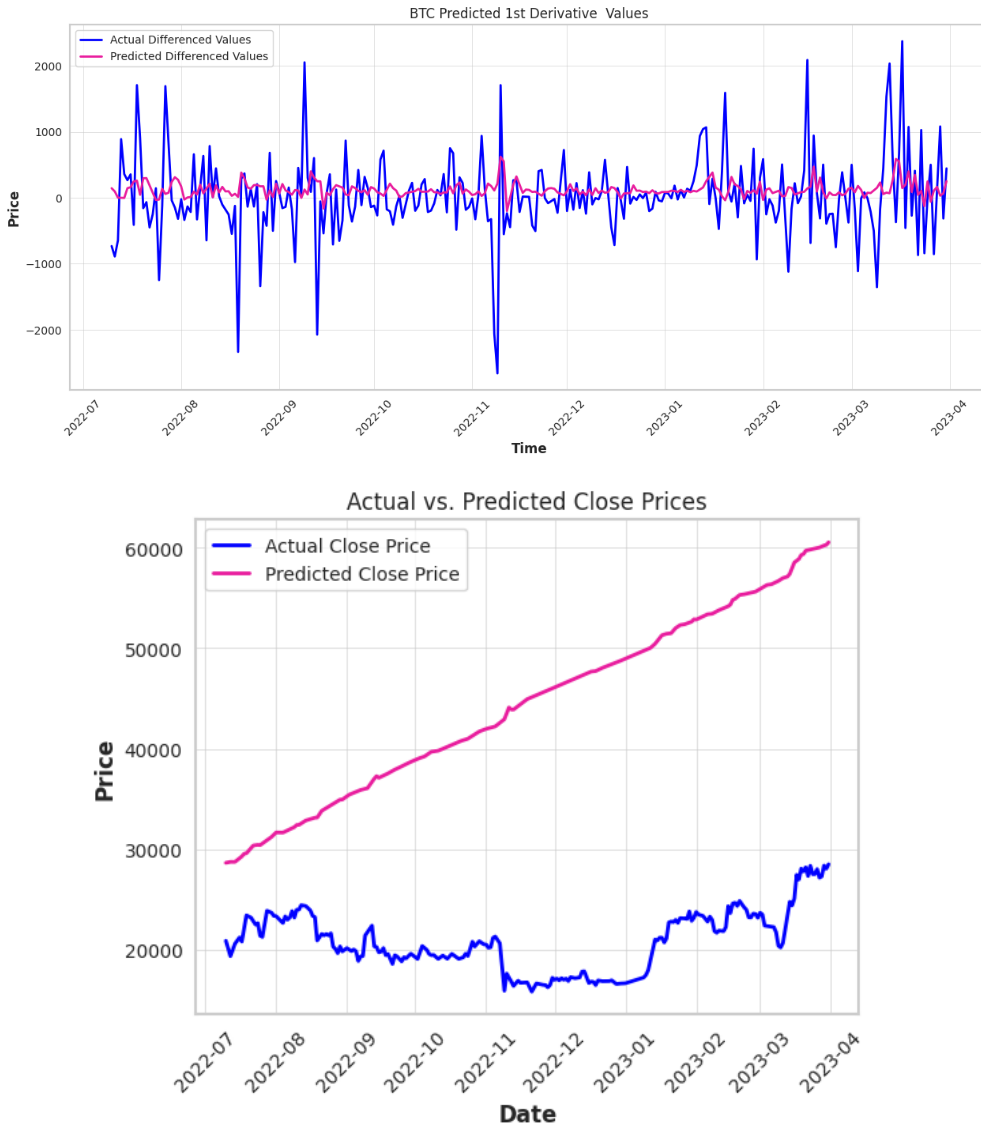


Figure 4.12. Predicted values and their reverse transformations into original prices in Period3

Period	RMSE	MSE	MAE	MAPE	R2
1	620.95	385579.18	382.75	198.20%	-0.84
2	602.06	362479.77	410.27	236.70%	-0.09
3	625.44	391173.09	425.13	165.64%	-0.03

Table 4.3. First differenced data prediction evaluations

Results and figures clearly demonstrate that LSTM struggled to make accurate predictions when applied to differenced data. This can be attributed to several factors. LSTM models are sensitive to the characteristics of the data they are trained on, and predicting close prices directly allows the model to learn the patterns and dependencies in the original price series. However, when differencing the data, some of these patterns are removed, making the prediction more challenging for the model. Differencing also transforms the data into a stationary format with different statistical properties, impacting the model's ability to capture meaningful patterns.

4.4 LSTM on Random Walk Data

To explore the predictive power of the LSTM model more thoroughly, another experiment was carried out. This time on a generated random walk data. In contrast to Bitcoin price data, true random walks are characterized by unpredictable behavior, thus providing a unique challenge for predictive modeling. Figure 4.13 demonstrates the generated random walk. The prediction values of the random walk vs the test data are shown in Figure 4.14.

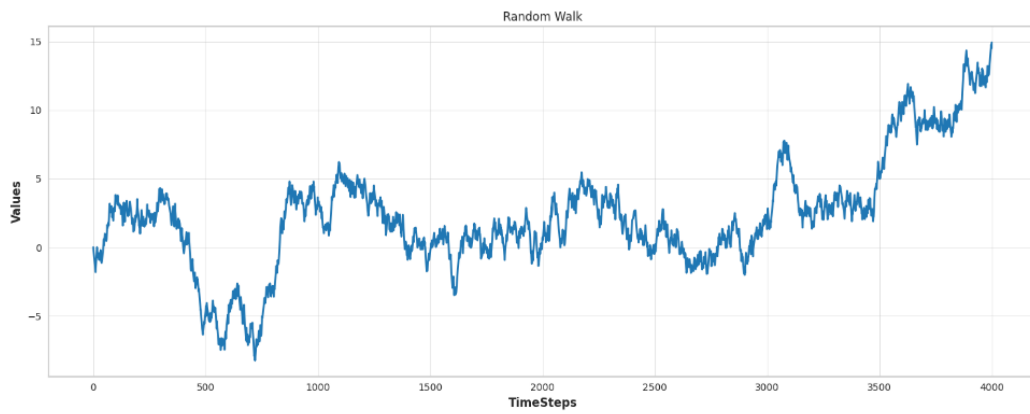
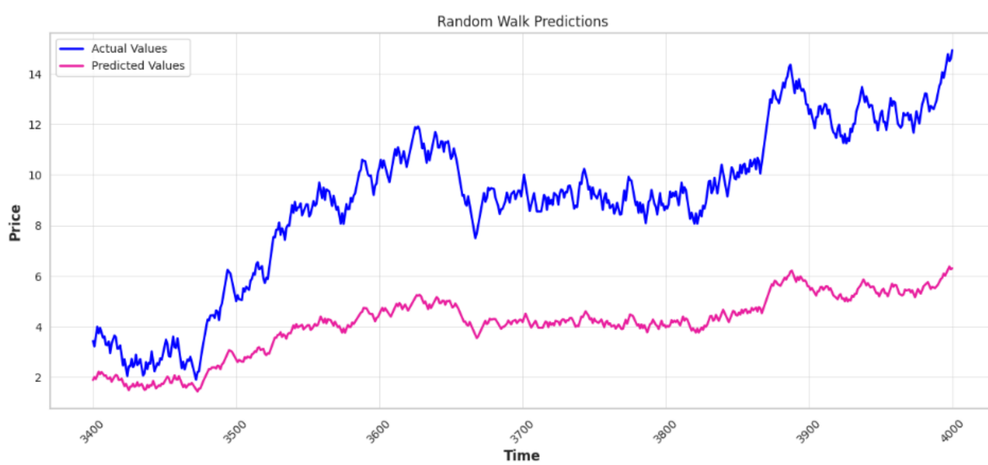


Figure 4.13. Generated Random walk data



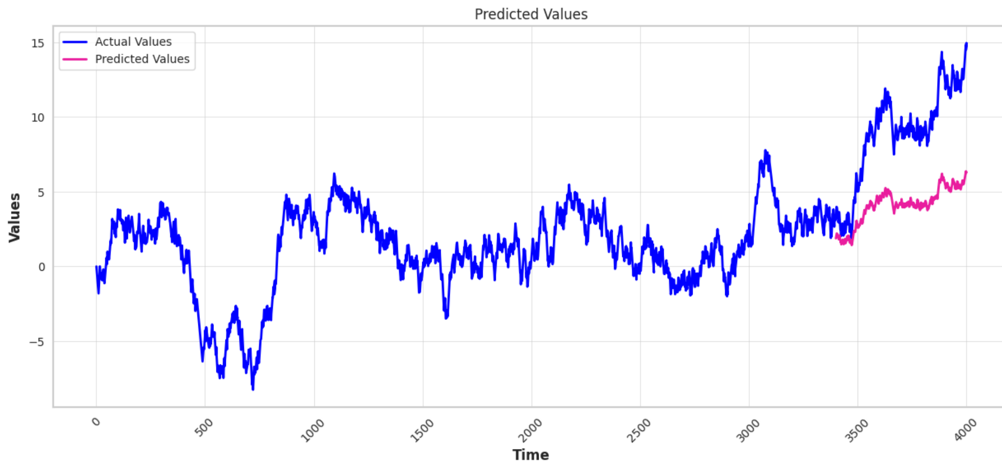


Figure 4.14. Random walk prediction with LSTM

The evaluation metrics for this experiment are as follows:

RMSE	MSE	MAE	MAPE	R2
5.23	27.33	4.86	52.18%	-1.82

Table 4.4. Evaluation metrics of Random walk prediction with LSTM

The difference in LSTM's performance comes down to its expertise in spotting patterns in data. In bitcoin dataset with complex time-based relationships, LSTM does exceptionally well. However, when faced with pure randomness, it struggles to predict the data that lacks structure, leading to less accurate predictions.

4.5 Hybrid Model Results

The hybrid model's performance is captured in the following plots and metrics:

For the **Close, Upper Bollinger Band, and Lower Bollinger Band** features, the hybrid model generally performs slightly worse than the Vanilla LSTM in terms of RMSE, MSE, MAE, and MAPE across all three periods.

Period	Model	RMSE	MSE	MAE	MAPE	R2
1	Vanilla	410.59	168584.89	282.95	3.01%	0.96
	Hybrid	550.99	303590.29	391.04	4.08%	0.93
2	Vanilla	442.81	196079.93	289.67	1.37%	0.99
	Hybrid	661.86	438062.10	462.81	2.13%	0.97
3	Stacked 2	591.57	349961.04	439.79	2.13%	0.96
	Hybrid	644.84	415814.54	420.89	2.00%	0.95

Table 4.5. Hybrid model results with Close, Upper Bollinger Band, and Lower Bollinger Band features

In the case of **Close, Returns**, the hybrid model exhibits better MAPE values compared to LSTM models. R2 scores remain promising, indicating its ability to explain the variance in the target variable, albeit slightly less accurately in terms of RMSE than the LSTM.

Period	Model	RMSE	MSE	MAE	MAPE	R2
1	Vanilla	311.86	97259.64	168.18	4.71%	0.24
	Hybrid	435.11	189318.94	311.81	3.29%	0.95
2	Vanilla	422.57	178568.53	222.36	1.22%	0.36
	Hybrid	592.24	350746.10	413.30	1.89%	0.98
3	Stacked 1	476.69	227240.08	255.75	5.77%	0.41
	Hybrid	626.88	392981.91	417.34	1.98%	0.96

Table 4.6. Hybrid model results with Close and Returns features.

For **Close** alone, the hybrid model's performance is slightly better than LSTM. It maintains competitive RMSE, MSE, MAE, and MAPE values while showing strong R2 scores.

Period	Model	RMSE	MSE	MAE	MAPE	R2
1	Vanilla	486.71	236882.64	348.60	3.60%	0.94
	Hybrid	466.03	217182.07	334.10	3.47%	0.95
2	Vanilla	599.68	359613.40	410.13	1.85%	0.98
	Hybrid	594.76	353737.09	417.89	1.91%	0.98
3	Stacked 1	669.66	448448.03	468.61	2.23%	0.95
	Hybrid	617.79	381663.86	411.25	1.95%	0.96

Table 4.7. Hybrid model results with Close feature

Chapter 5: Conclusion and Future Work

In this study, we provided a comprehensive exploration of Bitcoin price prediction using different models, feature combinations, and timeframes. Our investigation aimed to uncover the strengths and limitations of ARIMA, LSTM, and Hybrid ARIMA-LSTM models. Throughout our study, we incorporated an array of technical indicators. This strategic selection of features aimed to capture the behavior of Bitcoin price movements. We evaluated each feature's impact on the model's performance. We also observed that the performance of models varied across different time periods, reflecting the changing nature of the Bitcoin market.

The results obtained from our ARIMA models exhibited better predictive capabilities, for the period with lower volatility. The LSTM models, on the other hand showcased their ability in capturing complex patterns within the data. Their predictive accuracy was evident, especially in timeframes with higher volatility. We identified specific feature combinations that yielded optimal results for each period, underscoring the importance of feature engineering in adapting models to changing market conditions. As a pinnacle of our exploration, the Hybrid ARIMA-LSTM model emerged as a fusion of the strengths of both ARIMA and LSTM. This approach offered a novel approach of combining time series analysis and deep learning, using their abilities to enhance accuracy. The results obtained from the hybrid model indicated its potential as a powerful tool for forecasting cryptocurrency prices, bridging the gap between traditional and modern prediction methodologies.

Additionally, we assessed LSTM model on differenced data and true random walks. This comparison highlighted how their performance varies with data characteristics. LSTM performed well in Bitcoin price series analysis due to identifiable patterns. Yet, It struggled when dealing with pure randomness, emphasizing the significance of data characteristics in predictive modeling.

For the future research, there are new paths to explore such as the combination of sentiment analysis with social media data. This approach could give us a better understanding of how people's feelings affect price changes. Furthermore, adding external factors, like economic indicators, also holds a lot of promise for making predictions more accurate. Lastly, we might explore forecasting trends over longer periods. This isn't just about helping us make smart investments—it's about understanding how different factors play out over time in the market.

Bibliography

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”.

[2] “Cryptocurrency Prices, Charts And Market Capitalizations,” *CoinMarketCap*. https://coinmarketcap.com/ (accessed Apr. 07, 2023).

[3] M. Mudassir, S. Bennbaia, D. Unal, and M. Hammoudeh, “Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach,” *Neural Comput. Appl.*, Jul. 2020, doi: 10.1007/s00521-020-05129-6.

[4] N. Latif, J. Selvam, M. Kapse, V. Sharma, and V. Mahajan, “Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices,” *Australas. Account. Bus. Finance J.*, vol. 17, no. 1, pp. 256–276, Jan. 2023, doi: 10.14453/aabfj.v17i1.15.

[5] S. M. Raju and A. M. Tarif, “Real-Time Prediction of BITCOIN Price using Machine Learning Techniques and Public Sentiment Analysis.” arXiv, Jun. 18, 2020. doi: 10.48550/arXiv.2006.14473.

[6] Z. Chen, C. Li, and W. Sun, “Bitcoin price prediction using machine learning: An approach to sample dimension engineering,” *J. Comput. Appl. Math.*, vol. 365, p. 112395, Feb. 2020, doi: 10.1016/j.cam.2019.112395.

[7] Y. Jin *et al.*, “Prediction of COVID-19 Data Using an ARIMA-LSTM Hybrid Forecast Model,” *Mathematics*, vol. 10, no. 21, Art. no. 21, Jan. 2022, doi: 10.3390/math10214001.

[8] “Additive and multiplicative seasonality – can you identify them correctly? – Nikolaos Kourentzes,” Apr. 24, 2017. https://kourentzes.com/forecasting/2014/11/09/additive-and-multiplicative-seasonality/ (accessed Apr. 07, 2023).

[9] Douglas C. Montgomery, Cheryl L. Jennings, and Murat Kulahci, “Introduction to Time Series Analysis and Forecasting”.

[10] Howell Tong, *Non-Linear Time Series: A Dynamical System Approach*. Oxford University Press (UK) (August 5, 1993).

[11] Chris Chatfield, *The Analysis of Time Series: An Introduction*, Sixth Edition.

[12] *2.3 Time series patterns | Forecasting: Principles and Practice (2nd ed)*. Accessed: Dec. 18, 2022. [Online]. Available: https://otexts.com/fpp2/tspatterns.html

[13] S. Nadarajah and J. Chu, “On the inefficiency of Bitcoin,” *Econ. Lett.*, vol. 150, pp. 6–9, Jan. 2017, doi: 10.1016/j.econlet.2016.10.033.

[14] S. Palamalai, K. K. Kumar, and B. Maity, “Testing the random walk hypothesis for leading cryptocurrencies,” *Borsa Istanb. Rev.*, vol. 21, no. 3, pp. 256–268, Sep. 2021, doi: 10.1016/j.bir.2020.10.006.

[15] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*. in Springer Texts in Statistics. New York, NY: Springer New York, 2011. doi: 10.1007/978-1-4419-7865-3.
[16] G. S. Maddala and S. Wu, “A Comparative Study of Unit Root Tests with Panel Data and a New Simple Test,” *Oxf. Bull. Econ. Stat.*, vol. 61, no. S1, pp. 631–652, 1999, doi: 10.1111/1468-0084.0610s1631.
[17] H. Lütkepohl and F. Xu, “The role of the log transformation in forecasting economic variables,” *Empir. Econ.*, vol. 42, no. 3, pp. 619–638, Jun. 2012, doi: 10.1007/s00181-010-0440-1.
[18] D. McDowall, R. McCleary, and B. J. Bartos, *Interrupted Time Series Analysis*. Oxford University Press, 2019.
[19] V. Kotu and B. Deshpande, “Chapter 2 - Data Science Process,” in *Data Science (Second Edition)*, V. Kotu and B. Deshpande, Eds., Morgan Kaufmann, 2019, pp. 19–37. doi: 10.1016/B978-0-12-814761-0.00002-2.
[20] P. Katsiampa, “Volatility estimation for Bitcoin: A comparison of GARCH models,” *Econ. Lett.*, vol. 158, pp. 3–6, Sep. 2017, doi: 10.1016/j.econlet.2017.06.023.
[21] L. Kristoufek, “What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis,” *PLOS ONE*, vol. 10, no. 4, p. e0123923, Apr. 2015, doi: 10.1371/journal.pone.0123923.
[22] M. Benzekrı and H. Özütler, “On the Predictability of Bitcoin Price Movements: A Short-term Price Prediction with ARIMA,” *J. Econ. Policy Res. İktisat Polit. Araştırmaları Derg.*, vol. 8, no. 2, pp. 293–309, Jul. 2021, doi: 10.26650/JEPR.946081.
[23] *8.7 ARIMA modelling in R | Forecasting: Principles and Practice (2nd ed)*. Accessed: Feb. 17, 2023. [Online]. Available: https://otexts.com/fpp2/arima-r.html
[24] L. Felizardo, R. Oliveira, E. Del-Moral-Hernandez, and F. Cozman, “Comparative study of Bitcoin price prediction using WaveNets, Recurrent Neural Networks and other Machine Learning Methods,” in *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, Oct. 2019, pp. 1–6. doi: 10.1109/BESC48373.2019.8963009.
[25] P. T. Yamak, L. Yujian, and P. K. Gadosey, “A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting,” in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, in ACAI ’19. New York, NY, USA: Association for Computing Machinery, Feb. 2020, pp. 49–55. doi: 10.1145/3377713.3377722.
[26] C.-H. Wu, C.-C. Lu, Y.-F. Ma, and R.-S. Lu, “A New Forecasting Framework for Bitcoin Price with LSTM,” in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 168–175. doi: 10.1109/ICDMW.2018.00032.
[27] F. Es-sabery, A. Hair, J. Qadir, B. Sainz de Abajo, B. Garcia-Zapirain, and I. De la Torre Díez, “Sentence-Level Classification Using Parallel Fuzzy Deep Learning

Classifier," *IEEE Access*, vol. PP, pp. 1–1, Jan. 2021, doi: 10.1109/ACCESS.2021.3053917.
[28] "Percipio Book Deep Learning with Python, Second Edition." https://cgi.percipio.com/books/1d90b8f2-b0b6-4f15-afa3-a2533dadf913#epubcfi(/6/128!/4/2%5Bepubmain%5D/2%5Bch10lev1sec3%5D/2/2/1:0) (accessed Apr. 12, 2023).
[29] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.
[30] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[31] "Building a Recurrent Neural Network - Step by Step - v1." https://datascience-enthusiast.com/DL/Building_a_Recurrent_Neural_Network-Step_by_Step_v1.html (accessed Apr. 16, 2023).
[32] "API,Documentation | Stock Prices | Quote Comparison - Yahoo Finance." https://finance.yahoo.com/quotes/API,Documentation/view/v1/ (accessed Apr. 07, 2023).
[33] R. Aroussi, "yfinance: Download market data from Yahoo! Finance API." Accessed: Apr. 08, 2023. [OS Independent]. Available: https://github.com/ranaroussi/yfinance
[34] "statsmodels.tsa.stattools.adfuller — statsmodels." https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.adfuller.html (accessed Apr. 10, 2023).
[35] "statsmodels.stats.diagnostic.acorr_ljungbox — statsmodels." https://www.statsmodels.org/dev/generated/statsmodels.stats.diagnostic.acorr_ljungbox.html (accessed Apr. 16, 2023).