Final Thesis Report.pdf

by Hemant Kokane

Submission date: 18-Mar-2024 09:23AM (UTC+0000)
Submission ID: 226965764
File name: Final_Thesis_Report.pdf (2.58M)
Word count: 29618
Character count: 175878

PREDICTING BITCOIN PRICING TRENDS USING TIME SERIES FORECASTING TECHNIQUES.

HEMANT BAJARANG KOKANE

FINAL THESIS REPORT

MARCH 2024

TABLE OF CONTENTS

DEDICATION

This thesis is dedicated to my loving family, whose unwavering support and encouragement have been my anchor throughout this academic journey. To my parents, whose sacrifices and guidance have shaped me into the person I am today, I owe an immeasurable debt of gratitude. Your belief in me has been my source of strength and motivation.

I also dedicate this work to my dear friends and mentors, whose wisdom, encouragement, and camaraderie have enriched my life in countless ways. Your presence has illuminated my path and made every challenge more manageable.

Lastly, I dedicate this thesis to all those who tirelessly pursue knowledge and strive to make a difference in the world. May our collective efforts lead to a brighter future for generations to come.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis supervisor, Nitin Shelake, for their invaluable guidance, unwavering support, and scholarly insights throughout the entire process of researching and writing this thesis. Their mentorship has been instrumental in shaping the direction of my work and refining my analytical skills.

I am also deeply thankful to Upgrad buddy, Harshita Khanna, mentor Dr. Rupal and team for their constructive feedback and thoughtful suggestions, which have significantly enhanced the quality of this thesis.

I extend my appreciation to the Liverpool John Moores University, whose academic support made this research possible. Their investment in my academic pursuits has been instrumental in facilitating data collection, travel for fieldwork, and access to essential resources.

ABSTRACT

This research will focus on Bitcoin price prediction using long-short-term memory (LSTM) and autoregressive integrated moving average (ARIMA) algorithms. The main objective of this study is to compare Bitcoin prices predicted by both models so I can provide the best Bitcoin price prediction model. I am going to use the BTC-USD dataset which is publicly available on the Kaggle website. The research will begin with an in-depth analysis of historical Bitcoin price trends, identifying key factors influencing market dynamics, and identifying hidden patterns and trends in data. I am going to use some pre-processing techniques and data transformation so I can meet model requirements. I will split the data into 80/20 percent. 80% of the data will be train data and testing data will be 20%. I will use Min-max scaling to normalize the data. Subsequently, I will build machine learning or deep learning algorithms like ARIMA and LSTM and the same will be evaluated based on evaluation metrics. For model evaluation, I am going to use evaluation metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). After evaluating all the metrics thesis suggest that ARIMA model perform better than LSTM model. Therefore, for accurate and precise Bitcoin price predictions, ARIMA is the preferred choice.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. LSTM - Long Short-term Memory
2. ARIMA - Autoregressive Integrated Moving Average
3. SARIMAX - Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors
4. GRU - Gate Recurrent Unit
5. MAE – Mean Absolute Error
6. MSE – Mean Squared Error
7. MAPE – Mean Absolute Percentage Error
8. RMSE – Root Mean Squared Error
9. GARCH - Generalized Autoregressive Conditional Heteroskedasticity
10. VAR - Vector Autoregression
11. BTC – Bitcoin
12. RNN - Recurrent Neural Network
13. EDA – Exploratory Data Analysis
14. VGP - Vanishing Gradient Problem
15. SGD - Stochastic Gradient Descent
16. W-LSTM - Weighted Long Short-Term Memory
17. 2D-CNNLSTM - Two-Dimensional Convolutional Neural Network with Long Short-Term Memory
18. NASA - National Aeronautics and Space Administration

CHAPTER 1: INTRODUCTION

1.1 Background

The cryptocurrency landscape, characterized by its dynamic nature and decentralized structure, has witnessed unprecedented growth, with Bitcoin emerging as a frontrunner in this financial revolution. As the world embraces the possibilities inherent in digital currencies, the volatility and complexity of the cryptocurrency market pose intriguing challenges for investors and researchers alike. Among these challenges, predicting the price movements of Bitcoin stands as a focal point, influencing strategic decision-making, risk management, and investment portfolios.

In this study, I am going to predict Bitcoin prices by using time series forecasting techniques and build a machine-learning algorithm model with high prediction accuracy that helps investors invest money in cryptocurrency or stocks. While doing this study, the main objectives will be to analyze the hidden patterns in the Bitcoin dataset via EDA, determine the optimum pre-processing technique, and evaluate and compare the performance of models based on evaluation metrics.

I am going to compare results obtained from various deep learning algorithms, so I can provide the best algorithm for Bitcoin price prediction. There are lots of models available in the market, like autoregressive integrated moving average (ARIMA), SARIMAX, FbProphet, Long Short-term Memory (LSTM), and Gate Recurrent Unit (GRU). Out of this, I will use ARIMA and LSTM models, to forecast prices. In this study, I will compare the results obtained from both models and evaluate them by using some evaluation metrics. There are some evaluation metrics available, like MAE, MSE, RMSE, MAPE, and R2 Score (Coefficient of Determination). I will use Mean Absolute Error, Root Mean Squared Error, and Mean Absolute Percentage Error. To evaluate model accuracy.

This study is useful to investors or stakeholders in the cryptocurrency space. It can provide information to investors so they can form their decision about buying, selling, or holding Bitcoin. Accordingly, they can change their investment strategies. This model can anticipate potential market fluctuation so investors can minimize their risk. It can provide optimal entry and exit points, which can enhance investor's strategies. Financial institutes can use prediction model to regulate their policies, and they can form cryptocurrency-related policies.

The price of Bitcoin or cryptocurrency depends on demand and the volume available for trading. If investors can forecast the price of a cryptocurrency, they can invest more money. Maximum people are interested in investing in Bitcoin or Cryptocurrency, but they do have not time to search for it. Hence Cryptocurrency price predictions are important (Dinshaw et al., 2022).

Some of the authors build a model that provides the best daily strategy. They broke the solution into three parts. First, they forecast prices by using different models and then choose the best which is the LSTM model. They use the GARCH and VAR models as risk indicators. With the help of the PSO model, they optimize sharp index, and objective function for best strategy. Additionally, different cost functions are provided in this article to generalize our model so investors who have divergent risk attitudes can be saved. Their strategy achieves a gain of $27263 with a VAR value of 502.51 (Huang & Ye, 2022).

According to research both LSTM and ARIMA will perform well while predicting Bitcoin price. The LSTM model predicts Bitcoin prices more accurately than the ARIMA algorithm. While using 4 core CPU it takes more time. And minimum time while using 2 core GPU. ARIMA model is useful for short periods. Whereas the LSTM algorithm predicts a long period. For a short period, ARIMA predicts correctly but when the period increases the accuracy of the ARIMA model decreases (Hua, 2020).

Bitcoin uses a virtual platform for exchange. Bitcoin was launched in Jan-2009. After the 2008 Great Depression or mortgage crisis Bitcoin provided an alternative protection for investors. Bitcoin has more advantages and reliable currency compared to traditional currencies. In the past time, a medium of exchange was money, allowing traders to make purchases. This trust applies to Bitcoin in the cryptocurrency market. There is no single location of Bitcoin. And users control Bitcoin. Every Bitcoin transaction is recorded, and there is a limit supply limit of Bitcoin which is 21 M globally (Sener & Demir, 2022).

1.2 Problem statement

I will use time series forecasting techniques such as Autoregressive integrated moving average (ARIMA) and Long Short-Term Memory (LSMT) to predict Bitcoin prices in this dissertation or thesis. I read a few papers related to this thesis, and they covered the majority of topics, which I am going to touch on in this thesis.

The author compares the accuracy of prices in USD prediction. He uses two time series forecasting techniques, Long Short-term Memory (LSTM) and ARIMA. He gathered data by using PycURL from Bitfine. Keras and TensorFlow interfaces were used to build the LSTM model. He did a classical comparison by using the ARIMA model. He trained the model with 100 epochs and 10 pieces of data in each round. The model was trained with 5 and 10 previous data points. He compares the result of 5 data points with 10 data points. After training the model the average error rate was 0.4765938 and the standard deviation was 2.092208. Both algorithms perform well while forecasting Bitcoin prices. LSTM model took approx. 42 M via a four-core CPU and 1 M while using a two-core GPU. ARIMA model was good for the short term, but the LSTM model was more efficient while predicting long-span Bitcoin prices. (Hua, 2020)

The authors did a comparative study between LSTM and ARIMA models for short-term prediction of Bitcoin Prices. They forecast Bitcoin prices for the next day by using the static forecast method. To check the accuracy of prediction they split the data into train and test datasets. In the first training sample ARIMA model forecast better than the LSTM algorithm. But in the second training sample LSTM algorithm predicts better than the ARIMA algorithm. According to the authors, the accuracy of the LSTM model was 99.73 percent. Which was much better than the ARIMA model. ARIMA model predicts accurately because of an upward trend in data. Otherwise, ARIMA would not be able to predict correctly. (Latif et al., 2023)

The authors did a Bitcoin Price prediction based on machine-learning techniques and they did public sentiment analysis also. For sentiment analysis, they used Twitter and Reddit posts. They could use Facebook and LinkedIn data for better public opinion collection. The root-mean-square-error of the LSMT model with a single feature was 198.448 and with multiple features was 197.515 whereas the root-mean-square-error of the ARIMA model was 209.263 which states that the LSTM model performed accurately. The result shows that the LSTM algorithm predicts more accurately than the ARIMA algorithm for a long span. (Raju & Tarif, 2020)

The authors predicted the rates of three CTCs, namely Ethereum, BTC, and Litecoin. To predict cryptocurrency prices, they used machine learning algorithms like gated recurrent unit (GRU), LSTM, and Bi-LSTM. From the results of all three algorithms, they found that the GRU model predicts more accurately when compared with the LSTM and Bi-LSTM models. GRU presents MAPE per. of 0.24, 0.8, and 0.2 approx. for Bitcoin, Litecoin, and Ethereum respectively. Bi-LSTM presents MAPE per. of 5.99, 6.85, and 2.33 for Bitcoin, Litecoin, and Ethereum respectively. According to the authors, the GRU model is best for forecasting cryptocurrencies. (Hamayel & Owda, 2021)

To predict Bitcoin prices author uses data from the Bitcoin Price Index. They used machine learning techniques like Bayesian optimized recurrent neural network (RNN) and a Long-Short-Term-Memory (LSTM) network. In this study, the authors found that the LSTM model performed well. The LSTM model achieves the highest accuracy of 52 percent. And RMSE was 8 percent of the LSTM model. According to the authors, the ARIMA model performed poorly when compared with both deep learning models. The LSTM model outperformed the RNN model. However, the LSTM takes more time to train compared to the RNN. (McNally et al., 2018)

By using deep learning and machine learning techniques like LSTM, ARIMA, XGBoost, and Prophet authors predict Bitcoin prices. After Building models, authors compared metrics like Root-Mean-Square-Error, Mean-Absolute-Error, and R2. The authors also used sentiment analysis combined with the LTMS model to predict Bitcoin prices. After comparing the metrics of all models, they concluded that the LTMS model with sentiment analysis performed better than others. (Ramani et al., 2023a)

Lots of work done in this domain. Still, a few challenges are there in this domain. Lots of external environmental factors and sentiment affect Bitcoin prices. Some of the authors included sentiment analysis by using posts on social media.(Mishal et al., 2022) However, posts on social media may be biased. So, it is required to build a hybrid LTMS model with sentiment analysis which could use authentic posts for sentiment analysis. While predicting Bitcoin prices volume of Bitcoin trading a day should be considered it will improve prediction accuracy.

1.3 Aim and Objectives

The main aim of this research is to predict Bitcoin prices by using time series forecasting techniques. The goal of this research is to obtain a machine-learning algorithm model with high prediction accuracy that helps investors invest money in cryptocurrency or stocks.

The objectives of the research are outlined as follows.

- To analyze the hidden patterns in the Bitcoin dataset via EDA and visualization for a better understanding of the data trend.
- To determine the optimum pre-processing technique required for the LSTM and ARIMA model.
- To propose LSTM and ARIMA models to forecast Bitcoin price.
- To evaluate and compare the performance of both models based on evaluation metrics.

1.4 Significance of Study

A Bitcoin price-predicting model is useful to investors or stakeholders in the cryptocurrency space. It can provide information to investors so they can form their decision about buying, selling, or holding Bitcoin. Accordingly, they can change their investment strategies. This model can anticipate potential market fluctuation so investors can minimize their risk. It can provide optimal entry and exit points, which can enhance investor's strategies. Financial institutes can use prediction model to regulate their policies, and they can form cryptocurrency-related policies. The Bitcoin price prediction model can contribute to the overall growth and stability of the cryptocurrency world. The public can think the Bitcoin as an investment plan if the model predicts Bitcoin price accurately.

In this research, I am going to compare Machine learning algorithms like ARIMA and LSTM models. By evaluating the results of both models using various metrics, I can conclude which model predicts Bitcoin prices accurately. So people can use the perfect model while predicting cryptocurrency.

1.5 Scope of Study

1.5.1 In-scope of study

I am going to use historical data for predicting this model, so the model can capture trends, patterns, and seasonality in the Bitcoin prices. This study can provide short to medium-span forecasting, it can provide predictions for the next day, week, or month. It could be helpful for intra-day trading because it may provide optimal entry and exit points within a day. This study can help to assess the volatility of cryptocurrency prices over time.

1.5.2 Out-scope of study

This study may not be accurate for long-term prediction, especially when it tries to predict events that are out of the scope of historical data. There may be some unpredicted events or unexpected market shifts that are not present in train or historical data. This study may not work properly with the unpredictable nature of the financial market also called black swan events. This model may not cover global political factors and investor sentiments. This study may not cover price fluctuation because of Artificial or manual manipulation in the market.

1.6 Required Resources

1.6.1 Hardware Requirement

- Central Processing Unit (CPU): A standard multicore CPU
- Graphics Processing Unit (GPU): Graphics like NVIDIA
- Random Access Memory (RAM): At least 4 GB RAM

1.6.2 Software Requirement

- Python with the updated version.
- Deep Learning Frameworks like TensorFlow and Keras.
- Jupyter Notebook with the updated version.
- Libraries like Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Math, Stat
- Cloud-based platforms like Google Colab

1.7 Structure of The Study

The structure of the study is as follows:

- Chapter 1: Introduction-

  This chapter provides introduction and background for this research work.
- Chapter 2: Literature Review-

  This chapter mentions the related works in the fields of Automatic story generation and prompt-learning.
- Chapter 3: Research Methodology-

  This chapter gives a detailed walkthrough of the methodology followed during the experimentation stage.
- Chapter 4: Analysis-

  This chapter details the different experiments performed for the story generation task.
- Chapter 5: Results and Discussions-

  This chapter discusses the results of the experiments performed in Chapter 4.
- Chapter 6: Conclusion and Recommendation-

  This chapter concludes the work done in the thesis and discusses future improvements.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

2.1.1 Bitcoin Overview

According to research findings, Bitcoin is classified as a form of digital currency that functions within a decentralized network known as blockchain. It was established in 2009 by an unidentified individual or collective employing the alias Satoshi Nakamoto. Bitcoin is renowned for its distinctive attributes that set it apart from conventional fiat currencies. The aforementioned attributes encompass decentralization, restricted stock availability, pseudonymity, and transparency. Decentralization refers to the absence of central authority, such as a government or central bank, in the control of Bitcoin. Conversely, it is upheld and managed by a global network of computers. The supply of Bitcoin is finite, as it is constrained to a maximum of 21 million coins that can be generated. The purpose of this scarcity is to mitigate inflation and uphold the currency's worth. Pseudonymity pertains to the utilization of cryptographic addresses rather than real-world identities to record Bitcoin transactions on the blockchain. Although the transactions are publicly accessible, the identities of the parties involved are not directly associated with their respective addresses.(Alazzam et al., 2023; Ciarko et al., 2023)

According to a study, transparency is identified as another crucial attribute of Bitcoin. Every transaction is documented on the blockchain, a publicly accessible ledger. The presence of transparency facilitates enhanced accountability and mitigates the potential for fraudulent activities. Bitcoin functions inside a worldwide market, wherein its valuation is influenced by the interplay of supply and demand forces. The price of Bitcoin exhibits a notable degree of volatility, characterized by substantial variations that transpire within brief time intervals. The Bitcoin market is subject to a multitude of aspects, encompassing investor emotion, legislative changes, technological progress, and macroeconomic circumstances. Several factors have the potential to influence the demand for Bitcoin, hence affecting its price. Gaining insight into the market attributes of Bitcoin is crucial for researchers and investors seeking to forecast its price. Through the examination of past data and the identification of recurring trends, scholars have the ability to construct prognostic models that can anticipate forthcoming fluctuations in prices. (Centobelli et al., 2022; De Pace & Rao, 2023)

2.1.2 Significance of Accurate Cryptocurrency Price Prediction

Accurate price forecast in cryptocurrency markets holds substantial importance for various reasons, as indicated by a study. This tool aids investors and traders in making well-informed decisions on the purchase, sale, or retention of cryptocurrencies, hence mitigating the potential for financial losses. Additionally, this technology facilitates the identification of possible profit possibilities and the optimization of returns for market participants. The ability to make precise price predictions can be helpful in the formulation of efficient trading strategies and risk management methodologies, thereby augmenting the overall efficiency of the market. Moreover, this phenomenon has the potential to enhance the stability and trustworthiness of the cryptocurrency market, thereby attracting a larger number of players and facilitating its expansion. Moreover, the precise forecasting of prices can enhance the advancement of derivative products and financial instruments centered around cryptocurrencies, thereby broadening the array of investment alternatives accessible. 5,6

Overall, accurate price prediction in cryptocurrency markets is crucial for both individual investors and the broader financial ecosystem, promoting transparency, efficiency, and growth. (Ammer & Aldhyani, 2022; Danylkiv et al., 2022)

2.1.3 Introduction to LSTM and ARIMA Techniques

The acronym LSTM refers to Long Short-Term Memory, a specific architecture within the field of recurrent neural networks (RNNs). Recurrent Neural Networks (RNNs) are a kind of artificial neural networks specifically developed for the purpose of handling sequential data, including time series data. Long Short-Term Memory (LSTM) networks have demonstrated notable efficacy in capturing enduring dependencies within datasets, rendering them well-suited for applications such as time series forecasting. Long Short-Term Memory (LSTM) networks have demonstrated notable efficacy in addressing enduring dependencies within time series data, a critical aspect for achieving precise forecasting. They employ a memory cell and diverse gates to regulate the transmission of information, enabling them to save crucial data across extended sequences. (Elsaraiti & Merabet, 2021; Hua, 2020; Latif et al., 2023)

ARIMA is an acronym for Autoregressive Integrated Moving Average, a statistical model employed for the study and prediction of time series data. The model consists of three constituent elements, namely autoregressive (AR), moving average (MA), and differencing (I). The autoregressive (AR) component represents the linear correlation between an observation and a specific number of previous observations. The moving average (MA) component represents the error term as a linear mixture of previous error terms. The differencing component eliminates trends and seasonality from the data. In contrast, ARIMA models are grounded in statistical principles and are especially valuable for the examination and prediction of time series data that demonstrate patterns and seasonal variations. Time series models have the capability to effectively represent and forecast various types of data, encompassing economic indicators, stock prices, and weather patterns. (Elsaraiti & Merabet, 2021; Hua, 2020; Latif et al., 2023)

Predicting future values of a variable based on its historical values is the objective of time series forecasting. It finds widespread application in several domains including finance, economics, weather prediction, and stock market analysis. The use of LSTM and ARIMA in time series forecasting is prevalent owing to their capacity to effectively capture temporal dependencies and patterns inherent in the data. LSTM and ARIMA both has distinct advantages and disadvantages, and the selection between them relies on the particular attributes of the data and the forecasting objective. LSTM networks exhibit greater adaptability and has the ability to comprehend intricate patterns within the data, albeit necessitating a substantial quantity of training data and processing resources. Conversely, ARIMA models are more straightforward and comprehensible, however they may encounter difficulties when dealing with data that is very nonlinear and nonstationary. In brief, LSTM and ARIMA are widely employed predictive modeling methodologies widely utilized in the field of time series forecasting. Long Short-Term Memory (LSTM) networks exhibit significant efficacy in capturing enduring dependencies, but ARIMA models, grounded in statistical principles, prove valuable in the analysis of data trends and seasonality. The selection between the two options is contingent upon the distinct attributes of the data and the objective of the forecasting endeavor. (Elsaraiti & Merabet, 2021; Hua, 2020; Latif et al., 2023)

2.2 LSTM for Bitcoin Price Prediction

2.2.1 LSTM Architecture

Many studies show that LSTM is a type of RNN that is specifically designed to handle and process sequential data, such as time series data. In the context of time series data, LSTM is suitable because it can effectively capture and model the temporal dependencies or patterns present in the data. LSTM achieves this by using a memory cell, which allows it to remember and store information over long periods. This memory cell is equipped with gates that control the flow of information, allowing the LSTM to selectively retain or forget certain information based on its relevance to the current time step. (Abbasimehr & Paki, 2022; Song et al., 2020; Sorkun et al., 2020)

According to existing research, the LSTM architecture is comprised of several memory cells, each possessing its own distinct set of gates. The interconnected memory cells form a sequential arrangement, wherein the output of one cell serves as the input for the subsequent cell. The LSTM model possesses the capability to effectively process and disseminate information over time, rendering it highly suitable for the analysis of time series data. The cell state serves as the memory element within the Long Short-Term Memory (LSTM) architecture, facilitating the addition or removal of information through the utilization of gates. Conversely, the hidden state represents the output of the LSTM cell, encompassing details pertaining to the input, preceding hidden state, and present cell state. (Abbasimehr & Paki, 2022; Song et al., 2020; Sorkun et al., 2020)

Studies have shown that LSTM is well-suited for time series data due to its capability to effectively address the issues of vanishing and bursting gradients, which are frequently encountered concerns while training RNNs. Long Short-Term Memory (LSTM) models have demonstrated efficacy in capturing long-term dependencies within sequential data, addressing the issue of disappearing gradients, managing sequences of varying lengths, and modeling intricate patterns within time series datasets. (Abbasimehr & Paki, 2022; Song et al., 2020; Sorkun et al., 2020)

In general, the LSTM architecture is specifically tailored to process sequential data and is especially well-suited for time series data because of its capacity to capture temporal relationships and address gradient issues. The LSTM architecture is highly suitable for time series forecasting and other tasks that involve temporal patterns due to its capacity to capture long-term dependencies and successfully handle sequential data. (Abbasimehr & Paki, 2022; Song et al., 2020; Sorkun et al., 2020)

2.2.2 Reviews on Bitcoin Price Prediction Using LSTM

Several authors have done studies utilizing Long Short-Term Memory (LSTM) for the purpose of predicting Bitcoin prices. A study conducted by Rui Zhong utilized Long Short-Term Memory (LSTM) and a hybrid model incorporating wavelet decomposition and LSTM (W-LSTM) to forecast the price and price movement of Bitcoin based on historical data. In their study, Sajedi and Arjmand utilized a hybrid 2D-CNNLSTM model, incorporating OPTUNA hyperparameter adjustment, to effectively forecast Bitcoin values. Their performance surpassed that of other frequently adopted algorithms. Mr. R. Arunachalam employed Long Short-Term Memory (LSTM) within a Recurrent Neural Network (RNN) to forecast Bitcoin prices, consistently outperforming statistical techniques. LSTMs have strong performance in predicting Bitcoin prices. In a particular investigation, it was observed that LSTM models had superior performance in comparison to statistical approaches and alternative deep learning algorithms. Additionally, a separate study demonstrated that the integration of LSTM with sentiment analysis had superior performance in forecasting Bitcoin values when compared to alternative machine learning algorithms.

2.2.2.1 Analysis of LSTM in Long-Term Dependencies

LSTM neural networks are good at predicting Bitcoin price and capturing long-term dependencies. They are especially good at finding temporal dependencies and long-term patterns in sequential data, which makes them good for processing and analyzing data with long-term dependencies. However, LSTM models still have trouble with the vanishing gradient problem (VGP), which slows network convergence and lowers prediction accuracy.(-, 2023; Al-Selwi et al., 2023; Gunarto et al., 2023)

LSTM networks are particularly suitable for regression problems that involve sequential data, as they possess the capability to effectively assess and process long-term dependencies. Nevertheless, Long Short-Term Memory (LSTM) models are susceptible to the vanishing gradient problem (VGP), which impedes the learning process and results in subpar prediction accuracy. LSTM and Gated Recurrent Unit (GRU) were created as gated units to tackle this problem, but, VGP remains unresolved even inside these units. Numerous scholarly investigations have been dedicated to examining the inadequacy of LSTM networks in achieving long-term dependency convergence as a result of VGP. The power law forget gate has been suggested as a potential substitute for the exponential decay of information in Long Short-Term Memory (LSTM) models, leading to enhanced efficacy in capturing spatial relationships over extended distances. Moreover, the performance of LSTM networks is greatly influenced by the optimization approach employed. Among these methods, Stochastic Gradient Descent (SGD) has been recognized as the most exceptional for predicting power consumption using LSTM models. The Extended LSTM (E-LSTM) architecture is an alternative method to tackle the problem of parameter size in LSTMs. It effectively decreases the required number of parameters without compromising performance. (Al-Selwi et al., 2023; Turek et al., 2022)

This study presents an empirical investigation of Long Short-Term Memory (LSTM) networks, focusing on the issue of inefficiency in long-term dependency convergence due to the vanishing gradient problem. The authors primarily concentrate on the instance of degradation in NASA turbofan engines. The objective of this study is to conduct an empirical examination of LSTM networks, focusing on the inefficiency in achieving convergence of long-term dependencies due to VGP. This study focuses on the examination and empirical analysis of case studies pertaining to the degradation of NASA's turbofan engine. This study suggests that recurring Neural Networks (RNNs) are well-suited for regression issues that involve sequential data. This is because RNNs have a recurring internal structure that allows them to analyze and process data over lengthy periods of time. Nevertheless, Recurrent Neural Networks (RNNs) are susceptible to the phenomenon known as the phenomenal vanishing gradient issue (VGP), which leads to the network's cessation of learning and the production of subpar prediction accuracy, particularly in cases with long-term dependencies. (Al-Selwi et al., 2023)

According to existing research, the Long Short-Term Memory (LSTM) algorithm has been found to possess the ability to effectively manage long-term dependencies on its input and make predictions for an extended duration. According to the research conducted on all models, the model with the highest RMSE value of 0.699 was model 1, which had 2 hidden layers and 64 neurons. The RMSE results can be considerably influenced by increasing the number of hidden layers, namely by utilizing neurons 16 and 32 in Model 1. As indicated in Table 2.1. (Saputra et al., 2022)

Table 2.1 RMSE Experiment Results

HL	Neuron														
	Model 1					***Model 2***					***Model 3***				
	16	***32***	***64***	***128***	***256***	***16***	***32***	***64***	***128***	***256***	***16***	***32***	***64***	***128***	***256***
2	1.816	1.372	**0.699**	0.830	1.027	1.319	1.159	**0.851**	1.270	1.446	1.177	0.834	**0.701**	0.853	0.962
3	2.324	1.951	1.463	1.079	1.124	1.220	1.122	1.247	1.504	2.357	1.136	0.947	1.315	0.996	1.137
4	9.835	9.813	9.790	9.788	9.779	9.669	9.492	3.586	1.434	1.774	1.534	1.247	1.530	1.191	5.805
5	9.843	9.824	9.810	9.791	9.782	9.613	9.587	9.531	9.486	9.420	9.960	9.827	9.733	9.695	9.779
6	9.849	9.829	9.807	9.795	9.778	9.624	9.583	9.529	9.486	9.416	9.930	9.838	9.754	9.668	9.829
7	9.846	9.830	9.810	9.791	9.780	9.632	9.571	9.539	9.477	9.412	9.936	9.847	9.719	9.683	9.899
8	9.847	9.828	9.809	9.789	9.779	9.629	9.592	9.546	9.468	9.408	9.937	9.845	9.716	9.687	9.772
9	9.848	9.825	9.812	9.729	9.783	9.614	9.586	9.540	9.489	9.406	9.957	9.839	9.742	9.681	9.762
10	9.846	9.827	9.811	9.789	9.781	9.592	9.595	9.534	9.482	9.402	9.940	9.858	9.734	9.671	9.775

2.2.2.2 Key factors impacting LSTM performance

LSTM, short for Long Short-Term Memory, is a prevalent recurrent neural network (RNN) employed in diverse domains such as natural language processing, speech recognition, and time series analysis. Several crucial aspects can impact the performance of an LSTM model, as outlined below:

Research indicates that the performance of the model is significantly influenced by the quantity of LSTM cells in each layer, the number of layers, and the interconnections among them. Augmenting the quantity of LSTM cells initially enhances prediction, however over a specific threshold results in diminished outcomes. The quantity of layers also exerts an influence, as the inclusion of additional layers might enhance the overall accuracy of performance. Furthermore, the interconnections among the layers can be adjusted in order to boost parallelism and utilize hardware more effectively, leading to higher throughput and improved latency and resource consumption. Consequently, it is imperative to meticulously evaluate the quantity of LSTM cells, the number of layers, and the interconnections among them in order to attain maximum performance in LSTM models. (Dai et al., 2020; Jia & Zhang, 2020)

The performance of an LSTM model is significantly influenced by the selection of hyperparameters, as shown by researchers. The accuracy and dependability of the model can be influenced by many hyperparameters, including the optimizer, activation function, batch size, and the number of LSTM layers. The selection of the optimizer was determined to exert a significant impact on the classification efficacy of the LSTM model. Conversely, the model's performance is less affected by other characteristics such as the quantity of LSTM layers. The optimization of hyperparameters plays a pivotal role in attaining optimal model performance and expediting the training process. By optimizing hyperparameters, LSTM models can achieve enhanced prediction accuracy and reduced training time. Optimization algorithms such as Bayesian optimization - Hyperband (BOHB) and Bees Algorithm (BA) can further improve the performance of the LSTM model. In order to optimize the performance of an LSTM model in different applications, it is crucial to carefully choose the appropriate hyperparameters. (Andhika Viadinugroho & Rosadi, 2023; Sher et al., 2023; S. Wang et al., 2022)

Research has indicated that the selection of data preparation techniques exerts a substantial influence on the efficacy of an LSTM model. LSTM has demonstrated significant benefits in predicting time-series data within the realm of deep learning applications.

Nevertheless, the efficacy of Long Short-Term Memory (LSTM) is suboptimal in tests including multi-dimensional industrial test data. Several preprocessing approaches have been suggested for data preprocessing, including de-redundancy compression, pooling processing, and noise removal. Datasets with high correlation coefficients are chosen for pooling and denoising by comparing correlation coefficients and thresholds. The findings demonstrate that data preprocessing enhances the precision of predictions and validates the efficacy of the preprocessing technique. Moreover, within the realm of stock market forecasting, the process of data preparation holds significant importance in order to extrapolate predictions. Research has been conducted on various data manipulations, with a focus on comparing their fidelity. The selection of data preprocessing procedures is of paramount importance in improving the efficacy of Long Short-Term Memory (LSTM) models. (Tunkiel et al., 2022)

The performance of the LSTM model is positively influenced by augmenting the quantity of training data. It aids in mitigating overfitting and enhancing the accuracy and stability of predictions. The training performance can be further enhanced by enhancing the precision of the training data, specifically by reducing the step size. Subject-based training approaches that preserve the entire sequence for each student can effectively address the Knowledge Tracing problem and enhance the performance of the original model by around 50%. Furthermore, it is observed that the initial model necessitates a just 10% increase in student data to exceed the original performance when the prediction model is of smaller scale, and a 50% increase in data when the prediction model is of larger scale. (Choi et al., 2022; Peng et al., 2022; J. T. Shen & Lee, 2023)

Existing research has demonstrated that the utilization of regularization approaches significantly influences the performance of Long Short-Term Memory (LSTM) models. The application of kernel flow regularization on LSTM layers has been observed to be efficacious in mitigating overfitting issues within the domain of time series forecasting. Furthermore, it has been observed that the utilization of kernel flow and dropout regularizations, along with early stopping on LSTM layers, yields optimal outcomes when applied to certain time series datasets. An additional investigation revealed that the implementation of 20% Dropout regularization on a GRU model, which was trained on a limited dataset, resulted in a reduction of the root mean square error (RMSE) by 23%. In addition, previous studies have demonstrated that the utilization of a pretrained LSTM-based language model (LM) as a regularizer can enhance the precision of LSTM classifiers during training with limited datasets. In general, the aforementioned findings underscore the need of selecting suitable regularization strategies in order to enhance the efficacy of LSTM models. (Balestriero et al., 2022; Carpenter et al., 2021; Shirdel et al., 2021)

Several research have indicated that the performance of LSTM models is significantly influenced by computational resources. The utilization of limited computing resources can lead to expedited training durations for Long Short-Term Memory (LSTM) models, rendering them appropriate for scenarios characterized by temporal limitations. Furthermore, the incorporation of resource-efficient building blocks and activation functions has the potential to result in substantial reductions in resource consumption and enhanced performance of Long Short-Term Memory (LSTM) models. However, the intricate neurons and internal states of Long Short-Term Memory (LSTM) networks necessitate computing resources, hence posing difficulties in achieving rapid processing speeds for real-time applications on low-power, low-cost edge devices. Hence, the efficiency and effectiveness of LSTM models are heavily influenced by the availability and allocation of computational resources. (Ashawa et al., 2022; Mandke et al., 2021)

In short, the performance of an LSTM model is influenced by various factors, including its architecture, hyperparameters, data preprocessing, training data size, regularization techniques, and computational resources. Understanding and optimizing these factors can help researchers achieve better performance and results with their LSTM models.

2.2.3 Critical evaluation of LSTM-based models

LSTM-based models have been widely researched in the context of Bitcoin price dynamics, as demonstrated by the papers. The aforementioned algorithms have demonstrated encouraging outcomes in accurately forecasting the daily fluctuations in Bitcoin's price. Statistical approaches and other deep learning algorithms are surpassed in terms of accuracy by these methods. LSTM models have demonstrated superiority over statistical methods in predicting Bitcoin prices. Historical data has been utilized to forecast both the price and price movement (trend) of Bitcoin. The input characteristics for these models consist of the closing price, fundamental trading data, and technical indicators derived exclusively from fundamental trading data. Parameters such as batch size, number of neurons, and length of time sequences have an impact on the accuracy of Bitcoin price prediction using LSTM models. The aforementioned findings offer valuable insights into the optimization of different parameters during the construction of LSTM models to enhance the accuracy of Bitcoin price prediction. (Lin, 2023; Mr. R. Arunachalam et al., 2023; Zhong, 2023)

Studies found that LSTM models achieve better performance in predicting Bitcoin price compared to statistical methods. Deep learning algorithms, specifically LSTM in RNN, are superior for Bitcoin price prediction. LSTM model accurately predicts future Bitcoin prices. More Bitcoin price features need to be considered for increased efficiency. Some papers find that, close price as input obtains the best performance for price prediction. No improvement in price movement prediction with different input features or models.

2.3 ARIMA for Bitcoin Price Prediction

2.3.1 Overview of ARIMA model

The ARIMA model is a commonly employed statistical technique for predicting time series data. Stationarity is achieved by combining autoregressive (AR) and moving average (MA) components with differencing. The ARIMA model is represented by the equation (p, d, q), where p represents the order of the autoregressive component, d represents the order of differencing, and q represents the order of the moving average component.

The autoregressive (AR) component measures the relationship between current and previous observations, the instrumental variable (I) component applies differencing to maintain stationarity, and the moving average (MA) component considers the relationship between a current observation and a lagged residual. The process of establishing a model entails several key steps, including assessing stationarity, selecting suitable parameters (p, d, q), constructing the model, verifying its performance, and generating future forecasts.

The ARIMA model finds wide-ranging applications in the field of time series forecasting, encompassing various domains such as financial forecasting (e.g., stock prices, currency exchange rates), demand forecasting (e.g., sales, product demand), energy consumption forecasting, temperature and weather forecasting, economic indicator forecasting (e.g., GDP, inflation), traffic flow prediction, and healthcare forecasting (e.g., patient admissions, disease outbreaks).

The value of ARIMA lies in its simplicity and interpretability, which enable it to effectively capture temporal trends within datasets. Nevertheless, it may encounter difficulties when dealing with non-linear connections or abrupt fluctuations in data. In such instances, it may be prudent to contemplate employing more sophisticated models such as SARIMA or machine learning methodologies. In general, the ARIMA model is a robust tool utilized by analysts and researchers in the analysis of time series data. It offers a methodical framework for comprehending and predicting temporal trends in diverse fields.

2.3.2 Reviews on Bitcoin Price Prediction using ARIMA

AutoRegressive Integrated Moving Average (ARIMA) models have been utilized by researchers for the purpose of predicting Bitcoin prices, owing to its straightforwardness and efficacy in capturing temporal patterns. Research studies commonly adhere to a standardized approach, which encompasses many stages such as data preprocessing, model selection, parameter tuning, and implementation of training and evaluation. Nevertheless, the sensitivity of Bitcoin to external influences and its intrinsic market volatility provide obstacles that may restrict the effectiveness of ARIMA in capturing intricate linkages and abrupt fluctuations.

Ultimately, although ARIMA has been utilized in forecasting Bitcoin prices, the intricate nature and instability of the cryptocurrency suggest that integrating ARIMA with other methodologies or embracing sophisticated models may be imperative for achieving more precise predictions.

2.3.2.1 ARIMA's ability to capture short-term trends

The efficacy of ARIMA in capturing short-term trends and seasonality can be evaluated through a range of empirical investigations. A study conducted a comparison between ARIMA and LSTM models for predicting Bitcoin values. The study revealed that LSTM demonstrated superior performance compared to ARIMA in accurately capturing both the direction and value of Bitcoin prices within given time periods. A different study introduced a seasonal prediction model that relies on consistent seasonal patterns and employed ARIMA to perform short-term predictions. The results demonstrated a high level of prediction accuracy for time series with consistent seasonality. In addition, a research conducted on temperature data in Balikpapan employed the ARIMA model to construct a Triple Seasonal ARIMA model, which demonstrated high precision in forecasting various time periods. In addition, a methodology for diagnosing seasonality has established a connection between the notion of seasonality and the foundations of autoregressive (AR) and moving average (MA) models. This methodology offers a statistical tool for evaluating various types of seasonality, such as dynamic and stable seasonality. Collectively, these studies illustrate the efficacy of ARIMA in capturing transient patterns and seasonal variations across diverse fields. (Giovani et al., 2022; Latif et al., 2023; McElroy, 2021)

Some papers show that The Seasonal ARIMA model was very good at predicting 2 weeks. - The Triple Seasonal ARIMA model was accurate for 6 months. - The Seasonal ARIMA model was very good at predicting 2 weeks. The triple seasonal ARIMA model was accurate for a period of 6 months.

2.3.2.2 limitations and Challenges in ARIMA

ARIMA models posit linear correlations between variables, as stated in certain studies. This could potentially hinder their efficacy in capturing intricate, non-linear patterns that may exist in some time series data. Moreover, ARIMA models may encounter difficulties in effectively capturing intricate seasonal trends within the dataset. Although SARIMA (Seasonal ARIMA) effectively tackles certain seasonality concerns, it may not be enough for datasets characterized by erratic or fluctuating seasonal trends. (Dimri et al., 2020; Ospina et al., 2023)

According to several papers, ARIMA necessitates the time series to exhibit stationarity. Attaining stationarity may need differencing, however, in reality, certain data may exhibit resistance to stabilization. Data that is not stationary can result in model performance that is not dependable. It can be challenging to determine the appropriate order of autoregressive (p) and moving average (q) components. When lag orders are specified incorrectly, it can lead to suboptimal model performance and imprecise forecasts. (Mohamed, 2020; C. C. Wang et al., 2021)

Research indicates that ARIMA models exhibit sensitivity towards outliers. The presence of extreme values within a time series can have a disproportionate impact on the estimate of parameters, resulting in models that are less than ideal. Furthermore, ARIMA models include the assumption that the fundamental process of generating data is consistent throughout time. Due to their inherent intent to capture progressive changes, these models may exhibit suboptimal performance when confronted with structural changes or abrupt shifts in the time series. (Perone, 2020)

The scholarly literature examines the assumption of constant variance (homoscedasticity) in the residuals inside the ARIMA model. If there is a shift in variance over time, it is possible that the model may not effectively capture this particular trait, resulting in predictions that are not correct. Moreover, it is usually observed that ARIMA models are better suited for short-term forecasting. The efficacy of their performance may diminish when employed for long-term forecasting, particularly in instances when the fundamental dynamics of the data undergo alterations. (Mohamed, 2020; Siami-Namini et al., 2018)

2.3.3 Critical Evaluation of ARIMA Model

The ARIMA model is a commonly utilized time series forecasting model in diverse domains such as finance, economics, and meteorology. The ARIMA model operates under the premise that the projection of future values in a time series can be achieved via the examination of its historical values and the patterns they exhibit. It captures the relationship between the present value of the time series and its prior values through the autoregressive component. The differencing component is employed to achieve stationarity in the time series, hence eliminating any underlying trend or seasonality inherent in the data. The moving average component of the time series model represents the correlation between the present value of the time series and the residual errors resulting from the preceding predictions. When employing the ARIMA model for forecasting bitcoin prices, it is crucial to thoroughly study its performance in order to evaluate its precision and dependability. This assessment entails juxtaposing the projected prices with the factual prices of bitcoin throughout a designated timeframe. The performance of the ARIMA model can be assessed using a range of statistical measures, including mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). These measurements serve to quantify the disparity between the projected prices and the observed prices, so offering valuable insights into the predictive model's accuracy and precision. Furthermore, it is possible to conduct a visual examination of the projected and observed prices in order to detect any recurring trends or inconsistencies. The assessment of the ARIMA model's efficacy in forecasting bitcoin prices and facilitating informed decision-making is facilitated by a critical review of the model's performance.

2.4 Comparative Analysis of LSTM and ARIMA Model

2.4.1 Strengths and Weaknesses of LSTM and ARIMA

Studies indicate that LSTM and ARIMA models have been employed for forecasting the price of Bitcoin. LSTM, a deep learning system, has shown strengths in terms of prediction accuracy. It has been observed that combining sentiment analysis with LSTM delivers greater performance compared to other methods. In contrast, it has been shown that ARIMA, a conventional time-series model, exhibits diminished predictive accuracy as a result of the substantial volatility exhibited by the Bitcoin price. When comparing the two models, it has been observed that LSTM exhibits superior performance in terms of prediction accuracy when compared to ARIMA. Nevertheless, it is crucial to acknowledge that the accuracy of LSTM can be influenced by various aspects, including batch size and the number of neurons. In general, the LSTM model exhibits potential in effectively forecasting the price of Bitcoin, whereas the

ARIMA model may not be as proficient in addressing the substantial volatility inherent in the cryptocurrency market. (Pan, 2023; Ramani et al., 2023b)

Research has demonstrated that LSTM models provide multiple advantages in forecasting the price of Bitcoin. LSTM models, particularly Long Short-Term Memory in RNN, have demonstrated superior accuracy compared to statistical techniques. The achieved accuracies for daily price forecast surpass that of benchmark results. In terms of R2 and MAPE, LSTM models outperform other frequently employed algorithms like CNN, LSTM, and GRU. LSTM models possess the capability to engage in real-time forecasting, rendering them well-suited for the prediction of Bitcoin values within dynamic market environments. In addition, the examination of LSTM model parameters has demonstrated that modifying variables such as batch size, number of neurons, and duration of time sequences can enhance the precision of Bitcoin price forecasting. In general, LSTM models exhibit a notable level of accuracy, surpassing alternative algorithms, and give the advantage of parameter adjusting flexibility, hence enabling precise prediction of Bitcoin prices. (Lin, 2023; Mr. R. Arunachalam et al., 2023)

significant research have identified significant shortcomings associated with LSTM models used for predicting Bitcoin price. To begin with, the provision of higher batch sizes during minor epochs results in a decrease in the accuracy of predictions. Additionally, the accuracy of the predictions is influenced by the quantity of neurons present in the model. Furthermore, the utilization of a solitary time sequence with a length of 7 as opposed to durations of 14, 30, or 60 leads to an increased level of prediction inaccuracy. Furthermore, the utilization of closing prices from the preceding two years, as opposed to the preceding one, three, or five years, has the potential to enhance the precision of predictions. In addition, the selection of random data chunks for model training has the potential to result in unfitting outcomes and diminish the accuracy of predictions. The performance of LSTM models can be influenced by the selection of hyperparameters, including the activation function, learning rate, and dropout values. (Mr. R. Arunachalam et al., 2023)

According to authors, ARIMA models excel in predicting Bitcoin prices because they can effectively capture the typical temporal relationships in time-series data. These models employ historical prices as a means to estimate the stochastic sequence of Bitcoin prices over a given period, hence enabling precise forecasts of forthcoming values. The efficacy of ARIMA models in short-term forecasts has been demonstrated by their application in the analysis and forecasting of the adjusted closing price of Bitcoin. The models have been subjected to comparison and evaluation using performance metrics such as AIC, AICc, and BIC. Among these models, ARIMA (0, 2, 2) has demonstrated marginally superior outcomes. Furthermore, ARIMA models have been employed to simulate the patterns of Bitcoin's performance within a designated timeframe, and a comparison analysis has been carried out to ascertain the most suitable model. In general, ARIMA models provide a high level of precision and resilience in forecasting the price of Bitcoin, rendering them important instruments for investors and researchers. (Si, 2022; Srivastava et al., 2023; Vital et al., 2023)

specific authors argue that ARIMA models have specific limitations when it comes to predicting Bitcoin values. Firstly, it is possible that they may have difficulties in accurately predicting long-term trends, as their practical utility is constrained in this aspect. Moreover, ARIMA models are limited to monitoring the trajectory of Bitcoin prices and may not possess the capability to forecast both the direction and magnitude within a certain timeframe. Moreover, the inherent instability of Bitcoin can impact the efficacy of ARIMA models, resulting in inaccurate forecasts. Finally, it should be noted that ARIMA models may exhibit divergence and diminished accuracy in instances where the price trend of Bitcoin undergoes abrupt fluctuations. The aforementioned limitations underscore the necessity for alternative models, such as LSTM or Fbprophet, that can offer enhanced precision and adaptability in forecasting bitcoin values. (Latif et al., 2023; Z. Zhang, 2023)

2.4.2 Scenarios where LSTM Outperforms ARIMA

According to the authors, LSTM surpasses ARIMA in predicting Bitcoin prices by achieving superior accuracy and successfully capturing price fluctuations. The LSTM approach exhibits a lower Mean Absolute Percentage Error (MAPE) value of less than 10 percent, suggesting superior efficacy in modeling cryptocurrency values. Moreover, the utilization of LSTM in conjunction with feature engineering has demonstrated superior performance compared to other widely used price forecasting models. LSTM networks have demonstrated superior regression performance when exclusively utilizing the close price as input, with slight enhancements obtained through wavelet decomposition. The ARIMA statistical method, in comparison, may not properly capture the intricate patterns and movements of Bitcoin prices to the same extent as the LSTM method. (Modi et al., 2023; Pasak & Jayadi, 2023)

2.4.3 Scenarios where ARIMA Outperforms LSTM

Research indicates that ARIMA surpasses LSTM in predicting Bitcoin prices by reaching a reduced Mean Absolute Percentage Error (MAPE) value of less than 10 percent. Additionally, ARIMA effectively captures price fluctuations, offering useful insights for decision-making. Furthermore, ARIMA exhibits superior predictive accuracy in comparison to LSTM and is capable of monitoring the trajectory of Bitcoin prices. In contrast, LSTM integrates sentiment analysis with deep learning to achieve superior performance in predicting Bitcoin prices when compared to alternative algorithms. Nevertheless, the predictive accuracy of LSTM is comparatively inferior to that of ARIMA, as it solely captures the trend of Bitcoin prices without capturing the direction and value within the designated timeframe. Hence, although LSTM exhibits significant advantages, ARIMA demonstrates superior accuracy and reliability in the context of Bitcoin price prediction. (Pan, 2023; Pasak & Jayadi, 2023)

2.4.4 Hybrid Models Combining LSTM and ARIMA

Research findings indicate that the integration of LSTM and ARIMA hybrid models has demonstrated superior performance compared to each individual model in the prediction of Bitcoin price. The benefits of both long short-term memory (LSTM) and autoregressive integrated moving average (ARIMA) techniques are incorporated into these models. The integration of Long Short-Term Memory (LSTM) and Autoregressive Inflation Model (ARIMA) enables the accurate representation of both long-term relationships and short-term variations in the Bitcoin price data. The hybrid models exhibit superior accuracy in out-of-sample forecasts across extended time periods as compared to traditional models, as evidenced by the experimental findings. Moreover, the hybrid models yield reduced prediction errors compared to alternative computing strategies. Hence, it is advisable to employ hybrid models that combine LSTM and ARIMA for the purpose of forecasting Bitcoin price. (Kazeminia et al., 2023; Mtiraoui et al., 2023)(Latif et al., 2023; S. Li et al., 2022; Ramani et al., 2023c)

2.5 Factors Influencing Model Performance

Research indicates that numerous factors have an impact on machine learning or time series forecasting models, such as LSTM and ARIMA models. The efficacy of these models can be substantially impacted by the quality of data preparation. The accuracy of LSTM and ARIMA models can be influenced by several external factors and hyperparameters.

The efficacy of LSTM and ARIMA models for Bitcoin price prediction is greatly influenced by data preparation. In the domain of deep learning applications, Long Short-Term

Memory (LSTM) has demonstrated notable benefits in the prediction of time-series data. However, its efficacy in conducting experiments on multi-dimensional industrial test data is suboptimal. In order to tackle this issue, a preprocessing technique is suggested that incorporates deredundancy compression, pooling processing, and noise removal. This approach chooses datasets that have strong correlation coefficients for the purpose of pooling and denoising, leading to enhanced accuracy in predictions. Furthermore, when comparing LSTM with ARIMA models, it has been seen that LSTM models, when combined with data preprocessing techniques, have a consistent ability to anticipate fluctuations in Bitcoin prices. These models surpass ARIMA models in terms of their ability to monitor trends and accurately predict both the direction and value of Bitcoin prices within specified time intervals. (Latif et al., 2023; Z. Zhang, 2023)

The performance of LSTM and ARIMA models for Bitcoin price prediction has been observed to be influenced by external influences. The association between the price of Bitcoin and market or social elements, such as market trends and sentiment analysis, has been examined in numerous studies. It has been demonstrated that the incorporation of these external correlation factors enhances the precision of LSTM predictions, resulting in a decrease in the average absolute percentage error. Furthermore, it has been shown that the LSTM model exhibits the capacity to forecast both the direction and magnitude of Bitcoin prices, surpassing the trend-tracking power of the ARIMA model. LSTM and ARIMA models used for predicting Bitcoin prices are influenced by various factors such as market sentiment, macro-economic indicators, regulatory developments, technology improvements, market liquidity, global economic conditions, adoption and integration, and market speculation. The models can use these aspects as features to encompass wider market trends and dynamics. The enhancement of model accuracy in predicting Bitcoin prices can be achieved by the monitoring and integration of pertinent information pertaining to these parameters. (Latif et al., 2023; T. Li, 2022)

Multiple research have examined the influence of hyperparameter tuning on the efficacy of LSTM and ARIMA models in predicting Bitcoin prices. In the case of LSTM, parameters such as batch size, number of neurons, and length of time sequence determine the accuracy of predictions. Moreover, the utilization of closing prices over the preceding two years, as opposed to extended time intervals, enhances the precision of predictions. In comparison to the conventional methodology, the static forecast method with model re-estimation at each step yields superior outcomes for ARIMA. In terms of trend tracking and prediction of both the direction and value of Bitcoin values, LSTM demonstrates superior performance compared to

ARIMA. The performance of LSTM models is influenced by hyperparameters such as learning rate and dropout values. Smaller values are more appropriate for daily datasets, while bigger values are more ideal for hourly datasets. In the context of Bitcoin price prediction, the optimization of LSTM and ARIMA models heavily relies on the critical process of hyperparameter tuning. (Kim & Sung, 2022)

In short, data preprocessing significantly impacts LSTM and ARIMA models for Bitcoin price prediction. External factors, like sentiment and macroeconomic indicators, enhance LSTM accuracy, reducing errors. Hyperparameter tuning, crucial for LSTM and ARIMA, improves Bitcoin price prediction by optimizing parameters and model performance.

2.6 Challenges and Future Directions

2.6.1 Challenges

Predicting the price of Bitcoin using LSTM and ARIMA encounters various obstacles. Accurate prediction of Bitcoin is challenging due to its tremendous volatility. The utilization of random data chunks for model training can lead to inaccurate outcomes, hence diminishing the accuracy of predictions. The precision of Bitcoin price forecasting is influenced by several parameters inside the Long Short-Term Memory (LSTM) model, including batch size and neuron count. Furthermore, the duration of the temporal sequence employed for prediction can have an influence on the magnitude of the prediction error. Utilizing closing prices from the preceding two years, as opposed to extended time intervals, has the potential to enhance the precision of predictions. The aforementioned problems underscore the necessity of meticulous data selection and parameter optimization in order to get precise prediction of Bitcoin prices through the utilization of LSTM and ARIMA models. (Guo et al., 2021; F. Shen, 2022)

The values of Bitcoin demonstrate significant volatility and display non-linear trends, hence posing difficulties in properly forecasting future price movements. The ability of LSTM and ARIMA models to effectively capture the intricate dynamics of Bitcoin price changes may be limited. Missing values, outliers, and irregularities are frequently observed in Bitcoin price data. The process of preprocessing the data to address these difficulties can be a time-intensive task that necessitates meticulous deliberation in order to guarantee the precision and dependability of the forecasts. The LSTM and ARIMA models necessitate meticulous selection and tuning of several hyperparameters in order to attain optimal performance. Determining the optimal set of hyperparameters can be a formidable and iterative undertaking, necessitating specialized knowledge and empirical investigation. Bitcoin is a recently introduced financial instrument, and the available historical price data may not be adequate for the development of precise prediction models. Insufficient data might cause the models to overfit or underfit, leading to inaccurate predictions. The prices of Bitcoin are subject to the influence of diverse market dynamics and external factors, including regulatory modifications, news occurrences, and investor psychology. The integration of these factors into the prediction models can present difficulties and may necessitate the utilization of supplementary data sources and the skill of feature engineering. Long Short-Term Memory (LSTM) models are frequently regarded as black-box models, hence posing challenges in comprehending and interpreting the fundamental elements that influence the predictions. Researchers and analysts may face difficulties in explaining and validating the predictions because to the absence of interpretability. (Latif et al., 2023; Z. Zhang, 2023)

To summarize, LSTM and ARIMA are effective methods for predicting time series data, such as Bitcoin prices. However, they encounter difficulties associated with the unpredictable and non-linear nature of Bitcoin prices, data preparation, model choice and adjustment, insufficient historical data, market fluctuations, and model comprehensibility. Overcoming these challenges requires careful consideration and expertise to achieve accurate and reliable predictions.

2.6.2 Future Directions

The field of Bitcoin price prediction is expected to make significant progress in the future as researchers and practitioners tackle current obstacles and take advantage of emerging technologies. Hybrid models, which combine LSTM and ARIMA, are likely to become more prominent. These models offer a combination of LSTM's ability to capture temporal dependencies and ARIMA's capabilities for forecasting time-series data. (X. Zhang et al., 2021)

Regarding data preparation, there is an anticipated shift in emphasis towards the integration of a wider range of external inputs. Enhancing model accuracy may need the incorporation of advanced sentiment research, refined global economic data, and real-time regulatory developments. The process of feature engineering and selection is expected to undergo further refinement in order to extract more pertinent information, hence facilitating the discovery of influential elements for the forecast of Bitcoin prices. (Kilimci, 2020)

It is expected that the field of deep learning architectures will progress beyond Long Short-Term Memory (LSTM) models, delving into attention mechanisms, transformers, and sophisticated recurrent neural networks. Enhancing the comprehensibility of intricate models will be essential for establishing user confidence, leading to the creation of more interpretable models that offer understanding of the reasoning behind forecasts. (Carbó & Gorjon, 2022)

Due to the scarcity of historical data pertaining to Bitcoin, forthcoming models may explore the utilization of transfer learning and meta-learning methodologies. The integration of blockchain technology is a subject that warrants further investigation. By directly analyzing on-chain data, transaction volumes, and network metrics, important insights might possibly be gained. Dynamic adaptability to real-time market changes is a significant focus, with continuous learning mechanisms and dynamic parameter modifications increasing performance in uncertain settings. (Implications, 2023)

The emergence of quantum computing has the potential to bring about significant changes, as quantum algorithms are being investigated for their ability to expedite optimization procedures, expedite model training, and enhance prediction skills. The implementation of uniform assessment metrics is intended to enhance the ability to compare and benchmark various prediction models. In addition, there is an anticipated focus on promoting open-source collaboration within the community, which will facilitate the sharing of datasets, code repositories, and joint endeavors aimed at accelerating the advancement of novel models and methodologies. (Saju et al., 2022)

In summary, the future trajectory of Bitcoin price prediction entails the amalgamation of sophisticated approaches, incorporation of external data sources, and the ongoing endeavor to develop flexible and comprehensible models that can effectively tackle the distinctive obstacles presented by cryptocurrency markets. The evolution of predictive models in the bitcoin domain will be influenced by interdisciplinary collaboration and the research of future technologies.

2.7 Summary

Bitcoin is a decentralized digital currency that was created in 2009 by someone who goes by the name Satoshi Nakamoto. It stands out because it has a limited quantity, no single owner, and is open to everyone. A limited quantity of 21 million coins stops inflation, and decentralization makes sure that there is no central control. The blockchain's cryptographic names are used for pseudonymous transactions, which protects privacy. All activities are recorded for everyone to see, so there is clear transparency. Bitcoin's price, affected by various factors, operates in a global market with high volatility. Researchers and investors who use past data analysis and pattern recognition to guess how prices will move in the future need to understand how the market works.

Various studies utilize LSTM for Bitcoin price prediction, showcasing its usefulness. For a more complete forecast, Rui Zhong uses wavelet decomposition and LSTM together (W-LSTM). Hedieh Sajedi and Masoud Arjmand use a hybrid 2D-CNNLSTM model, outperforming popular algorithms. Mr. R. Arunachalam uses LSTM in RNN to get better results. Even though it has problems like the vanishing gradient problem, LSTM is very good at recording long-term dependencies. Optimizing LSTM, using Gated units, and proposing alternative architectures like Extended LSTM handle these challenges. In many situations, like NASA's turbofan engine degradation, real-world studies show that LSTM is very good at dealing with long-term dependencies so that predictions are correct. Studies that compare ARIMA and LSTM for predicting Bitcoin prices show that LSTM is better at catching both direction and value during certain times. Studies on temperature data and seasonality diagnostics have shown that ARIMA is good at predicting short-term and steady seasonality. But ARIMA's linear assumption, sensitivity to non-stationarity, outliers, and structural changes make it hard to capture complicated patterns. This is why it works better for short-term forecasting. For predicting the price of Bitcoin, both LSTM and ARIMA models are used, but LSTM is more accurate. When you combine LSTM with mood analysis, the results are better than with other algorithms. However, LSTM's accuracy depends on things like batch number and neurons. LSTM does better than statistical methods, but it has some problems. It also lets you change the parameters in a lot of different ways. ARIMA models are great at figuring out how changes in time affect short-term Bitcoin price expectations, but they're not so good at figuring out long-term trends. ARIMA works well in some situations, but it might not work well when Bitcoin is very volatile and changes quickly. Researchers are focusing on other models, such as LSTM, for predicting the price of cryptocurrencies that change over time. Data preprocessing has a big affect on the LSTM and ARIMA models used to predict Bitcoin prices, which changes how well they work. In deep learning, LSTM is great at predicting time series, but it may not work as well in multidimensional industry tests. A suggested preparation method that uses compression, pooling, and noise removal improves the accuracy of LSTM, making it better than ARIMA at spotting trends and guessing Bitcoin's direction and value over certain time periods. LSTM and ARIMA models are affected by things outside of the models, such as market mood, economic indicators, and changes in regulations. Adding these factors as features to LSTM makes it more accurate and better than ARIMA at tracking trends. Tuning hyperparameters, especially for LSTM, is a key part of getting the best results from Bitcoin price predictions. Things like batch size and time order affect how accurate the predictions are. LSTM is better than ARIMA at predicting Bitcoin's direction and value, as well as keeping an eye on trends. This is because it can optimize its hyperparameters.

Predicting the price of Bitcoin with LSTM and ARIMA is hard because the price changes a lot and the mechanics are complicated. Handling missing numbers and outliers is part of preprocessing and needs careful thought. Choosing the right hyperparameters is very important, but there isn't a lot of past data to work with. Changes in regulations and other outside things affect predictions. In the future, there may be hybrid models, more advanced sentiment analysis, deep learning architectures that change over time, and the use of blockchain measures to improve accuracy. Quantum computing, standardized evaluation measures, and open-source collaboration could all change the field in big ways.

CHAPTER 3: RESEARCH METHODOLOGY

3.0 Introduction

The methodology utilized in our thesis about the prediction of Bitcoin values through the utilization of ARIMA and LSTM models encompasses a methodical and all-encompassing approach to forecasting cryptocurrency prices. Our methodology covers several critical components, including data collection, preprocessing, model construction, training, validation, and evaluation, with the primary purpose of boosting knowledge and prediction capacities within the realm of bitcoin trading and investment. In order to initiate our study endeavor, we initiated the crucial undertaking of data collection. The historical Bitcoin price data was rigorously obtained from Kaggle. Ensuring the dataset's integrity and quality. This initial stage established the groundwork for further examinations and endeavors in model building.

Following this, the data that was gathered underwent a thorough preprocessing procedure in order to make it suitable for modeling purposes. The process entailed doing data cleansing, filtering, and transformation to resolve prevalent problems such as missing numbers and outliers. In addition, we conducted data rescaling to standardize the data for the LSTM model. After preparing the data, we proceeded to create a model using two different approaches, ARIMA and LSTM, to predict Bitcoin values. The ARIMA model, a well-established method for analyzing time series data, was utilized to capture linear relationships and temporal trends within the dataset. In contrast, the LSTM model, which falls under the category of recurrent neural networks (RNNs), was employed due to its capacity to effectively capture intricate nonlinear associations and enduring interdependencies. After the creation of the models, the ARIMA and LSTM models were subjected to thorough training and validation procedures. The dataset was divided into groups for training, validation, and testing purposes. The performance of the trained models was assessed using various assessment measures, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R2 Score, and other relevant metrics. Through a meticulous assessment of model performance, our objective was to get a deeper understanding of the efficacy of ARIMA and LSTM models in the prediction of cryptocurrency values.

3.1 Workflow

Data Reading and Understanding → Exploratory Data Analysis and Visualization → Data Manipulation and Pre-Processing → Data Transformation and Augmentation → Building ARIMA and LSTM Models → Bitcoin Price Prediction → Evaluation and Comparison.

3.2 Dataset

I am going to use the Bitcoin dataset which is publicly available on the Kaggle website. The duration of the BTC-USD dataset is from 17/09/2014 to 25/03/2022. Prices in BTC_USD data are available in the form of US Dollars. The given data points are daily. There are a total of 2747 data points available with seven columns. The first column is about the date of each data point. The second column is open i.e., the opening price of BTC on each day. The third column is High which states the maximum price for each day. The 4^{th} column states the minimum price of each day. The 5^{th} column is about the Closing price of BTC on each day. The 6^{th} column is about the adj. closing price and the Seventh column is about the number of BTC traded every day. Therefore, this dataset provides open, high, low, and close (OHLC) data on Bitcoin. I am going to use the same dataset for both the LSTM and ARIMA models. Also, I am going to use the closing price of Bitcoin as the target variable.

3.3 Exploratory Data Analysis and Visualization

In this, I am going to explore the Bitcoin data. By using Python Notebook, we can explore facts like Information about the data and descriptions of the data. Also, I am going to find missing values, outliers, data duplicates, Nan values, and data types of columns. If there is any outlier then it is good practice to cap it rather than eliminating such records. I am going to treat missing values and Nan values based on mean, mode, and median as the case. For visualization, I am going to use both Matplotlib and Seaborn libraries. When I require an attractive and detailed chart, I will use Seaborn otherwise I will use Matplotlib.

3.3.1 Why Python Notebook?

The utilization of a Python notebook, such as Jupyter Notebook or Google Colab, for the purpose of forecasting Bitcoin time series presents a multitude of benefits. Python notebooks offer a dynamic and iterative programming environment, enabling users to engage in experimentation with diverse methods, settings, and visualizations in a sequential manner, focusing on individual cells. Python notebooks smoothly connect with a diverse range of Python libraries, such as Pandas, NumPy, Matplotlib, and Scikit-Learn. These libraries enable data manipulation, visualization, and the building of machine learning models. Notebooks possess an interactive quality that facilitates the seamless examination of Bitcoin time series data by means of interactive plots and charts. The practice of discovering patterns and trends is beneficial as it provides valuable insights for making informed decisions related to data preparation and feature engineering. Python notebooks have exceptional proficiency in documentation and collaboration, enabling users to incorporate textual content, mathematical formulae, and graphic representations in conjunction with their code. The documentation plays a vital role in ensuring repeatability and facilitating the dissemination of ideas among teams or communities. The convenience of implementing numerous time series forecasting methods, such as ARIMA, SARIMA, and LSTM, is enhanced by the integration with robust machine learning libraries like Scikit-Learn, TensorFlow, and PyTorch. The Python data science community provides substantial assistance, enabling users to access information and solutions for any issues they may have in their forecasting endeavors. In addition, the Pandas module in Python facilitates convenient data manipulation, which, when combined with the inherent reproducibility of notebooks, contributes to the enhanced reliability of Bitcoin time series forecasting models. In brief, Python notebooks are favored for Bitcoin time series analysis and forecasting due to their interactive, adaptable, and collaborative characteristics, as well as the rich library ecosystem and community support they offer. In summary, the interactive, adaptable, and cooperative characteristics of Python notebooks, in conjunction with the vast array of libraries available, render it a favored option.

3.3.2 Treatment of Missing Values

Effectively managing missing values during the process of Exploratory Data Analysis (EDA) is of utmost importance in order to derive dependable insights from a given dataset. The first phase entails the identification of missing values through the utilization of functions such as 'isnull()' or 'info()' in the Python programming language (Pandas). Graphical representations, such as heatmaps and bar charts, facilitate comprehension of the dispersion and trends of missing values across several attributes. There exist multiple ways that can be utilized to address missing values. Imputation techniques, such as the utilization of the fillna() method in Pandas, are frequently employed to address missing values by substituting them with the mean, median, or mode. Interpolation can be used to estimate missing values in time series data by leveraging the 'interpolate()' function in Pandas, which relies on existing trends. The utilization of the 'dropna()' function for removing missing values is a viable approach in cases where the number of missing values is both restricted and randomly distributed. When the data structure allows, advanced methods such as K-nearest neighbors (KNN) imputation or machine learning-based models can be utilized. Creating binary indicator variables to flag missing values preserves information regarding the absence of data. The selection of an imputation approach may be influenced by domain-specific knowledge, and the utilization of multiple imputation can be employed to address the inherent uncertainty in imputed values. In conclusion, it is imperative to thoroughly document the chosen methodology, taking into account the characteristics of the missing data and the objectives of the research. It is crucial to exercise caution regarding the potential consequences on subsequent analysis and to maintain transparency in the management of missing values during the exploratory data analysis (EDA) procedure. The optimal approach for handling missing values is contingent upon the characteristics of the data and the objectives of your research. A universally applicable solution does not exist.The Interpolation technique will be employed to address missing values.

Interpolation is commonly preferred as a method for addressing missing values in time series data due to its capacity to uphold temporal order, leverage available information, and retain the data's smoothness. It is capable of adjusting to irregular time intervals, reducing interruptions, and providing versatility through different interpolation methods. Interpolation ensures a continuous and accurate depiction of time-dependent patterns by predicting missing values based on the trend and relationships within the time series. This method gives contextually relevant imputations.

3.3.3 Treatment of Outliers

The management of outliers is an essential component of data preprocessing, as it serves to mitigate the potential impact of extreme results on the study. The process of first identification entails the utilization of visual representations, such as box plots, histograms, or statistical approaches like Z-scores or Interquartile Range (IQR), to visually represent data. Visual examination plays a pivotal role in comprehending the contextual significance of outliers, as it enables the identification of their validity as data points or their potential indication of errors.

Commence by use box plots, histograms, or scatter plots to visually represent the data and detect any possible anomalies. Quantitative identification of outliers can be facilitated through the utilization of statistical techniques such as Z-scores or the Interquartile Range (IQR). Analyze graphical depictions of the data in order to comprehend the contextual significance of outliers. Outliers can either be legitimate data points or suggest flaws in the process of data collecting. To enhance the symmetry of the distribution and mitigate the influence of outliers, one can employ mathematical modifications such as logarithmic, square root, or Box-Cox transformations. The process of truncating extreme values involves establishing a threshold at which values are classified as outliers. In addition, one can employ the technique of winsorizing, wherein extreme values are substituted with the highest or minimum non-outlying values falling within a predetermined range. Substitute outliers with a value that is more rational. One potential approach is to employ imputation techniques by utilizing neighboring data points, such as the mean, median, or a value estimated by a regression model. Data grouping or binning can mitigate the influence of outliers by classifying values into distinct ranges. This is especially advantageous for distributions that are skewed. Utilize statistical techniques such as the Z-score or interquartile range (IQR) to detect and eliminate or modify outliers. Z-score-based approaches are utilized to find data that deviate from the mean by a predetermined number of standard deviations. In order to enhance the resilience of machine learning models against outliers, it is advisable to employ robust regression models or ensemble approaches that exhibit less sensitivity towards extreme values. Utilise expertise in the field to ascertain if an outlier is a legitimate data point or a consequence of measurement inaccuracies. Seeking advice from professionals in the field can offer helpful perspectives. Perform distinct analysis with and without outliers to assess their influence on the outcomes. This aids in comprehending the degree of sensitivity of the analysis towards extreme values.The Identification approach will be employed in the treatment of outliers. The identification phase is crucial in addressing time series outliers since it provides accuracy, context, and customized tactics. The accurate identification of outliers can be achieved by the utilization of statistical approaches such as z-scores or the Interquartile Range (IQR), as well as the application of domain knowledge. The accurate identification of true extreme values and probable errors is crucial in order to distinguish between them, facilitating informed decision-making during the treatment process. The process of identification facilitates the acquisition of contextual comprehension, hence enabling the customization of treatment tactics to suit the unique characteristics of individual outliers. Graphical representations, such as box plots or time series plots, play a significant role in providing a lucid depiction of data distribution, facilitating the visual detection of anomalies and trends. The provision of this visual understanding holds significant value in facilitating informed decision-making regarding the most suitable treatment method. The identification phase is crucial in preventing overcorrection by retaining or suitably adjusting legitimate extreme values while correcting mistakes or anomalies. This enhances the preservation of the time series dataset's integrity, promotes clear and open communication, and facilitates collaborative endeavors in data analysis.

3.3.4 Treatment of Data Duplicates

Ensuring accurate and dependable insights in Exploratory Data Analysis (EDA) necessitates the careful management of data duplication. There exist multiple approaches for addressing data duplications.

Begin by looking for duplicate records in your dataset using tools like `duplicated()` in Python (Pandas). This method facilitates the identification of rows that possess similar values across all columns. The most easy technique is to eliminate duplicate rows from the dataset using the `drop_duplicates()` method in Pandas. This practice guarantees the uniqueness of each observation. In the event where duplicates are attributed to particular columns, it is possible to do deduplication by selecting a subset of columns. This feature proves advantageous in cases where specific columns have recurring values that do not necessarily indicate a general duplicate. Select whether to retain the initial or last instance of duplicate rows. This is done using the `keep` parameter in the `drop_duplicates()` method. For datasets with duplicate entries that need to be aggregated, use procedures like grouping and aggregating. This phenomenon is frequently observed in the analysis of time series data. Rather than eliminating duplicates, you have the option to introduce a new column to identify duplicate entries. In this manner, one is able to preserve data pertaining to the existence of duplicate entries inside the dataset. In cases when duplicates exhibit minor changes, it is advisable to employ fuzzy matching algorithms for the purpose of identifying and managing approximate duplicates. Analyze data visually to identify patterns or anomalies that could suggest the presence of duplicates, particularly when working with unstructured or textual data.

The removal of data duplicates is often regarded as the most effective approach for managing duplicate data in time series due to various compelling justifications. In time series analysis, it is essential to maintain the chronological order of data and eliminate duplicates to ensure temporal consistency. The removal of redundant information serves to streamline the analytical process and enhance the clarity of interpreting underlying patterns. This methodology aids in mitigating the presence of biased outcomes by mitigating the influence of skewed statistical measurements and erroneous model findings. Moreover, it improves the efficacy of machine learning models that are responsive to fluctuations in input, hence leading to enhanced precision. The elimination of duplicate entries facilitates subsequent analysis procedures, hence optimizing data acquisition for the purpose of forecasting models, trend detection, and statistical examinations. In general, eliminating duplicates guarantees the integrity and quality of data, hence minimizing the likelihood of errors in subsequent analysis.

3.3.5 Treatment of Nan Values

The management of NaN (Not a Number) or missing values is an essential component of Exploratory Data Analysis (EDA) in order to guarantee the attainment of precise and significant findings. Below are several techniques for handling NaN values:

To begin, it is necessary to detect the existence of NaN values within the dataset by employing tools such as 'isnull()' in Python (Pandas). This method will facilitate the identification of missing data within each column. To simplify the process, one can employ the 'dropna()' method in Pandas to eliminate rows or columns containing NaN values. Exercise caution when employing this approach, as it has the potential to result in the loss of vital information, particularly if a significant amount of the data is absent. It is recommended to substitute NaN values with alternative measures, such as the mean, median, or mode, corresponding to the respective column. The fillna() method might be utilized for this purpose. Imputation is a valuable technique in situations where the removal of missing data is either undesirable or impractical. In the context of time series data, the fillna() method is employed to replace NaN values with either the previous (ahead fill) or next (reverse fill) valid value in the series. Interpolation algorithms are employed to approximate missing values by leveraging the observed trend of the available data points. This is relevant for data that follows a time series or sequential pattern. Utilize imputation approaches rooted in machine learning, such as k-nearest neighbors (KNN) imputation or regression imputation, particularly when confronted with intricate interactions within the dataset. To address the issue of missing data, an alternative approach is to generate a binary indicator variable that serves as a flag. By doing so, you preserve the information regarding the absence of data in the dataset. Utilize domain-specific expertise to ascertain the optimal imputation approach. For example, substituting absent values with values that are relevant to the context.

I will employ the Imputation technique to handle Nan values, as it is a favored method in time series analysis due to its numerous benefits. It maintains the chronological sequence of observations, which is essential for retaining patterns that vary over time. Imputation utilizes the available information in the time series to generate precise estimates for missing values, hence enhancing the comprehensiveness of the dataset. In contrast to the complete elimination of NaN values, imputation serves to mitigate data loss, thereby guaranteeing a dataset that is representative for study. This comprehensiveness enables subsequent examinations, such as prediction, detection of patterns, and statistical modeling, resulting in more precise and dependable outcomes. Imputation can be applied to variations in time intervals between observations, effectively managing the intricacies of data that is sampled irregularly. Additionally, this strategy serves to address potential biases that may occur from the removal of NaN values. By imputing values based on existing data patterns, the possibility of obtaining skewed findings is reduced. Analysts have the option to select the most appropriate imputation method, such as mean imputation, linear interpolation, or regression imputation, based on the specific attributes of the time series data. In general, imputation is a highly successful approach for dealing with NaN values in time series data. It offers a thorough and contextually precise depiction of the temporal patterns existing in the dataset.

3.4 Data Pre-Processing and Transformation

3.4.1 Data Pre-Processing

The concept of data splitting in time series forecasting is of utmost importance as it entails the division of a dataset that is ordered in time into separate sets for training and testing purposes. This approach provides numerous distinct advantages that are specific to the temporal characteristics of the data. Temporal dependencies are evident in time series data, wherein the sequence of observations holds significance. The model is trained using historical data and tested on future data by dividing the data chronologically, which accurately reflects the real-world forecasting scenario. This guarantees that the model acquires knowledge about patterns and correlations from previous data, enabling it to generate forecasts for future time intervals. Time-ordered data splitting is mostly employed to prevent data leaking, which refers to the inadvertent influence of future information on the training of a model. Ensuring the prevention of data leakage is of utmost importance in order to acquire precise and authentic performance assessments. Evaluating a model's generalization capacity by testing it on unknown future observations provides a reliable assessment, revealing insights about the model's ability to accurately forecast additional, unseen time points. Furthermore, the process of data splitting allows for the refinement of model hyperparameters on the training set, guaranteeing that the model is adjusted to historical trends. The testing set functions as a realistic validation environment by representing future time points. In certain instances, it may be necessary to incorporate a validation set to provide additional model tuning and selection prior to conducting the final evaluation on the testing set.

Furthermore, it is common for time series data to display dynamic patterns throughout time as a result of trends, seasonality, or other temporal fluctuations. The process of data splitting enables the evaluation of a model's ability to adapt to changing patterns, hence enhancing the model's dependability in predicting real-world situations. Iterative and dynamic model evaluation can be facilitated through the utilization of advanced time series cross-validation techniques, such as walk-forward validation or rolling origin validation. These methodologies iteratively enhance the training and testing datasets, capturing dynamic trends and guaranteeing a comprehensive evaluation of model efficacy. Data splitting is crucial in time series forecasting as it helps maintain the temporal structure, simulate real-world deployment circumstances, avoid data leaking, optimize model hyperparameters, assess robustness to shifting patterns, and utilize advanced cross-validation techniques. The implementation of this method is essential in the development of precise and dependable time series forecasting models.

I am going to split the data sequentially not random. Because it is time series data in which the next data point is related to the previous one. We can split data as 60/40, 70/30, 80/20, or 90/10. I would prefer an 80/20 split. Where training data is 80% and 20% is testing.

3.4.2 Data Transformation

A variety of strategies are utilized to alter the structure or distribution of data, with the aim of enhancing its suitability for statistical analysis, machine learning, and other tasks that rely on data. These strategies are crucial in preparing data for diverse applications, boosting its quality, interpretability, and applicability for specific algorithms. Typical methods of transformation include:

The process of normalization entails the scaling of numerical properties to a specified range, commonly ranging from 0 to 1. Normalization is a process that removes variances in the magnitudes of variables, which is essential for algorithms that are sensitive to variations in scale. Z-score normalization, also known as standardization, involves the transformation of numerical data to possess a mean of 0 and a standard deviation of 1. This approach becomes advantageous for algorithms that make the assumption of a normal distribution, as it guarantees that variables may be compared using a standardized scale. Log transformation is a statistical technique that is commonly used to analyze data that exhibits exponential growth or skewed distributions. Its purpose is to stabilize variance and make correlations more linear. The Box-Cox Transformation is a collection of power transformations, which encompasses the implementation of logarithm as a specific instance. The process of stabilizing variance aids in the approximation of data to a normal distribution.

Polynomial transformation is a technique used in polynomial regression models to generate higher-degree polynomial features that can effectively capture non-linear interactions.

The process of categorizing continuous data into intervals or bins, which serves to simplify intricate relationships and improve the comprehensibility of the data. The process of transforming category data into numerical representations that are appropriate for machine learning. Methods encompass one-hot encoding or label encoding, which are determined by the features of the data. Feature Scaling: It is crucial for algorithms that utilize distance metrics or gradient descent optimization to ensure that numerical features have a consistent scale. The utilization of smoothing techniques, such as moving averages or other methods, serves to mitigate noise and accentuate patterns within time series data. Aggregation refers to the process of merging data at a higher level, such as daily to monthly, in order to decrease the amount of detail and capture more general patterns. principle Component Analysis (PCA) is a method used to reduce the dimensionality of data by converting it into uncorrelated variables called principle components, while still preserving important information. Winsorizing is a technique that limits the impact of extreme values by setting a threshold, without completely removing them.

The selection of these procedures is contingent upon the nature of the data and the specific analytical or modeling needs. Issues such as scale disparities, non-linearity, noise, and outliers are effectively tackled, hence enhancing the data's appropriateness for following analytical procedures. The utilization of suitable transformation techniques enhances the resilience, precision, and comprehensibility of outcomes in statistical examinations or machine learning models.

The utilization of min-max scaling is a prevalent method employed in the field of machine learning, particularly in LSTM (Long Short-Term Memory) models, for the purpose of normalizing input data. The process of normalization holds significance as it contributes to the enhancement of neural network performance and training stability. I will employ min-max scaling due to its capacity to provide a uniform input range, expedite convergence, ensure numerical stability, enhance generalization, and establish more accurate weight initialization.

LSTMs require a three-dimensional input to represent the input data. The dimensions commonly associated with this three-dimensional shape are the number of samples, time steps, and number of features. It is important to transform the two-dimensional data, such as a time series or a sequence of observations, into the desired three-dimensional format through the process of data reshaping. I will utilize Python functions such as 'reshape'.

3.5 Time Series Forecasting Models

3.5.1 SARIMAX Model

SARIMAX is an advanced time series forecasting model that enhances the capabilities of the standard SARIMA model by include exogenous variables. The primary purpose of SARIMA models is to effectively capture and predict the seasonal patterns and trends observed in time series data. In contrast, SARIMAX models provide an enhanced capability to incorporate the influence of external predictors or exogenous factors into the analysis.

Breakdown of the components of SARIMAX:

1. Seasonal (S), AutoRegressive (AR), Integrated (I), and Moving Average (MA) Components:

Like SARIMA, SARIMAX includes these components to model the underlying structure of the time series. These components capture the seasonality, trend, and autocorrelation in the data.

2. Exogenous (X) Factors:

The term "X" in the SARIMAX framework denotes exogenous variables, which encompass external factors or predictors capable of exerting an influence on the time series. The aforementioned variables are not inherent to the time series data, but rather are taken into account in conjunction with the intrinsic components in order to enhance the precision of forecasting.

The SARIMAX model is typically represented as SARIMAX(p, d, q)(P, D, Q, s) + X, where p, d, and q denote the dimensions of the non-seasonal autoregressive, differencing, and moving average components, respectively. The order of the seasonal autoregressive, differencing, and moving average components, as well as the length of the seasonal cycle, are denoted as P, D, Q, and s, respectively. The variable X represents the exogenous factors.

The use of exogenous variables in SARIMAX enables the integration of supplementary information that has the potential to enhance the accuracy of forecasts. In the context of sales data forecasting, an exogenous variable refers to advertising expenditures or promotional activities that exert an influence on sales but are not intrinsically integrated into the time series.

SARIMAX models are exceptionally advantageous in the analysis of time series data that demonstrates the presence of both seasonal patterns and exogenous factors. They offer a versatile structure for recording intricate patterns and connections, rendering them appropriate for a diverse array of forecasting applications.

It is imperative to carefully choose suitable values for the model parameters (p, d, q, P, D, Q, s) while utilizing SARIMAX, taking into consideration the specific attributes of the data. The process of model training entails the estimation of parameters and coefficients, which are subsequently utilized to make predictions about future observations within the time series.

3.5.2 Fb Prophet Model

Prophet is a time series forecasting tool that has been developed by Facebook and is open-source in nature. The system is specifically engineered to effectively manage daily observations that exhibit patterns across many temporal scales, including annual, weekly, and daily variations. Prophet is especially suitable for datasets that exhibit pronounced seasonal patterns, many instances of seasonality, and occurrences of holidays. The Prophet model encompasses several prominent features and components.

1. Additive Components: The Prophet model is a statistical technique that breaks down a time series into three primary additive components: trend, seasonality, and holidays. The trend component encompasses enduring shifts over an extended period, whereas seasonality encompasses periodic variations, and holidays encompass exceptional occurrences or occasions..

2. Automatic Detection of Holidays: The Prophet model autonomously identifies and incorporates holidays into its forecasting. Individuals have the ability to furnish a personalized compilation of holidays that are pertinent to their dataset. Prophet, in turn, regards these dates as prospective factors that may influence the time series.

3. Multiple Seasonality: Prophet has the capability to effectively manage numerous seasonalities, hence offering advantages for datasets characterized by intricate patterns that manifest distinct periodicities. One such approach is to simultaneously model both weekly and yearly seasonality.

4. Saturating Minimum: The proposed model has a saturating minimum parameter for the growth component, enabling the forecast to incorporate scenarios in which a time series may experience saturation or attain a minimum value.

5. Handling Missing Data: Prophet exhibits resilience to the presence of missing data and outliers. The algorithm is capable of effectively managing datasets that have irregular gaps or missing values, eliminating the need for imputation or preprocessing.

6. Flexibility in Holiday Effects: Users have the ability to customize festive effects according to their preferences. This feature enables the adjustment of the influence of holidays on the prediction.

7. Scalability: Prophet is specifically engineered to possess scalability and efficiency, rendering it well-suited for handling extensive datasets including numerous observations.

8. Intuitive Interface: Prophet offers an intuitive interface, ensuring that users with different degrees of proficiency in time series forecasting can easily utilize it. The syntax is straightforward and comparable to other data analysis tools, enhancing user-friendliness.

Prophet is commonly utilized by customers who supply a dataframe consisting of two columns: "ds" (representing the date or timestamp column) and "y" (representing the target variable to be predicted). Subsequently, the model is fitted to the historical data, enabling the generation of forecasts for subsequent time points.

Prophet has become famous due to its capacity to generate precise and comprehensible predictions with minimal human involvement. This approach has notable efficacy in situations when the time series displays pronounced seasonal trends and is subject to the impact of vacations.

3.5.3 ARIMA Model

The AutoRegressive Integrated Moving Average (ARIMA) model is a commonly employed and robust time series forecasting technique. The purpose of this system is to effectively collect and forecast trends within time series data, rendering it highly advantageous for various applications like financial market predictions, sales forecasting, and weather forecasting.

Breakdown of the key components of the ARIMA model:

1. AutoRegressive (ARThe autoregressive (AR) component is responsible for capturing the association between the present observation and its previous values. The word "autoregressive" denotes that the model does regression analysis by subtracting its own prior values from the current value.

2. Integrated (I) Component: The I component denotes the process of differencing, wherein each subsequent observation is subtracted from its preceding one. The purpose of this differencing step is to achieve stationarity in the time series, ensuring that its statistical characteristics, such as mean and variance, remain consistent over the time period. The concept of stationarity holds significant importance in the precise modeling of time series data.

3. Moving Average (MA) Component: The moving average (MA) component examines the relationship between the present observation and a residual error derived from a moving average model that is applied to previous observations. This aids in capturing transient variations or interference in the time series.

The notation for an ARIMA model is expressed as ARIMA(p, d, q), where:

p is the order of the autoregressive component (AR),

d is the degree of differencing required to achieve stationarity (I),

q is the order of the moving average component (MA).

Building an ARIMA model requires choosing p, d, and q based on time series data. Analysis of autocorrelation and partial autocorrelation functions is common..

ARIMA models capture linear time series trends and patterns well. They may struggle with complex patterns, non-linear correlations, or multi-seasonal data. Extended models like Seasonal ARIMA (SARIMA) or Prophet or machine learning models may be used in such instances.

ARIMA is a fundamental time series forecasting model due to its interpretability, ease of implementation, and practicality.

I chose the ARIMA model because of its Flexibility, Simple Interpretability, Effective handling of trends and seasonability, well-established statistical principles, forecasting accuracy, and availability of software.

3.5.4 LSTM Model

LSTM is a recurrent neural network framework component. Traditional recurrent neural networks struggle to capture and learn long-span dependencies in cyclic data. LSTM models do.

Key features of LSTM models:

1. Memory Cells: LSTMs have memory cells to store and update data. Capturing long-term interdependence in time series data requires this skill.

2. Gates: Long Short-Term Memory (LSTM) models employ three distinct types of gates, namely the input gate, forget gate, and output gate, to regulate the transmission of information within the memory cell. These gates regulate the information that enters and exits the cell, allowing the model to learn when to recall or forget specific information.

3.Learning Short and Long-Term Patterns: LSTMs demonstrate exceptional proficiency in acquiring both immediate and enduring patterns in time series data. The retention of information across extended sequences by memory cells allows the model to effectively depict intricate temporal connections.

4. Sequence-to-Sequence Modeling: Long Short-Term Memory (LSTM) models has the ability to effectively handle input sequences and produce output sequences, rendering them well-suited for sequence-to-sequence applications, such as time series forecasting. By utilizing previous data as input, the model is capable of making predictions for future values within the sequence.

5.Parallel Processing: LSTMs have the ability to process input in parallel, which improves their efficiency in comparison to conventional RNNs. This is accomplished by employing numerous memory cells that function simultaneously.

6. Bidirectional LSTMsIn addition to conventional Long Short-Term Memory (LSTM) models, bidirectional LSTMs are capable of processing input sequences in both forward and backward directions. This characteristic allows the model to incorporate contextual information from both previous and future observations, which has the potential to enhance the accuracy of predicting.

When employing Long Short-Term Memory (LSTM) models for the purpose of time series forecasting, the model undergoes training using past data, and subsequently utilizes the acquired patterns to provide forecasts for forthcoming time intervals. Optimizing performance generally requires hyperparameter tweaking, which involves altering the number of layers, units, and training epochs.

The effectiveness of LSTMs in capturing intricate relationships in time series data relies on the unique attributes of the dataset and the meticulous choice of model parameters. It is noteworthy to mention that recent developments in neural networks, such as Transformers, have demonstrated potential in the domain of time series forecasting and are garnering significant interest among the academic community.

I chose the LSTM Model for Bitcoin prediction because it captures long-term dependencies, handles exploding gradient problem, learns hidden patterns efficiently, does parallel processing, adapt different time lags, handles irregularity of sample effectively, does not require manual featuring, and can handle multivariate features. So, I prefer the LSTM model rather than FbProphet, and other time series forecasting algorithms.

3.6 Evaluation Metrics

When evaluating LSTM (Long Short-Term Memory) models, ARIMA (Auto-Regressive Integrated Moving Average), or any time series forecasting algorithm, it's essential to use appropriate metrics to assess their performance. Some commonly used evaluation metrics for Time series forecasting models are:

- **Mean Absolute Error :** Represents the avg. absolute diff. between the predicted values and the actual values.
- **Mean Squared Error :** Measures the avg. of the squared gap between predicted and actual values.
- **Root Mean Squared Error :** It is the sqrt of the MSE and provides a measure of the avg. of the errors in the units as the original data.
- **Mean Absolute Percentage Error :** It expresses the average per. diff. between the forecasted and actual values.
- **R2 Score :** It measures the proportion of the variance in the target variable (target) that is predictable from the independent variables (predictions).

I am going to compare the ARIMA and LSTM Models based on mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MSPE), and based on the accuracy of the model. Because they are commonly used metrics that can be employed to assess the performance of time series predicting algorithms. It is always good to use multiple metrics for better understanding and better comparison between two algorithms. So, I decided to use multiple evaluation metrics for comparing models.

3.7 Summary

The way we used ARIMA and LSTM models in our study for our Bitcoin price prediction thesis is set up to give us a complete way to predict cryptocurrency prices. It starts with carefully gathering data. For example, past Bitcoin price data can be found on Kaggle. The next step is preprocessing, which includes steps like cleaning, screening, and transforming the data to fix any problems or mistakes in the dataset. Feature engineering techniques are also used to find important predictors that may help our models be better at making predictions.

When it comes to building models, we will use two different approaches: ARIMA and LSTM. ARIMA is a classic method for analyzing time series data that finds linear dependencies and temporal trends. LSTM is a type of recurrent neural network that is very good at recognizing long-term dependencies and complicated, nonlinear relationships. We want to use the best parts of each of these methods to make our predictions more accurate and reliable in the constantly changing cryptocurrency market.

The next steps are to train, validate, and evaluate the model. There are training, validation, and testing sets in the bitcoin collection. To check how accurate and reliable a model is, evaluation measures like Mean Absolute Error (MAE), R2 score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are used.

CHAPTER 4: ANALYSIS

4.1 Introduction

In this analysis chapter, we delve into the heart of our research methodology. The experimental setup serves as the bridge between theory and practice, allowing us to rigorously evaluate the performance of our chosen models ARIMA and LSTM in the context of Bitcoin price prediction.

We meticulously curate historical Bitcoin price data, ensuring its quality, consistency, and relevance. Cleaning noisy data, handling missing values, and normalizing features are essential steps. Extracting meaningful features from raw data is crucial. We explore lagged price values, trading volumes and other relevant indicators. We split the dataset into training and test sets. The ARIMA and LSTM models undergo rigorous training, fine-tuning hyperparameters, and cross-validation. Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE) and R2 Score are our compass to evaluate both models. These metrics guide us in assessing model accuracy and robustness. We establish baseline performance using simple moving averages and historical averages. These benchmarks provide context for evaluating our advanced models.

ARIMA vs. LSTM: We anticipate that LSTM, with its ability to capture long-term dependencies, will outperform ARIMA in terms of predictive accuracy Impact of Data Granularity: We hypothesize that using daily data will yield better results compared to weekly or monthly data. We explore how variations in model parameters impact performance. Robustness to noise and overfitting are critical considerations.

4.2 Data Understanding

I used the Bitcoin dataset which is publicly available on the Kaggle website. The duration of the BTC-USD dataset is from 17/09/2014 to 25/03/2022. Prices in BTC_USD data are available in the form of US Dollars. The given data points are daily. There are a total of 2747 data points available with seven columns. The first column is about the date of each data point. The second column is open i.e., the opening price of BTC on each day. The third column is High which states the maximum price for each day. The 4th column states the minimum price of each day. The 5th column is about the Closing price of BTC on each day. The 6th column is about the adj. closing price and the Seventh column is about the number of BTC traded every day. Therefore, this dataset provides open, high, low, and close (OHLC) data on Bitcoin. I used the same dataset for both the LSTM and ARIMA models. Also, I am going to use the Adjusted closing price of Bitcoin as the target variable. The head and of the data are shown in Fig. 4.1.

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	21056800
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	34483200
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	37919700
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	36863600
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	26580100
...	...	...	...	...	...	...	...
2742	2022-03-21	41246.132813	41454.410156	40668.042969	41077.996094	41077.996094	24615543271
2743	2022-03-22	41074.105469	43124.707031	40948.281250	42358.808594	42358.808594	32004652376
2744	2022-03-23	42364.378906	42893.507813	41877.507813	42892.957031	42892.957031	25242943069
2745	2022-03-24	42886.652344	44131.855469	42726.164063	43960.933594	43960.933594	31042992291
2746	2022-03-25	43958.675781	44982.519531	43711.871094	44395.964844	44395.964844	30379415552

Fig. 4.1- Head and Tail of the Data

In Fig. 4.1 Date column Represents the specific date for each row of data. Open column Indicates the opening price of the stock on that date. High Denotes the highest price reached during the trading day. Low Represents the lowest price observed during the same day. Close column Refers to the closing price at the end of the trading session. Adjusted Close is Adjusted closing price (considering factors like dividends and stock splits) and Volume Indicates the total number of shares traded on that date.

After calling the function info() we get the information about each column type and null values. From Fig. 4.2 we can see that there are total 2747 data points. There is no any null value

```
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       2747 non-null   datetime64[ns]
 1   Open       2747 non-null   float64
 2   High       2747 non-null   float64
 3   Low        2747 non-null   float64
 4   Close      2747 non-null   float64
 5   Adj Close  2747 non-null   float64
 6   Volume     2747 non-null   int64
dtypes: datetime64[ns](1), float64(5), int64(1)
memory usage: 150.4 KB
```

Fig. 4.2 – Data Information

in given data. Data type of date column is datetime64. Column type of Open, High, Low, Close, Adj Close is float64. And data type of last column that is Volume is int64.

4.3 Data Analysis

Box plot is used to find outliers if any. First, I draw the box plot of all six columns.

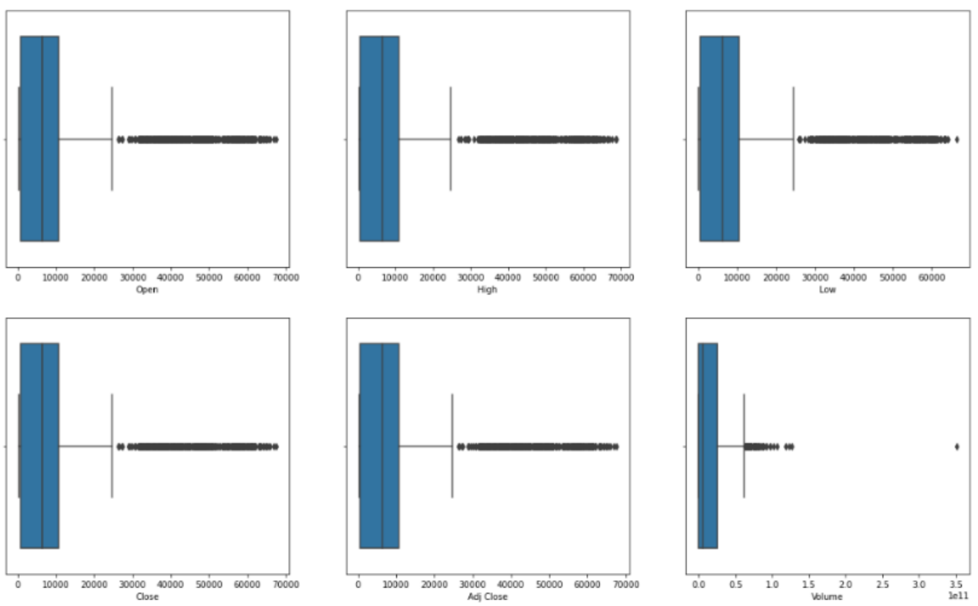
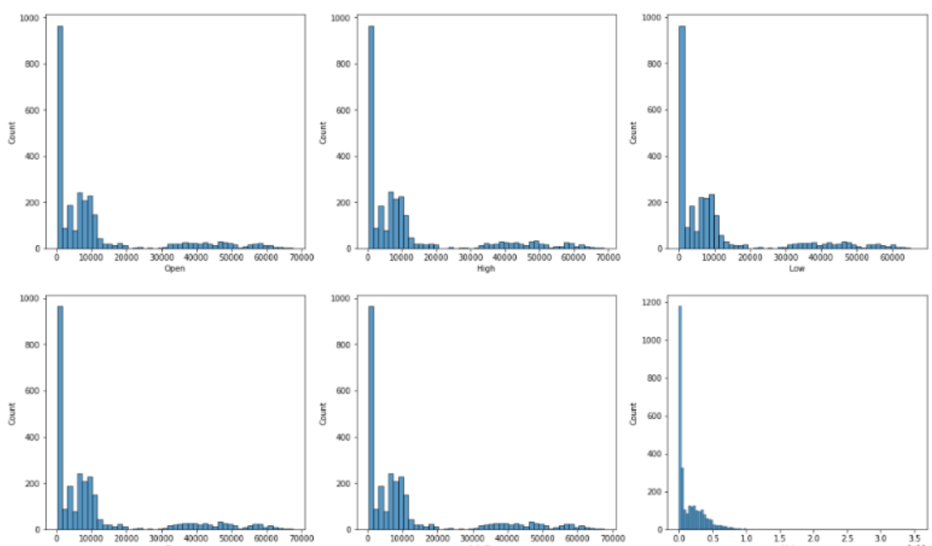


Fig. 4.3 – Box Plot of All column

From the Fig. 4.3 we can say that most of the datapoints are found between minimum score to maximum score. Very few data points were out of the box plot and those are on upper side of the box plot.



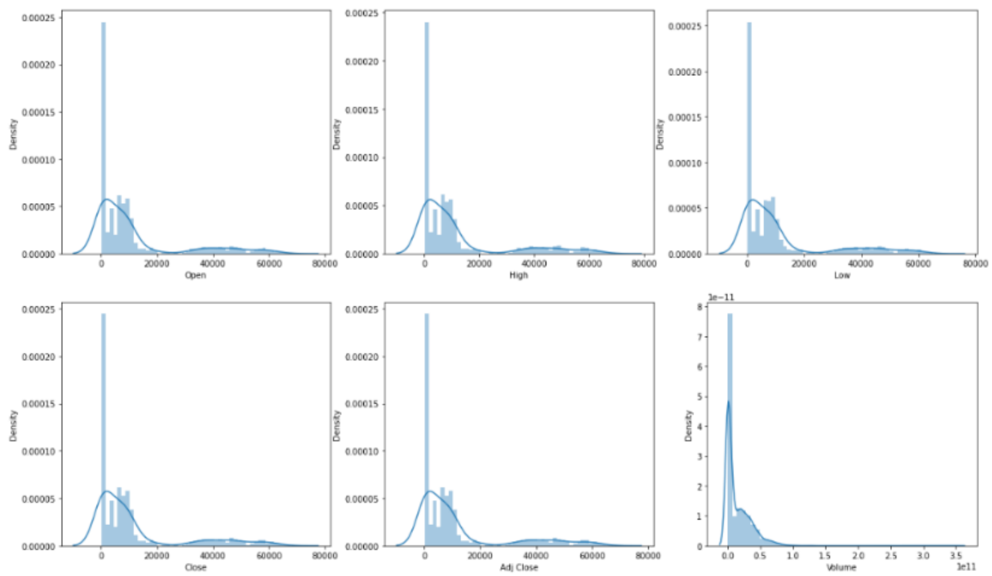


Fig. 4.4 – Hist & Dist Plot of all Column

From the Fig. 4.4 Shows density of each data point of bitcoin data. From fig. 4.4 we can say that most of the data point or bitcoin prices are between 0 to 15000. That mean most of the data points occurred between 0 to 15000 USD. Data points above 20000 are very less when we compare it with data point below 20000**.**

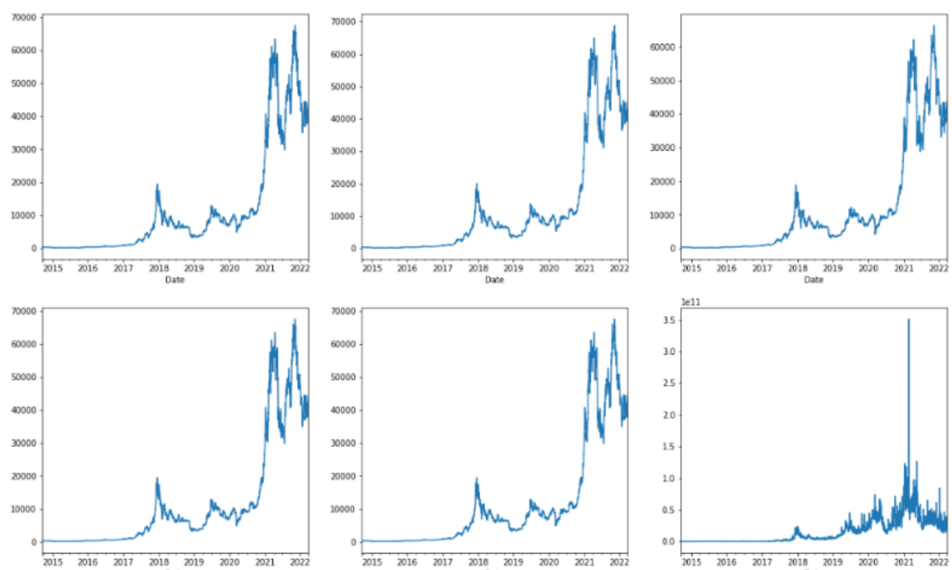


Fig. 4.5 Line Chart of all Column

From Fig. 4.5 shows line chart of all column. Fig.4.5 shows that there is lot of fluctuation on bitcoin prices. It shows that bitcoin prices till year 2018 were more stable. From the year 2018 to year 2021 there was normal fluctuation in bit coin prices. But from 2021 onwards bitcoin prices were fluctuated drastically. And when we compare price fluctuation with volume fluctuation then it comparatively low.

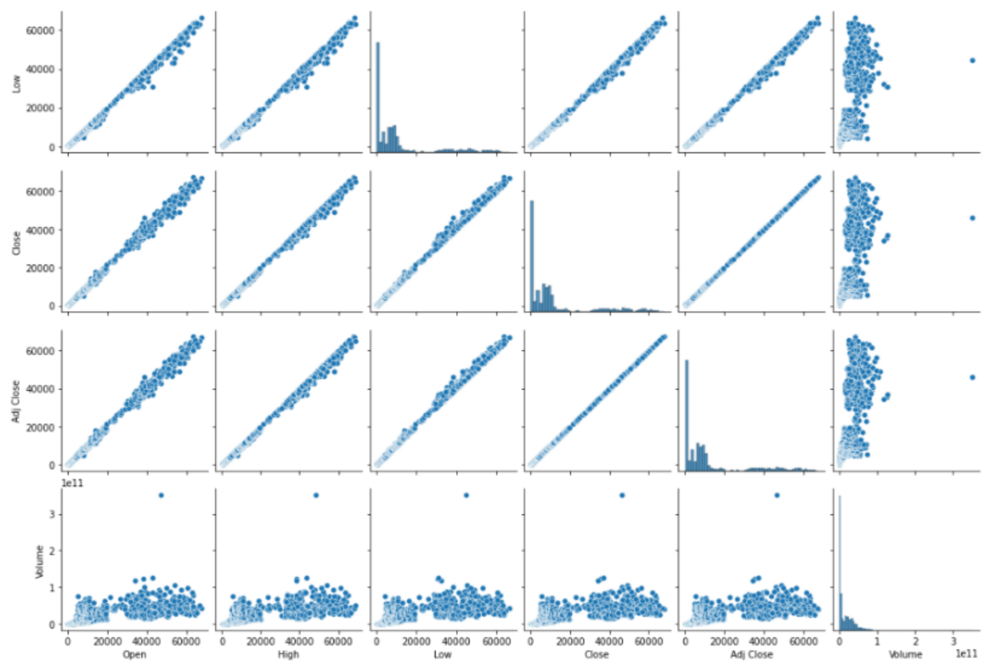


Fig. 4.6 – Scatter Plot of all Column

From the fig. 4.6 we can say that there is strong positive relation between all column except volume column. Scatter plot shows co relation between numerical data points. Left to right upward direction of scatter plot means there is strong positive relation between to variables. Left to right downward direction od scatter plot means strong negative co relation between two variables.

On the other hand, heat map also shows co relation between two numerical variables. In fig. 4.7 we can notice two colour shade. Dark green means there is strong relation to each other. All variables except volume shows annot value of one. That means they are strongly dependant to each other. Relation between price and volume is not strong as others but still it is good.

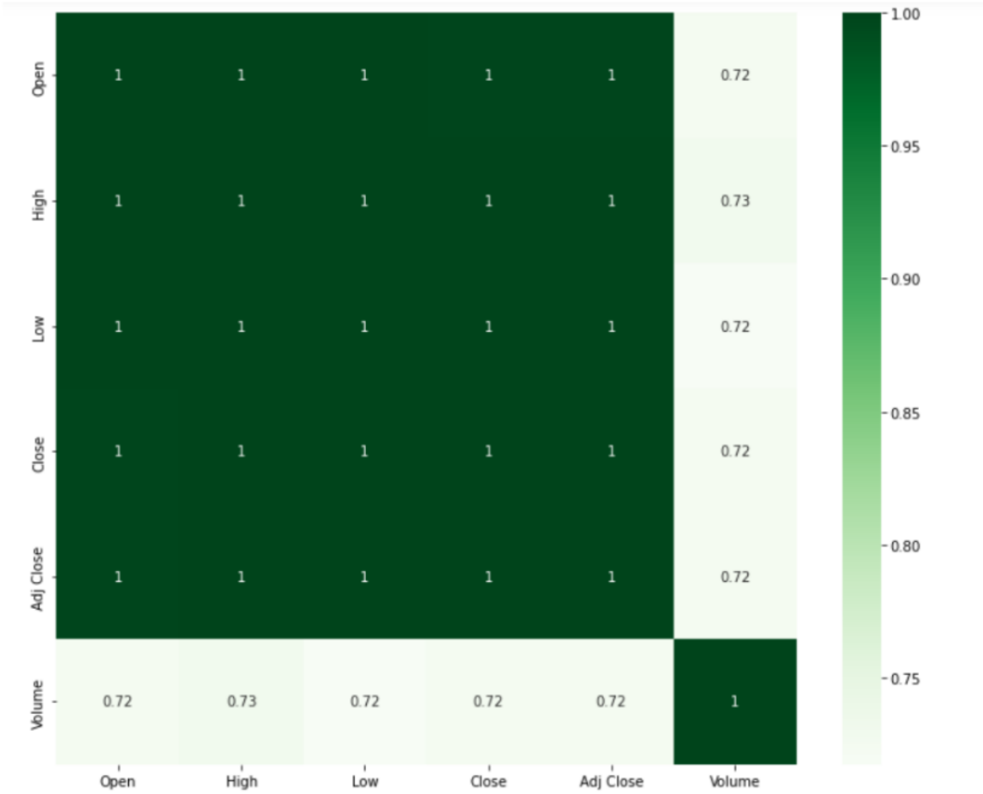


Fig. 4.7 – Heat Map of all Column

From the fig 4.8, The left y-axis represents the price, ranging from 0 to approximately 70,000. The right y-axis represents the trading volume (multiplied by 10^6), ranging from 0 to approximately 300,000 10^6. The x-axis represents the time period from “2014 Sep17” to “2021 Jul22.”

Price Remained relatively stable until around March 2020. Experienced a significant increase, peaking at just under 70,000. After the peak, the price became more volatile with noticeable fluctuations. Volume Shows several spikes but has a significant increase around July 22, 2021.

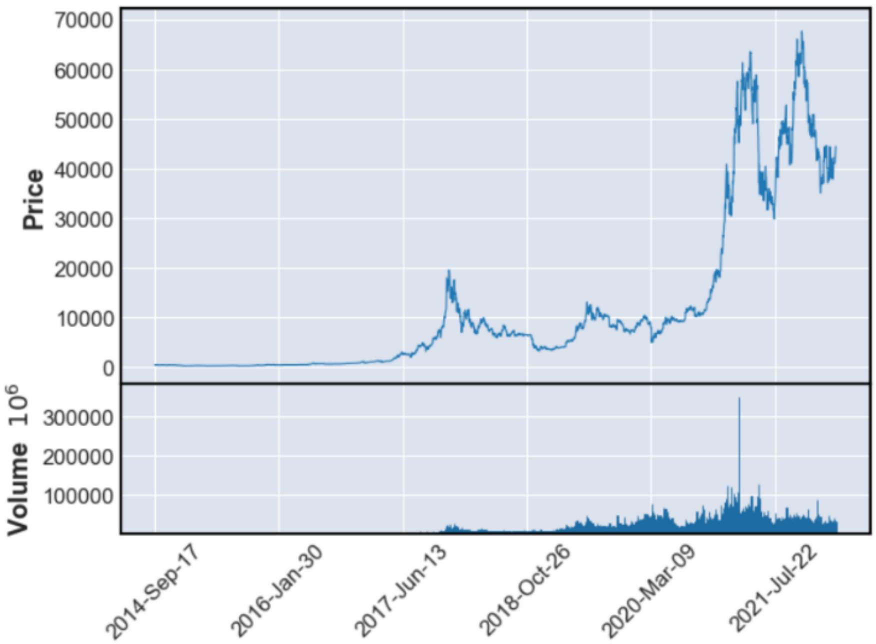


Fig. 4.8 – Price and Volume

Fig. 4.9 – Candle of Price and Volume

Fig. 4.9 shows candle chart of volume and price column on the basis of date time. There is total five candle chart of all bitcoin data. In fig. 4.9 From the fig 4.8, The left y-axis represents the price, ranging from 0 to approximately 70,000. The right y-axis represents the trading volume (multiplied by 10^6), ranging from 0 to approximately 300,000 10^6. The x-axis represents the time period from “2014 Sep17” to “2021 Jul22.”

Red candle shows downward trend in the bitcoin price for a single day. A red candlestick indicates that the stock or asset closed lower on that day compared to the previous day. It represents a bearish trend, suggesting downward price movements. Sellers dominated, causing the price to decline during that trading period.

A green candlestick indicates that the stock or asset closed higher on that day compared to the previous day. It represents a bullish trend, suggesting upward price movements. Buyers were in control, pushing the price higher during that trading period.

Fig. 4.10 shows there is no relation between close price and volume.

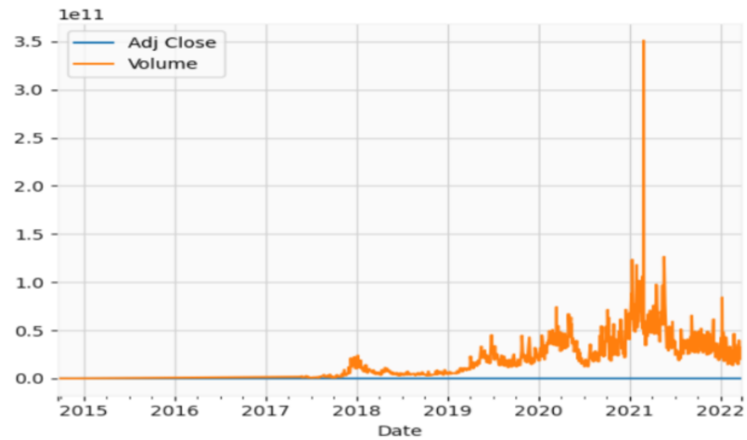


Fig. 4.10 – Line Chart without Normalise

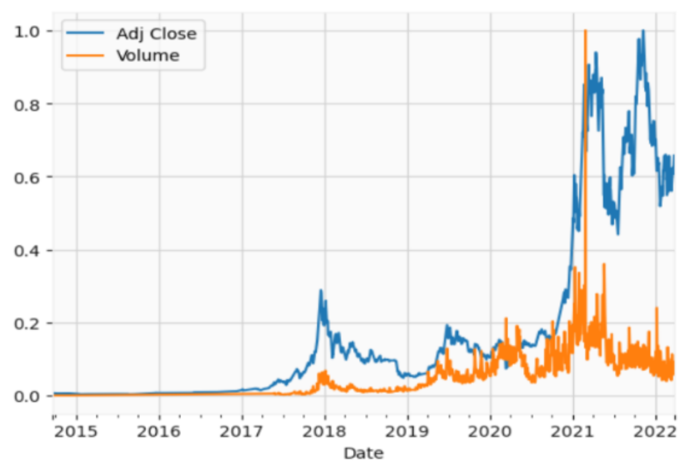


Fig. 4.11 – Line Chart with Normalise

From fig. 4.11 We can say that there is relation between volume and close price. When we plot volume and closing price directly then it shows direct line of closing price. Because amount of volume is much larger than Adjusted closing price. But when we normalise both variable between zero to one then we can say that prices increase with increase in volume or closing prices decrease with volume.

Fig. 4.12 is a scatter plot of adjusted close price and volume. From the fig. 4.12 we can say there is no strong relation between volume and adjusted closing price.

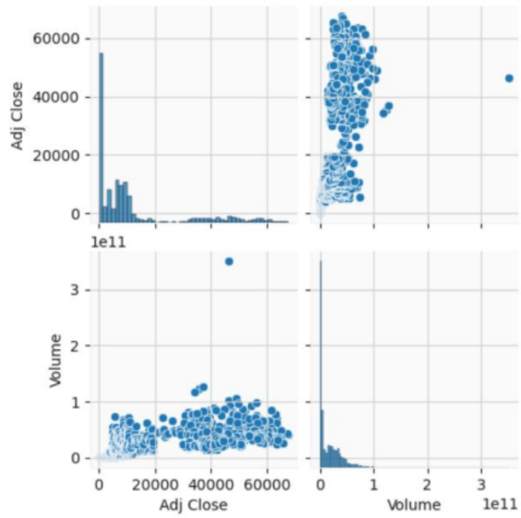


Fig. 4.12 – Scatter Plot of Volume & Price with Outlier

In fig. 4.13 we removed volume outlier. After removing outlier volume and adjusted price shows positive relation. It means price increase with increase in volume.

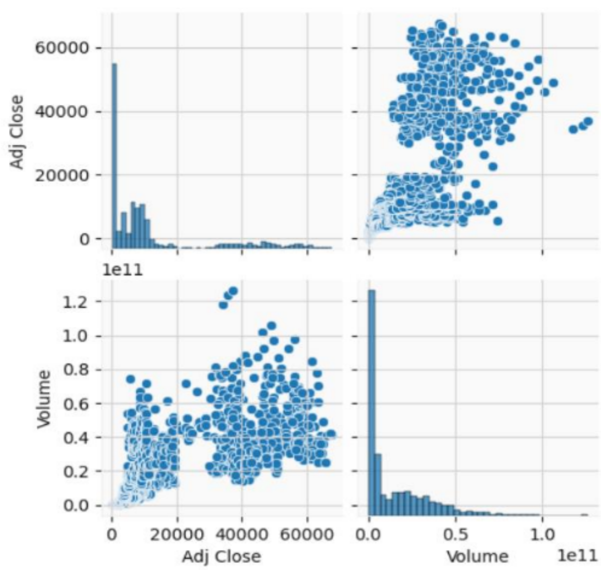


Fig. 4.13 – Scatter Plot of Volume & Price without Outlier

Fig. 4.14 is heat map of adjusted close and volume. We can see that before removing volume outlier the corelation coefficient was 0.72. but after removing one outlier from volume variable, it shows corelation coefficient of 0.75. It means dependency on volume increase.

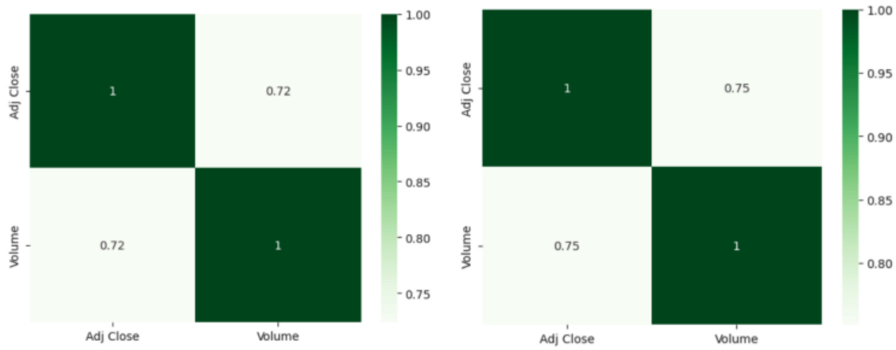


Fig. 4.14 – Heat Map of Volume & Close

Fig. 4.15 is line chart of average adjusted closing price on the basis of year. We can say that there is drastic change in price from year 2020 to year 2021. And after that it starts decline.

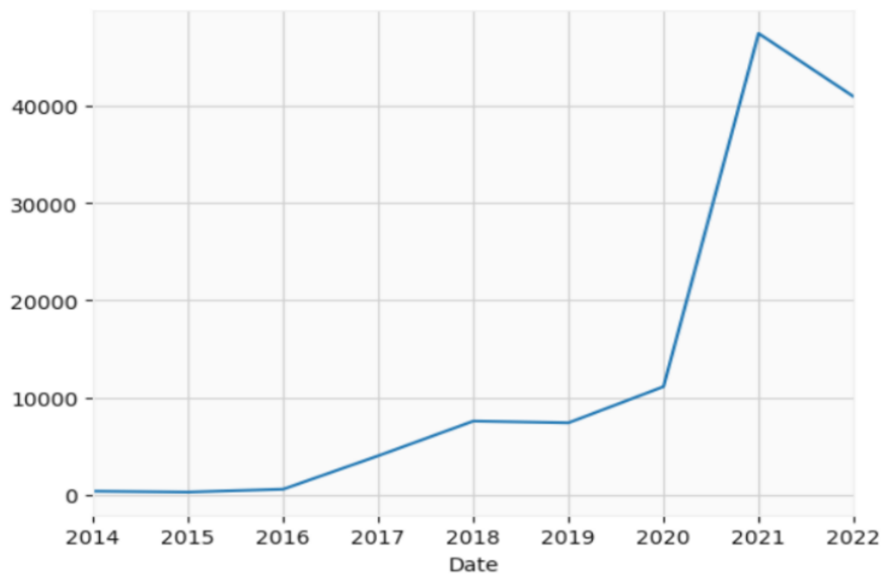


Fig. 4.15 Yearly Closing Prices

Fig. 4.16 is line chart of average volume on the basis of year. We can say that there is drastic change in volume of bitcoin from year 2029 to year 2021. And after year 2021 it starts decline.

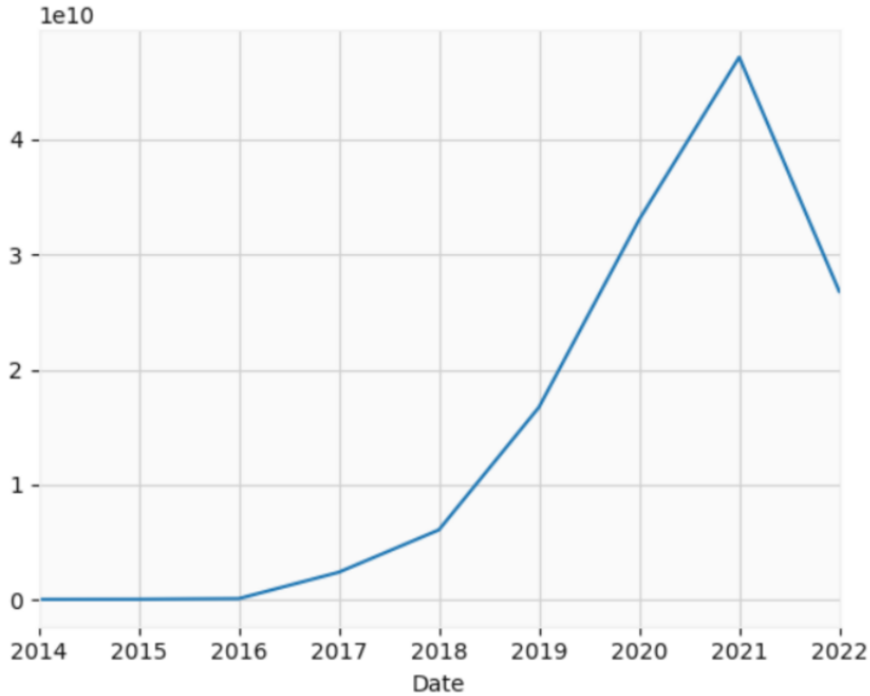


Fig. 4.16 – Yearly Volume

In fig. 4.17 the y-axis represents the adjusted closing price, ranging from 0 to approximately 70,000. The x-axis represents the time period from “2014 Sep17” to “2021 Jul22.” It shows average of closing price for each year. From the figure we can say that average price for year 2021 and year 2022 is highest than other year.

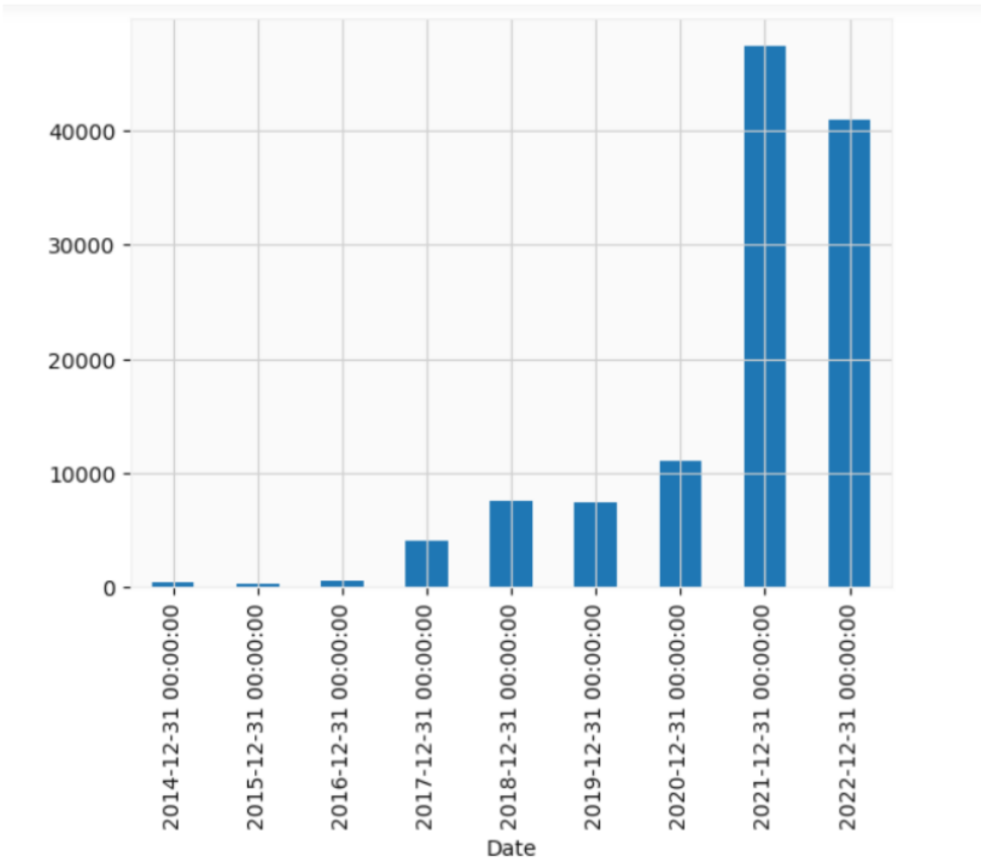


Fig. 4.17 – Average Price Bar Chart

Closing price for year 2014 to year 2019 was less than 10000. But after that it increase drastically. In year 2021 it touched to highest point. But after year 2021 it starts decreasing. And closing price reach to 41000 from 48000 between year 2021 & 2022.

In fig. 4.18 the y-axis represents the adjusted closing price, ranging from 0 to approximately 70,000. The x-axis represents the month wise time period from “2014 Sep” to “2022 Mar.” It shows average of closing price for each month. From the figure we can say that average price from the month 11 of year 2021 is highest than previous months.

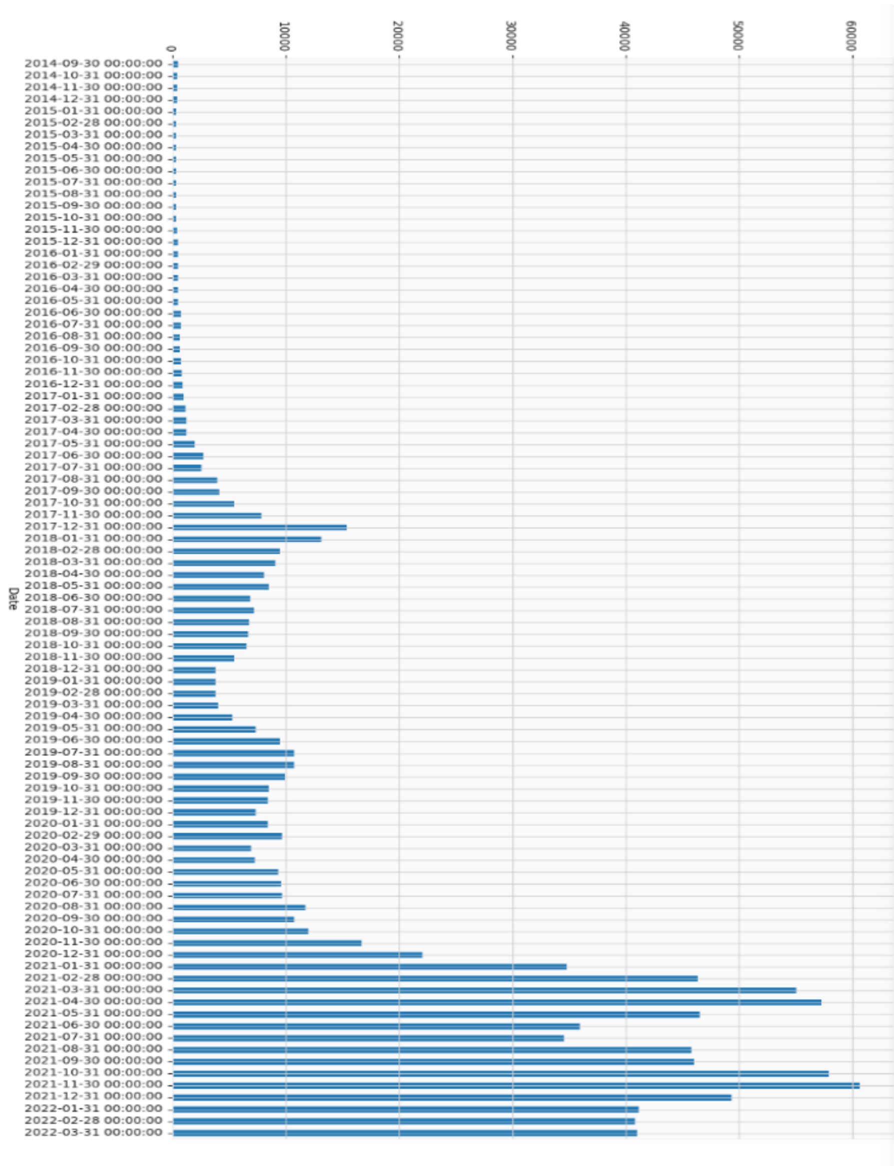


Fig. 4.18 – Average Price Bar Chart Monthly

4.4 Data Preparation

4.4.1 Train Test Split

There are total 2747 data points are available of bitcoin price. Out of 2747 data point I use first 80 percent data point as training data. That is 2197. And last 20 percent data point were use as testing data point which shows in fig. 4.19 and fig. 4.20. There are total 550 testing points.

```
In [51]: training_data = list(bit_usd_close[:to_row])
         training_data

Out[51]: [457.334015,
          424.440002,
          394.79599,
          408.903992,
          398.821014,
          402.152008,
          435.790985,
          423.204987,
          411.574005,
          404.424988,
          399.519989,
          377.181,
          375.46701,
          386.944,
          383.61499,
          375.071991,
          359.511993,
          328.865997,
          320.51001,

In [52]: len(training_data)

Out[52]: 2197
```

Fig. 4.19 – Training Data

```
In [53]: testing_data = list(bit_usd_close[to_row:])
         testing_data

Out[53]: [10538.459961,
          10246.186523,
          10760.066406,
          10692.716797,
          10750.723633,
          10775.269531,
          10709.652344,
          10844.640625,
          10784.491211,
          10619.452148,
          10575.974609,
          10549.329102,
          10669.583008,
          10793.339844,
          10604.40625,
          10668.96875,
          10915.685547,
          11064.458008,
          11296.361328,

In [54]: len(testing_data)

Out[54]: 550
```

Fig. 4.20 - Testing Data

In fig. 4.21 X-Axis (Dates): Represents the time period from 2015 to 2022. Y-Axis (Closing Prices): Indicates the value the bitcoin closing prices at the end of each trading day.

The green line that is train data shows the historical closing prices of the asset. It remains relatively stable until around late 2019 or early 2020. After this point, it experiences a sharp increase in value. The trend suggests a bullish movement.

Fig. 4.21 - Train Test Line Chart

The blue that is test data line starts being plotted around late 2019 or early 2020. It follows a similar trend as the Train Data but with more volatility. The Test Data also experiences a significant increase in closing prices. The overall movement is bullish, aligning with the Train Data.

4.4.2 Rescaling the Data

When we train a neural network, such as a Long Short-Term Memory (LSTM) recurrent neural network, it's essential to **rescale the data**. Rescaling ensures that the input data falls within a specific range, which is crucial for effective learning and convergence of the network.

```
In [90]: scaler = MinMaxScaler(feature_range=(0,1))
         bit_usd_lstm_train_scaled = scaler.fit_transform(bit_usd_lstm_train['Adj Close'].values.reshape(-1,1))

In [91]: bit_usd_lstm_train_scaled

Out[91]: array([[0.01445348],
                [0.01275083],
                [0.0112164 ],
                ...,
                [0.56504351],
                [0.55696479],
                [0.53232561]])

In [92]: bit_usd_lstm_train_scaled.shape

Out[92]: (2197, 1)
```

Fig. 4.22 - Rescaling of Train Data

```
In [111]: lstm_test_m

Out[111]: array([[0.00509892],
                 [0.        ],
                 [0.00896501],
                 [0.00779004],
                 [0.00880201],
                 [0.00923024],
                 [0.0080855 ],
                 [0.01044046],
                 [0.00939111],
                 [0.00651189],
                 [0.00575339],
                 [0.00528854],
                 [0.00738646],
                 [0.00954548],
                 [0.0062494 ],
                 [0.00737574],
                 [0.01167989],
                 [0.01427534],
                 [0.01832106],

In [112]: lstm_test_m.shape

Out[112]: (550, 1)
```

Fig. 4.23 - Rescaling of Test Data

There are some methods to rescale the data like normalisation, standardisation. In this thesis I used normalisation method to rescale the data in to zero and one. For the same I used MinMaxScaler formula from sklearn library. Which shows in fig 4.22 and 4.23

4.5 Model Building

4.5.1 ARIMA Model

For bitcoin price prediction I use ARIMA model. Let's discuss about hyperparameter tuning and model summary.

4.5.1.1 Hyperparameter Tuning

ARIMA model have three key hyperparameter, p q and d. p stand for the number of lag observation included in the model, also known as autoregressive model. q stands for the number of times the raw observation is differenced that is integration order. And third d stand for the size of the moving average window also known as moving average.

First, I define a range of values for p, d, and q. And the I iterate through all possible combinations of these values. Then I train ARIMA models with each combination and evaluate the models using a validation set or cross-validation. After that I Choose the combination that yields the best performance. And the best combination was (4, 1, 0).

4.5.1.2 Summary of ARIMA Model

```
==============================================================================
Dep. Variable:                      y   No. Observations:                 2746
Model:                 ARIMA(4, 1, 0)   Log Likelihood              -22204.217
Date:                Fri, 15 Mar 2024   AIC                          44418.434
Time:                        08:47:47   BIC                          44448.022
Sample:                             0   HQIC                         44429.125
                               - 2746
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.0238      0.010     -2.465      0.014      -0.043      -0.005
ar.L2         -0.0043      0.009     -0.475      0.635      -0.022       0.014
ar.L3          0.0094      0.010      0.963      0.336      -0.010       0.029
ar.L4          0.0372      0.008      4.539      0.000       0.021       0.053
sigma2      6.225e+05   5314.638    117.131      0.000    6.12e+05    6.33e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):             40277.25
Prob(Q):                              0.97   Prob(JB):                         0.00
Heteroskedasticity (H):            5529.15   Skew:                            -0.15
Prob(H) (two-sided):                  0.00   Kurtosis:                        21.76
===================================================================================
```

Fig. 4.24 - Summary of ARIMA Model

Summary of ARIMA model shows in fig. 4.24. in the figure 4.24 we can see total number of observations are 2746. P, q and d values are (4, 1, 0) respectively. Model also show P-value, standard error, z-score and other information.

4.5.2 LSTM Model

For bitcoin price prediction I use LSTM model. Let's discuss about hyperparameter tuning and model summary.

Table 4.1 - LSTM Model Summary

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 128)	66,560
dropout (Dropout)	(None, 60, 128)	0
lstm_1 (LSTM)	(None, 60, 64)	49,408
dropout_1 (Dropout)	(None, 60, 64)	0
lstm_2 (LSTM)	(None, 32)	12,416
dropout_2 (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Total params: 385,253 (1.47 MB)

Trainable params: 128,417 (501.63 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 256,836 (1003.27 KB)

The table 4.1 lists different layers used in the neural network. Each layer has a specific type (e.g., LSTM, Dropout, Dense). Layer indicates the type of layer (e.g., LSTM, Dropout, Dense). Output Shape Describes the shape of the output produced by each layer. Param # represents the number of parameters associated with each layer. LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) layer that handles sequential data. Dropout is a regularization technique that randomly drops out a fraction of input units during training to prevent overfitting. We used dropout for three times. Dense is a standard neural network layer where each neuron is connected to every neuron in the previous and subsequent layers. The output shape specifies the dimensions of the data produced by each layer. For example, (None, 60, 128) indicates a 3D tensor with 60-time steps and 128 features.

The "Param #" column represents the number of learnable parameters in each layer. These parameters are adjusted during training to optimize the model. the total number of parameters in the entire neural network is provided at the bottom. It includes trainable and non-trainable parameters.

4.6 Summary

In this analysis chapter, we delve into the heart of our research methodology. The experimental setup serves as the bridge between theory and practice, allowing us to rigorously evaluate the performance of our chosen models ARIMA and LSTM in the context of Bitcoin price prediction.

In this chapter we focused on analysis of bitcoin price. For the same I load the bitcoin price data after that did data cleaning and describing. Then we plot lots of charts to understand the data and to find hidden pattern in the bitcoin data. In this chapter we did univariate analysis, bivariate analysis of the data. After completing EDA I did some data pre-processing. First, we split the data into train and test. For the LSTM model, we did a Rescaling of the data. In this thesis, I did rescaling using MinMaxScaler. Then we build the ARIMA and LSTM models.

After building the models, we summarise and discuss the detailed information of the model. In the next chapter, I am going to evaluate the results of both models using evaluation metrics.

CHAPTER 5: RESULTS AND DISCUSSIONS

5.1 Introduction

In this chapter, we present the results and discuss the outcomes of our Bitcoin price prediction thesis utilizing ARIMA and LSTM models. Our study aims to forecast the price of Bitcoin, a volatile and widely traded cryptocurrency, employing two distinct yet powerful predictive techniques.

Analysis stars from finding hidden pattern in data. And then did some preprocessing of the data, then provide a comprehensive overview of the performance of both the ARIMA and LSTM models in predicting Bitcoin prices. This includes evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and others, to assess the accuracy and reliability of each model.

In this chapter, we discuss the results obtained from the ARIMA and LSTM model. We present the model's ability to capture the temporal dynamics of Bitcoin prices. Additionally, we discuss any challenges encountered during the modelling process and analysed the significance of different parameters in influencing the accuracy of predictions. We compare the performance of the ARIMA and LSTM models in predicting Bitcoin prices and discuss the same.

5.2 Finding of Hidden Patterns

The fig. 5.1 displays a line chart representing bitcoin price over time. The X-axis is labelled as “Date” and spans from the year **2018** to **2022**. The Y-axis represents a Adjusted closing prices ranging from **0** to approximately **70,000**.

Fig. 5.1 - Price Line Chart

From **2018** to **late 2020**, the closing price remains relatively stable. Around **late 2020**, there’s a sharp increase in the price, reaching its peak close to the beginning of **2021**. After reaching its peak, the bitcoin price fluctuates significantly throughout **2021**. By **early 2022**, the price stabilizes somewhat. The significant increase around **2020** suggests a notable event or change. The subsequent fluctuations indicate volatility or changing conditions. Line chart of bitcoin price shows that prices of bitcoin are volatile and unpredictable to certain extent.

Fig. 5.2 shows the scatter plot between volume and closing price of bitcoin. This figure was before removing outlier from volume variable. It shows that there is no strong relation between volume of bitcoin traded and price of the bitcoin.

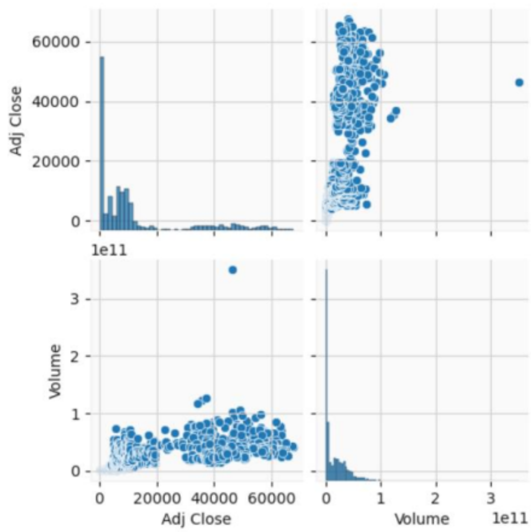


Fig. 5.2 - Before Outlier

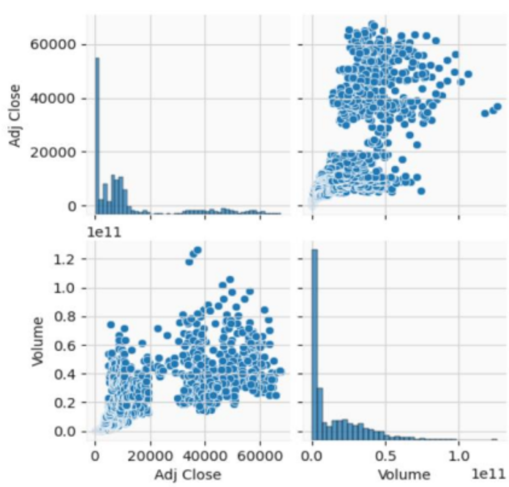


Fig. 5.3 - After Outlier

But when we remove single data point from the volume variable the same scatter plot shows strong relation between volume of the bitcoin data and closing price of the bitcoin data. Which shows in fig. 5.3.

Fig. 5.4 shows the line plot of closing price of bitcoin. It is the line chart of average adjusted closing price on the basis of year. We can say that there is drastic change in price from year 2020 to year 2021. And after that it starts decline.

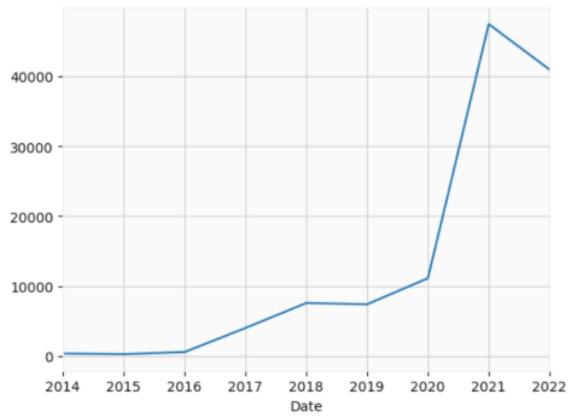


Fig. 5.4 - Average Price Per Year

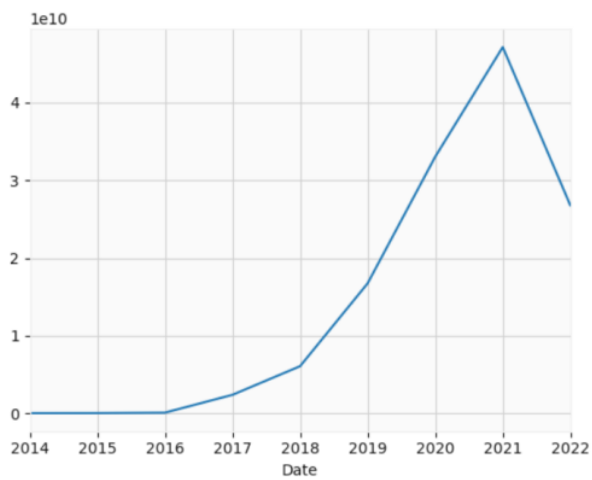


Fig. 5.5 - Average Volume Per Year

Fig. 5.5 is about average volume of bitcoin data on the basis of year. Figure shows that there is sharp increase in trading volume of bitcoin from year 2028 onwards. But after year 2021 it starts decline. That means there is strong relation between volume of the bitcoin and price of the bitcoin. Which shows in fig. 5.5.

5.3 Optimum Preprocessing Techniques

5.3.1 Train Test Split

There are total 2747 data points are available of bitcoin price. Out of 2747 data point I use first 80 percent data point as training data. That is 2197. And last 20 percent data point were use as testing data point which shows in fig. 5.6.

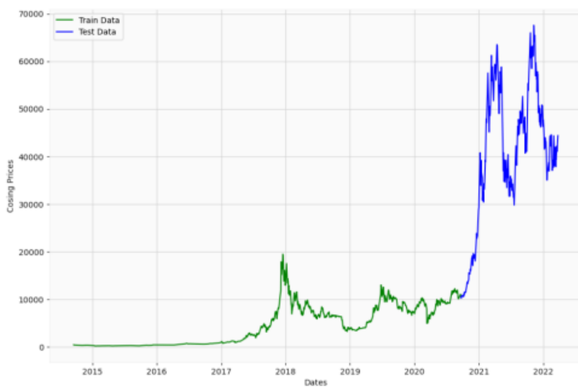


Fig. 5.6 - Train Test Split

Ther are different practices while splitting the data into train and test. Train test split randomly, for an example 60 & 40, 70 & 30, 90 & 10. For this thesis purpose I split data into 80 & 20. And provide better result after building models. Train test split shows in fig. 5.6.

5.3.2 Rescaling the Data

There are some methods to rescale the data like normalisation, standardisation. In this thesis I used normalisation method to rescale the data in to zero and one. For the same I used MinMaxScaler formula from sklearn library.

When we train a neural network, such as a Long Short-Term Memory (LSTM) recurrent neural network, it’s essential to **rescale the data**. Rescaling ensures that the input data falls within a specific range, which is crucial for effective learning and convergence of the network.

5.4 Model Building

5.4.1 ARIMA Model Building

Summary of ARIMA model shows in fig. 5.7. in the figure 5.7 we can see total number of observations are 2746. P, q and d values are (4, 1, 0) respectively. Model also show P-value, standard error, z-score and other information.

```
==============================================================================
Dep. Variable:                      y   No. Observations:                 2746
Model:                 ARIMA(4, 1, 0)   Log Likelihood              -22204.217
Date:                Fri, 15 Mar 2024   AIC                          44418.434
Time:                        08:47:47   BIC                          44448.022
Sample:                             0   HQIC                         44429.125
                               - 2746
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.0238      0.010     -2.465      0.014      -0.043      -0.005
ar.L2         -0.0043      0.009     -0.475      0.635      -0.022       0.014
ar.L3          0.0094      0.010      0.963      0.336      -0.010       0.029
ar.L4          0.0372      0.008      4.539      0.000       0.021       0.053
sigma2      6.225e+05   5314.638    117.131      0.000    6.12e+05    6.33e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):             40277.25
Prob(Q):                              0.97   Prob(JB):                         0.00
Heteroskedasticity (H):            5529.15   Skew:                            -0.15
Prob(H) (two-sided):                  0.00   Kurtosis:                        21.76
===================================================================================
```

Fig. 5.7 - ARIMA Summary

5.4.2 ARIMA Model Evaluation

The image 5.8 displays a line chart representing data over time. The X-axis is labelled as “Dates” and spans from September 2020 to March 2022. The Y-axis is labelled as “Closing Prices” and ranges from approximately 10,000 to 70,000.

There are two lines on the graph: A red line (labelled as btc_price_actual) represents the actual closing prices of Bitcoin during this period. A blue dashed line (labelled as btc_price_prediction) shows the predicted closing prices based on an ARIMA model. Both lines exhibit similar trends, indicating that the ARIMA model predictions closely align with the actual prices.

The actual and predicted prices follow a similar trajectory. Around late 2020, there’s a notable increase in Bitcoin’s closing price. The peak occurs close to mid-2021. After reaching its peak, the price experiences fluctuations but remains relatively high compared to previous years. Exactly same trend is shown by predicted bitcoin price line.

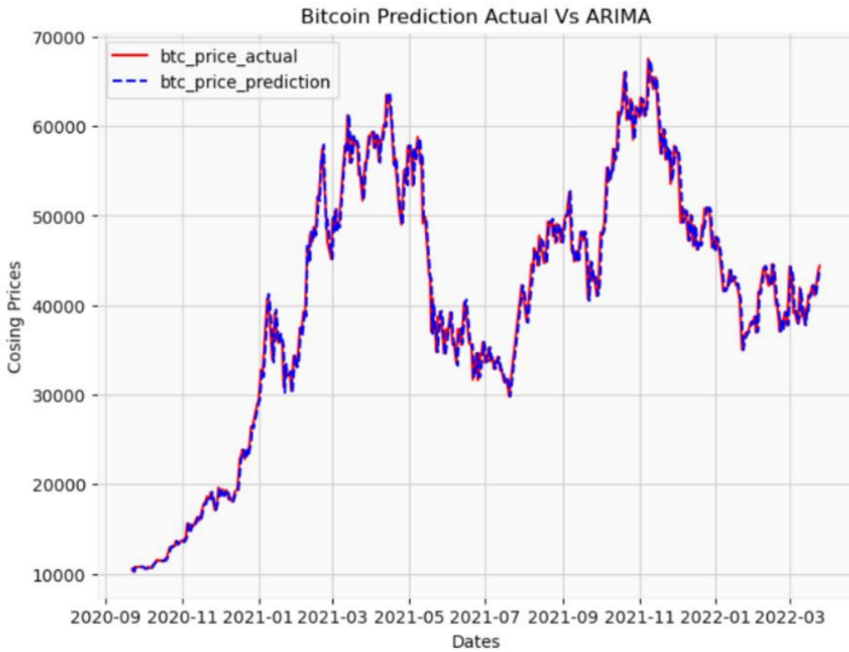


Fig. 5.8 - Prediction Actual vs ARIMA

5.4.3 ARIMA Model Evaluation Metrics

There are lots of Evaluation metrics are available to evaluate model. This thesis covered some of metrics like, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE) and R2 Score. Which shown in table 5.1.

Table 5.1 – ARIMA Evaluation Metrics

Evaluation Metrics	**Result**
Mean Absolute Error	1178.50
Mean Absolute Percentage Error	0.03
Mean Squared Error	2830220
Root Mean Squared Error	1682.32
R2 Score	0.98

5.4.4 LSTM Model Building

The table 5.2 lists different layers used in the neural network. Each layer has a specific type (e.g., LSTM, Dropout, Dense). Layer indicates the type of layer (e.g., LSTM, Dropout, Dense). Output Shape Describes the shape of the output produced by each layer. Param # represents the number of parameters associated with each layer. LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) layer that handles sequential data. Dropout is a regularization technique that randomly drops out a fraction of input units during training to prevent overfitting. We used dropout for three times. Dense is a standard neural network layer where each neuron is connected to every neuron in the previous and subsequent layers. The output shape specifies the dimensions of the data produced by each layer. For example, (None, 60, 128) indicates a 3D tensor with 60-time steps and 128 features.

Table 5.2 – LSTM Model Summary

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 128)	66,560
dropout (Dropout)	(None, 60, 128)	0
lstm_1 (LSTM)	(None, 60, 64)	49,408
dropout_1 (Dropout)	(None, 60, 64)	0
lstm_2 (LSTM)	(None, 32)	12,416
dropout_2 (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Total params: 385,253 (1.47 MB)

Trainable params: 128,417 (501.63 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 256,836 (1003.27 KB)

“Param #” column represents the number of learnable parameters in each layer. These parameters are adjusted during training to optimize the model. the total number of parameters in the entire neural network is provided at the bottom. It includes trainable and non-trainable parameters.

5.4.5 LSTM Model Evaluation

The image 5.9 displays a line chart representing data over time. The X-axis is labelled as “Bitcoin Day interval” and spans from September 2020 to March 2022 or 0 to 490 data points. The Y-axis is labelled as “Bitcoin Prices” and ranges from approx. 10,000 to 70,000.

There are two lines on the graph: A blue line (labelled as Actual Prices) represents the actual closing prices of Bitcoin during this period. A orange line (labelled as Predicted Prices) shows the predicted closing prices based on an LSTM model. Both lines exhibit similar trends, indicating that the LSTM model predictions closely align with the actual prices.

The actual and predicted prices follow a similar trajectory to the certain extent. Exactly same trend is shown by predicted bitcoin price line to the certain extent. But when we compare prediction made by LSTM model with prediction made by ARIMA model. Then result shows that LSTM model is lagging little bit.

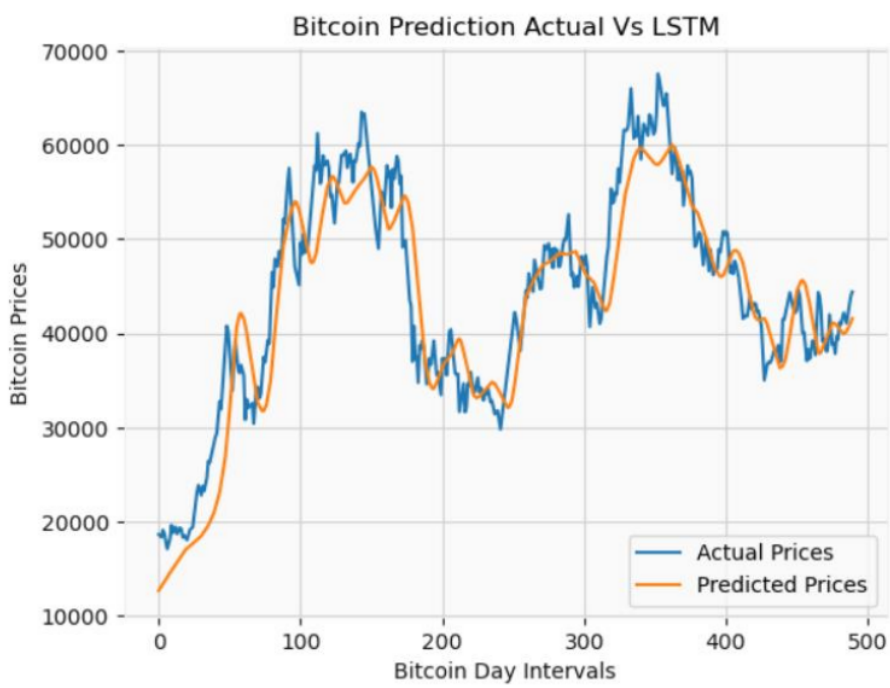


Fig. 5.9 - Prediction Actual vs LSTM

5.4.6 LSTM Model Evaluation Metrics

There are lots of Evaluation metrics are available to evaluate LSTM model. This thesis covered some of metrics like, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE) and R2 Score. Which shown in table 5.3.

Table 5.3 – LSTM Evaluation Metrics

Evaluation Metrics	Result
Mean Absolute Error	3530.32
Mean Absolute Percentage Error	0.08
Mean Squared Error	19538224
Root Mean Squared Error	4420.20
R2 Score	0.85

5.5 Comparative Analysis of ARIM & LSTM Model

Figure 5.10 shows three lines. Blue, orange and green. Blue represents actual price of bitcoin data. Orange represents bitcoin price predicted by LSTM model. and green line represents bitcoin predicted price by ARIMA model. X- axis represents day intervals and Y – axis shows adjusted closing price of bitcoin.

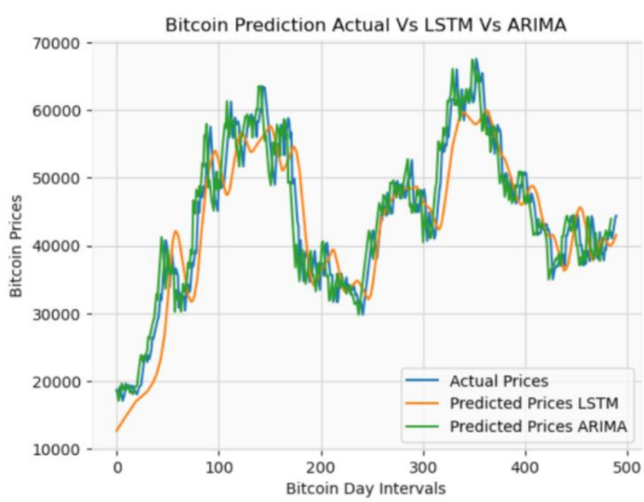


Fig. 5.10 – Prediction Actual vs LSTM vs ARIMA

Green and blue line follow similar trajectory that means prediction made by ARIMA model is good. ARIMA model predict bitcoin prices exactly same as actual bitcoin prices. But when we compare orange line with actual prices that is blue line, we can find lots of lagging in the prediction. The LSTM model prediction line is jumping ups and down from actual bitcoin price line. That means ARIMA model perform better than LSTM model for given bitcoin data.

Table 5.4 explain that, ARIMA achieves a significantly lower MAE (1178.50) compared to LSTM (3530.32). A lower MAE indicates that ARIMA's predictions are closer to the actual values. ARIMA has a much lower MAPE (0.03) than LSTM (0.08). A lower MAPE suggests better accuracy in predicting percentage errors. ARIMA's MSE (2830220) is substantially smaller than LSTM's (19538224). A smaller MSE implies better precision in predicting squared errors. ARIMA's RMSE (1682.32) is significantly lower than LSTM's (4420.20). A lower RMSE indicates better overall prediction performance. ARIMA achieves a high R2 score of 0.98, indicating that it explains most of the variability in the target data around its mean. LSTM's R2 score is 0.85, which is still good but not as high as ARIMA's. In summary, based on these evaluation metrics, ARIMA outperforms LSTM in terms of accuracy, precision, and overall predictive capability.

Table 5.4 – Comparative Analysis of ARIM & LSTM

Evaluation Metrics	ARIMA Result	LSTM Result
Mean Absolute Error	1178.50	3530.32
Mean Absolute Percentage Error	0.03	0.08
Mean Squared Error	2830220	19538224
Root Mean Squared Error	1682.32	4420.20
R2 Score	0.98	0.85

5.6 Summary

This chapter discussed all results of all objectives. This chapter discussed about how I found hidden patterns in the data, it discusses about optimum preprocessing techniques, model building and Evaluation of LSTM and ARIMA model base on certain evaluation techniques. And finally, it did comparative analysis between ARIMA and LSTM model. Based on these evaluation metrics, ARIMA outperforms LSTM in terms of accuracy, precision, and overall predictive capability.

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

In our study, we explored two popular time-series forecasting models ARIMA and LSTM. Based on some evaluation metrics this thesis compared performance of the both models, ARIMA and LSTM. We make some conclusions based on the result opt from the Bitcoin data analysis. In this final section, we summarize the key findings and implications of our Bitcoin price prediction thesis, which utilized both ARIMA and LSTM models. Also, we summarize conclusion, future work and also recommendations. Through this research, we aimed to provide valuable insights comparative analysis into the effectiveness of these two distinct methodologies in forecasting the volatile and dynamic nature of Bitcoin prices.

6.2 Discussion and Conclusion

In conclusion, the comparative analysis of ARIMA and LSTM models for predicting Bitcoin prices reveals the following:

1. ARIMA Model-
   - Achieves better accuracy and precision across all evaluation metrics.
   - Lower Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).
   - Higher R2 Score (0.98), indicating effective explanation of variability around the mean.
2. LSTM Model-
   - Still performs well with an R2 Score of 0.85.
   - May have other advantages (e.g., handling sequential data) not captured by these metrics.

Therefore, for accurate and precise Bitcoin price predictions, ARIMA is the preferred choice.

6.3 Contribution to Knowledge

The study provides valuable insights into the performance of two prominent time-series forecasting models: ARIMA and LSTM. Researchers, practitioners, and investors can use this information to make informed decisions when choosing a model for predicting Bitcoin prices. The results demonstrate empirically that ARIMA consistently outperforms LSTM across various evaluation metrics.

This empirical evidence adds to the existing body of knowledge regarding the effectiveness of ARIMA in financial time series prediction. By quantifying metrics such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE), the study highlights ARIMA's superior accuracy and precision. Practitioners can rely on ARIMA for more accurate price forecasts, leading to better investment decisions. The study emphasizes the importance of R2 in assessing how well the model explains the variability in the target data. Researchers can adapt the comparative analysis framework to evaluate models for different cryptocurrencies or financial instruments.

In summary, the comparative analysis contributes to our understanding of model performance in cryptocurrency price prediction. It informs practitioners about the strengths and limitations of ARIMA and LSTM, ultimately advancing knowledge in the field of financial forecasting.

6.4 Future Recommendations

The future direction of Bitcoin price prediction involves a convergence of advanced methodologies, integration of external data sources, and the continuous pursuit of adaptable, interpretable models capable of addressing the unique challenges posed by cryptocurrency markets. Interdisciplinary collaboration and exploration of emerging technologies also recommended to shape the evolution of predictive models in the cryptocurrency domain. It is recommended to implement real-time prediction frameworks to enable timely decision-making for traders and investors.

REFERENCES

-, Dr. S. M. (2023). An International Study of Application of Long Short-Term Memory (LSTM) Neural Networks for the prediction of stock and forex markets. *International Journal For Multidisciplinary Research*, *5*(3), 1–9. https://doi.org/10.36948/ijfmr.2023.v05i03.3345

Abbasimehr, H., & Paki, R. (2022). Improving time series forecasting using LSTM and attention models. *Journal of Ambient Intelligence and Humanized Computing*, *13*(1), 673–691. https://doi.org/10.1007/s12652-020-02761-x

Alazzam, F. A. F., Salih, A. J., Amoush, M. A. M., & Khasawneh, F. S. A. (2023). the Nature of Electronic Contracts Using Blockchain Technology - Currency Bitcoin As an Example. *Revista de Gestao Social e Ambiental*, *17*(5), 1–18. https://doi.org/10.24857/rgsa.v17n5-014

Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., & Muneer, A. (2023). LSTM Inefficiency in Long-Term Dependencies Regression Problems. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *30*(3), 16–31. https://doi.org/10.37934/araset.30.3.1631

Ammer, M. A., & Aldhyani, T. H. H. (2022). Deep Learning Algorithm to Predict Cryptocurrency Fluctuation Prices: Increasing Investment Awareness. *Electronics (Switzerland)*, *11*(15), 1–22. https://doi.org/10.3390/electronics11152349

Andhika Viadinugroho, R. A., & Rosadi, D. (2023). A weighted metric scalarization approach for multiobjective BOHB hyperparameter optimization in LSTM model for sentiment analysis. *Information Sciences*, *644*(May), 119282. https://doi.org/10.1016/j.ins.2023.119282

Ashawa, M., Douglas, O., Osamor, J., & Jackie, R. (2022). Improving cloud efficiency through optimized resource allocation technique for load balancing using LSTM machine learning algorithm. *Journal of Cloud Computing*, *11*(1). https://doi.org/10.1186/s13677-022-00362-x

Balestriero, R., Bottou, L., & LeCun, Y. (2022). The Effects of Regularization and Data Augmentation are Class Dependent. *Advances in Neural Information Processing Systems*, *35*, 1–21.

Carbó, J. M., & Gorjon, S. (2022). Application of Machine Learning Models and Interpretability Techniques to Identify the Determinants of the Price of Bitcoin. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4087481

Carpenter, M., Luo, C., & Wang, X. S. (2021). The effects of regularisation on RNN models for Covid-19 time series forecasting. *Proceedings - 2021 20th International Conference on Ubiquitous Computing and Communications, 2021 20th International Conference on Computer and Information Technology, 2021 4th International Conference on Data Science and Computational Intelligence and 20*, *May 2021*, 281–287. https://doi.org/10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00054

Centobelli, P., Cerchione, R., Vecchio, P. Del, Oropallo, E., & Secundo, G. (2022). Blockchain technology for bridging trust, traceability and transparency in circular supply chain. *Information and Management*, *59*(7). https://doi.org/10.1016/j.im.2021.103508

Choi, J., Won, J., Jang, S., & Kim, S. (2022). Learning Enhancement Method of Long Short-Term Memory Network and Its Applicability in Hydrological Time Series Prediction. *Water (Switzerland)*, *14*(18). https://doi.org/10.3390/w14182910

Ciarko, M., Poszwa, G., Paluch-Dybek, A., & Timur, M. C. (2023). Cryptocurrencies as the future of money: Theoretical aspects, blockchain technology and origins of cryptocurrencies. *Virtual Economics*, *6*(3), 70–93. https://doi.org/10.34021/ve.2023.06.03(5)

Dai, X., Yin, H., & Jha, N. K. (2020). Grow and Prune Compact, Fast, and Accurate LSTMs. *IEEE Transactions on Computers*, *69*(3), 441–452. https://doi.org/10.1109/TC.2019.2954495

Danylkiv, K., Dropa, Y., Petyk, M., Hembarska, N., & Siryk, Z. (2022). Development of Innovative Tools in Global Financial Markets. *Journal of Vasyl Stefanyk Precarpathian National University*, *9*(3), 65–77. https://doi.org/10.15330/jpnu.9.3.65-77

De Pace, P., & Rao, J. (2023). Comovement and instability in cryptocurrency markets. *International Review of Economics and Finance*, *83*, 173–200. https://doi.org/10.1016/j.iref.2022.08.010

Dimri, T., Ahmad, S., & Sharif, M. (2020). Time series analysis of climate variables using seasonal ARIMA approach. *Journal of Earth System Science*, *129*(1). https://doi.org/10.1007/s12040-020-01408-x

Dinshaw, C., Jain, R., & Hussain, S. A. I. (2022). Statistical Scrutiny of the Prediction Capability of Different Time Series Machine Learning Models in Forecasting Bitcoin Prices. *Proceedings of 4th International Conference on Cybernetics, Cognition and Machine Learning Applications, ICCCMLA 2022*, *Ml*, 329–336. https://doi.org/10.1109/ICCCMLA56841.2022.9989057

Elsaraiti, M., & Merabet, A. (2021). A comparative analysis of the arima and lstm predictive models and their effectiveness for predicting wind speed. *Energies*, *14*(20). https://doi.org/10.3390/en14206782

Giovani, M., Anggriani, I., & Simatupang, S. A. W. D. (2022). Comparison in Predicting the Short-Term Using the Sarima, Dsarima and Tsarima Methods. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, *16*(4), 1487–1496. https://doi.org/10.30598/barekengvol16iss4pp1487-1496

Gunarto, D. M., Sa'adah, S., & Utama, D. Q. (2023). Predicting Cryptocurrency Price Using RNN and LSTM Method. *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, *12*(1), 1–8. https://doi.org/10.32736/sisfokom.v12i1.1554

Guo, Q., Lei, S., Ye, Q., & Fang, Z. (2021). MRC-LSTM: A Hybrid Approach of Multi-scale Residual CNN and LSTM to Predict Bitcoin Price. *Proceedings of the International Joint Conference on Neural Networks*, *2021-July*, 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534453

Hamayel, M. J., & Owda, A. Y. (2021). A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. *AI (Switzerland)*, *2*(4), 477–496. https://doi.org/10.3390/ai2040030

Hua, Y. (2020). Bitcoin price prediction using ARIMA and LSTM. *E3S Web of Conferences*, *218*. https://doi.org/10.1051/e3sconf/202021801050

Huang, H., & Ye, T. (2022). Asset Trading Strategies Based on LSTM - Bitcoin and Gold as an Example. *CAIBDA 2022 - 2nd International Conference on Artificial Intelligence, Big Data and Algorithms*, 650–655.

Implications, F. (2023). Deciphering the Blockchain: A Comprehensive Analysis of Bitcoin's Evolution, Adoption, and Future Implications. *Journal of Electrical Electronics Engineering*, *2*(2). https://doi.org/10.33140/jeee.02.02.09

Jia, C., & Zhang, Y. (2020). Multi-cell compositional LSTM for NER domain adaptation. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 5906–5917. https://doi.org/10.18653/v1/2020.acl-main.524

Kazeminia, S., Sajedi, H., & Arjmand, M. (2023). Real-Time Bitcoin Price Prediction Using Hybrid 2D-CNN LSTM Model. *2023 9th International Conference on Web Research, ICWR 2023*, 173–178. https://doi.org/10.1109/ICWR57742.2023.10139275

Kilimci, Z. H. (2020). International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Sentiment Analysis Based Direction Prediction in Bitcoin using Deep Learning Algorithms and Word Embedding Models. *Original Research Paper International Journal of Intelligent Systems and Applications in Engineering IJISAE*, *2020*(2), 60–65. www.ijisae.org

Kim, J. H., & Sung, H. (2022). Understanding Bitcoin Price Prediction Trends under Various Hyperparameter Configurations. *Computers*, *11*(11). https://doi.org/10.3390/computers11110167

Latif, N., Selvam, J. D., Kapse, M., Sharma, V., & Mahajan, V. (2023). Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices. *Australasian Accounting, Business and Finance Journal*, *17*(1), 256–276. https://doi.org/10.14453/aabfj.v17i1.15

Li, S., Luo, R., Zhang, W., Jiang, J., Zhu, L., Wang, Z., & Jiang, X. (2022). Preprocessing Method for Industrial Data Based on LSTM Model Prediction. *Proceedings - 2022 10th International Conference on Information Systems and Computing Technology, ISCTech 2022*, 681–684. https://doi.org/10.1109/ISCTech58360.2022.00112

Li, T. (2022). Prediction of Bitcoin Price Based on LSTM. *Proceedings - 2022 International Conference on Machine Learning and Intelligent Systems Engineering, MLISE 2022*, 19–23. https://doi.org/10.1109/MLISE57402.2022.00012

Lin, Q. (2023). Price Prediction of Bitcoin Using LSTM Neural Network. *BCP Business & Management*, *38*(November 2022), 2738–2744. https://doi.org/10.54691/bcpbm.v38i.4183

Mandke, A., Litake, O., & Kadam, D. (2021). Analyzing Architectures for Neural Machine Translation using Low Computational Resources. *International Journal on Natural Language Computing*, *10*(5), 9–16. https://doi.org/10.5121/ijnlc.2021.10502

McElroy, T. (2021). A Diagnostic for Seasonality Based Upon Polynomial Roots of ARMA Models. *Journal of Official Statistics*, *37*(2), 367–394. https://doi.org/10.2478/jos-2021-0016

McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. *Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018*, 339–343. https://doi.org/10.1109/PDP2018.2018.00060

Mishal, M. H., Jannat Rakhi, N., Rashid, F., Hamid, K., Morol, M. K., Al Jubair, A., & Nandi, D. (2022). Prediction of Cryptocurrency Price using Machine Learning Techniques and Public Sentiment Analysis. *Proceedings of 2022 25th International Conference on Computer and Information Technology, ICCIT 2022*, 657–662. https://doi.org/10.1109/ICCIT57492.2022.10055524

Modi, P. D., Arshi, K., Kunz, P. J., & Zoubir, A. M. (2023). A Data-driven Deep Learning Approach for Bitcoin Price Forecasting. *International Conference on Digital Signal Processing, DSP*, *2023-June*, 1–4. https://doi.org/10.1109/DSP58604.2023.10167930

Mohamed, J. (2020). Time Series Modeling and Forecasting of Somaliland Consumer Price Index: A Comparison of ARIMA and Regression with ARIMA Errors. *American Journal of Theoretical and Applied Statistics*, *9*(4), 143. https://doi.org/10.11648/j.ajtas.20200904.18

Mr. R. Arunachalam, M. Santhoshini, R. Tamil Prabha, & R. Tamil Prabha. (2023). Predicting the Price of Bitcoin using LSTM Recurrent Neural Network. *International Journal of Advanced Research in Science, Communication and Technology*, 615–618. https://doi.org/10.48175/ijarsct-9626

Mtiraoui, A., Boubaker, H., & BelKacem, L. (2023). A hybrid approach for forecasting bitcoin series. *Research in International Business and Finance*, *66*(May), 102011. https://doi.org/10.1016/j.ribaf.2023.102011

Ospina, R., Gondim, J. A. M., Leiva, V., & Castro, C. (2023). An Overview of Forecast Analysis with ARIMA Models during the COVID-19 Pandemic: Methodology and Case Study in Brazil. *Mathematics*, *11*(14), 1–18. https://doi.org/10.3390/math11143069

Pan, L. (2023). Cryptocurrency Price Prediction Based on ARIMA, Random Forest and LSTM Algorithm. *BCP Business & Management*, *38*, 3396–3404. https://doi.org/10.54691/bcpbm.v38i.4313

Pasak, S., & Jayadi, R. (2023). Investment Decision on Cryptocurrency: Comparing Prediction Performance Using ARIMA and LSTM. *Journal of Information Systems and Informatics*, *5*(2), 407–427. https://doi.org/10.51519/journalisi.v5i2.473

Peng, A., Zhang, X., Xu, W., & Tian, Y. (2022). Effects of Training Data on the Learning Performance of LSTM Network for Runoff Simulation. *Water Resources Management*, *36*(7), 2381–2394. https://doi.org/10.1007/s11269-022-03148-7

Perone, G. (2020). An ARIMA Model to Forecast the Spread of COVID-2019 Epidemic in Italy. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3564865

Raju, S. M., & Tarif, A. M. (2020). *Real-Time Prediction of BITCOIN Price using Machine Learning Techniques and Public Sentiment Analysis*. http://arxiv.org/abs/2006.14473

Ramani, K., Jahnavi, M., Reddy, P. J., Venkatachakravarthi, P., Meghanath, P., & Imran, S. K. (2023a). Prediction of Bitcoin Price through LSTM, ARIMA, XGBoost, Prophet and Sentiment Analysis on Dynamic Streaming Data. *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023*, *1*, 1514–1518. https://doi.org/10.1109/ICACCS57279.2023.10113014

Ramani, K., Jahnavi, M., Reddy, P. J., Venkatachakravarthi, P., Meghanath, P., & Imran, S. K. (2023b). Prediction of Bitcoin Price through LSTM, ARIMA, XGBoost, Prophet and Sentiment Analysis on Dynamic Streaming Data. *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023*, *1*, 1514–1518. https://doi.org/10.1109/ICACCS57279.2023.10113014

Ramani, K., Jahnavi, M., Reddy, P. J., Venkatachakravarthi, P., Meghanath, P., & Imran, S. K. (2023c). Prediction of Bitcoin Price through LSTM, ARIMA, XGBoost, Prophet and Sentiment Analysis on Dynamic Streaming Data. *2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023*, *1*, 1514–1518. https://doi.org/10.1109/ICACCS57279.2023.10113014

Saju, B., Gopal, M. K., Nithya, B., Asha, V., & Kumar, V. (2022). Analysis on Role of Quantum Computing in Machine Learning. *2022 4th International Conference on Cognitive Computing and Information Processing, CCIP 2022*, 1–8. https://doi.org/10.1109/CCIP57447.2022.10058679

Saputra, A. W., Wibawa, A. P., Pujianto, U., Putra Utama, A. B., & Nafalski, A. (2022). LSTM-based Multivariate Time-Series Analysis: A Case of Journal Visitors Forecasting. *ILKOM Jurnal Ilmiah*, *14*(1), 57–62. https://doi.org/10.33096/ilkom.v14i1.1106.57-62

Sener, E., & Demir, I. (2022). Wavelet-Denoised ARIMA and LSTM Method to Predict Cryptocurrency Price. *ISMSIT 2022 - 6th International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings*, 577–582. https://doi.org/10.1109/ISMSIT56059.2022.9932727

Shen, F. (2022). Incorporating Sentiment and Temporal Information for Bitcoin Price Prediction. *BCP Business & Management*, *34*, 784–793. https://doi.org/10.54691/bcpbm.v34i.3097

Shen, J. T., & Lee, D. (2023). *Imputing Knowledge Tracing Data with Subject-Based Training via LSTM Variational Autoencoders Frameworks*. *i*. http://arxiv.org/abs/2302.12910

Sher, M., Minallah, N., Ahmad, T., & Khan, W. (2023). Hyperparameters analysis of long short-term memory architecture for crop classification. *International Journal of Electrical and Computer Engineering*, *13*(4), 4661–4670. https://doi.org/10.11591/ijece.v13i4.pp4661-4670

Shirdel, M., Asadi, R., Do, D., & Hintlian, M. (2021). *Deep Learning with Kernel Flow Regularization for Time Series Forecasting*. http://arxiv.org/abs/2109.11649

Si, Y. (2022). Using ARIMA model to analyse and predict bitcoin price. *BCP Business & Management*, *34*, 1210–1216. https://doi.org/10.54691/bcpbm.v34i.3161

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

Song, X., Liu, Y., Xue, L., Wang, J., Zhang, J., Wang, J., Jiang, L., & Cheng, Z. (2020). Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model. *Journal of Petroleum Science and Engineering*, *186*(November 2019), 106682. https://doi.org/10.1016/j.petrol.2019.106682

Sorkun, M. C., Durmaz Incel, Ö., & Paoli, C. (2020). Time series forecasting on multivariate solar radiation data using deep learning (LSTM). *Turkish Journal of Electrical Engineering and Computer Sciences*, *28*(1), 211–223. https://doi.org/10.3906/elk-1907-218

Srivastava, V., Kumar Dwivedi, V., & Kumar Singh, A. (2023). Prediction of Bitcoin Price using Optimized Genetic ARIMA Model and Analysis in Post and Pre Covid Eras*. *Proceedings - 2023 3rd International Conference on Smart Data Intelligence, ICSMDI 2023*, 143–148. https://doi.org/10.1109/ICSMDI57622.2023.00033

Tunkiel, A. T., Sui, D., & Wiktorski, T. (2022). Impact of data pre-processing techniques on recurrent neural network performance in context of real-time drilling logs in an automated prediction framework. *Journal of Petroleum Science and Engineering*, *208*(PE), 109760. https://doi.org/10.1016/j.petrol.2021.109760

Turek, J., Beckage, N., Vo, V., Honey, C., & Willke, T. (2022). *Long Short-Term Memory with Slower Information Decay*. 2–5. https://doi.org/10.52591/lxai2021072413

Vital, G., Quintero, J., Miranda, P., & Soto, C. (2023). *Modelos ARIMA para el an´ alisis sistematizado de criptomonedas ARIMA models for the systematic analysis of cryptocurrencies*. 1–7.

Wang, C. C., Chien, C. H., & Trappey, A. J. C. (2021). On the application of ARIMA and LSTM to predict order demand based on short lead time and on-time delivery requirements. *Processes*, *9*(7). https://doi.org/10.3390/pr9071157

Wang, S., Ma, C., Xu, Y., Wang, J., & Wu, W. (2022). A Hyperparameter Optimization Algorithm for the LSTM Temperature Prediction Model in Data Center. *Scientific Programming*, *2022*. https://doi.org/10.1155/2022/6519909

Zhang, X., Qi, W., & Zhan, Z. (2021). A Study on Machine-Learning-Based Prediction for Bitcoin's Price via Using LSTM and SVR. *Journal of Physics: Conference Series*, *1732*(1). https://doi.org/10.1088/1742-6596/1732/1/012027

Zhang, Z. (2023). Price Prediction of Bitcoin, Ethereum and XRP Based on the ARMA Model. *BCP Business & Management*, *44*, 48–54. https://doi.org/10.54691/bcpbm.v44i.4792

Zhong, R. (2023). An application of deep learning on bitcoin price forecasting. *Applied and Computational Engineering*, *4*(1), 422–429. https://doi.org/10.54254/2755-2721/4/20230510

APPENDIX A: RESEARCH PROPOSAL

PREDICTING BITCOIN PRICING TRENDS USING TIME SERIES FORECASTING TECHNICS.

HEMANT BAJARANG KOKANE

RESEARCH PROPOSAL

NOVEMBER 2023

Abstract

This research will focus on Bitcoin price prediction using long-short-term memory (LSTM) and autoregressive integrated moving average (ARIMA) algorithms. The main objective of this study is to compare Bitcoin prices predicted by both models so I can provide the best Bitcoin price prediction model. I am going to use the BTC-USD dataset which is publicly available on the Kaggle website. The research will begin with an in-depth analysis of historical Bitcoin price trends, identifying key factors influencing market dynamics, and identifying hidden patterns and trends in data. I am going to use some pre-processing techniques and data transformation so I can meet model requirements. I will split the data into 80/20 percent. 80% of the data will be train data and testing data will be 20%. I will use Min-max scaling to normalize the data. Subsequently, I will build machine learning or deep learning algorithms like ARIMA and LSTM and the same will be evaluated based on evaluation metrics. For model evaluation, I am going to use evaluation metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). After completing this study, I can suggest the best algorithm for Bitcoin price forecasting.

Table of Contents

1. Background

The cryptocurrency landscape, characterized by its dynamic nature and decentralized structure, has witnessed unprecedented growth, with Bitcoin emerging as a frontrunner in this financial revolution. As the world embraces the possibilities inherent in digital currencies, the volatility and complexity of the cryptocurrency market pose intriguing challenges for investors and researchers alike. Among these challenges, predicting the price movements of Bitcoin stands as a focal point, influencing strategic decision-making, risk management, and investment portfolios.

In this study, I am going to predict Bitcoin prices by using time series forecasting techniques and build a machine-learning algorithm model with high prediction accuracy that helps investors invest money in cryptocurrency or stocks. While doing this study, the main objectives will be to analyze the hidden patterns in the Bitcoin dataset via EDA, determine the optimum pre-processing technique, and evaluate and compare the performance of models based on evaluation metrics.

I am going to compare results obtained from various deep learning algorithms, so I can provide the best algorithm for Bitcoin price prediction. There are lots of models available in the market, like autoregressive integrated moving average (ARIMA), SARIMAX, FbProphet, Long Shortterm Memory (LSTM), and Gate Recurrent Unit (GRU). Out of this, I will use ARIMA and LSTM models, to forecast prices. In this study, I will compare the results obtained from both models and evaluate them by using some evaluation metrics. There are some evaluation metrics available, like MAE, MSE, RMSE, MAPE, and R2 Score (Coefficient of Determination). I will use Mean Absolute Error, Root Mean Squared Error, and Mean Absolute Percentage Error. To evaluate model accuracy.

This study is useful to investors or stakeholders in the cryptocurrency space. It can provide information to investors so they can form their decision about buying, selling, or holding Bitcoin. Accordingly, they can change their investment strategies. This model can anticipate potential market fluctuation so investors can minimize their risk. It can provide optimal entry and exit points, which can enhance investor's strategies. Financial institutes can use prediction model to regulate their policies, and they can form cryptocurrency-related policies

The price of Bitcoin or cryptocurrency depends on demand and the volume available for trading. If investors can forecast the price of a cryptocurrency, they can invest more money. Maximum people are interested in investing in Bitcoin or Cryptocurrency, but they do have not time to search for it. Hence Cryptocurrency price predictions are important (Dinshaw et al., 2022).

Some of the authors build a model that provides the best daily strategy. They broke the solution into three parts. First, they forecast prices by using different models and then choose the best which is the LSTM model. They use the GARCH and VAR models as risk indicators. With the help of the PSO model, they optimize sharp index, and objective function for best strategy. Additionally, different cost functions are provided in this article to generalize our model so investors who have divergent risk attitudes can be saved. Their strategy achieves a gain of $27263 with a VAR value of 502.51 (Huang & Ye, 2022).

According to research both LSTM and ARIMA will perform well while predicting Bitcoin price. The LSTM model predicts Bitcoin prices more accurately than the ARIMA algorithm. While using 4 core CPU it takes more time. And minimum time while using 2 core GPU. ARIMA model is useful for short periods. Whereas the LSTM algorithm predicts a long period. For a short period, ARIMA predicts correctly but when the period increases the accuracy of the ARIMA model decreases (Hua, 2020).

Bitcoin uses a virtual platform for exchange. Bitcoin was launched in Jan-2009. After the 2008 Great Depression or mortgage crisis Bitcoin provided an alternative protection for investors. Bitcoin have more advantage and reliable currency compared to traditional currencies. In the past time, a medium of exchange was money, allowing traders to make purchases. This trust applies to Bitcoin in the cryptocurrency market. There is no single location of Bitcoin. And users control Bitcoin. Every Bitcoin transaction is recorded, and there is a limit supply limit of Bitcoin which is 21 M globally (Sener & Demir, 2022)

2. Problem statement

I will use time series forecasting techniques such as Autoregressive integrated moving average (ARIMA) and Long Short-Term Memory (LSMT) to predict Bitcoin prices in this dissertation or thesis. I read a few papers related to this thesis, and they covered the majority of topics, which I am going to touch on in this thesis.

The author compares the accuracy of prices in USD prediction. He uses two time series forecasting techniques, Long Short-term Memory (LSTM) and ARIMA. He gathered data by using PycURL from Bitfine. Keras and TensorFlow interfaces were used to build the LSTM model. He did a classical comparison by using the ARIMA model. He trained the model with 100 epochs and 10 pieces of data in each round. The model was trained with 5 and 10 previous data points. He compares the result of 5 data points with 10 data points. After training the model the average error rate was 0.4765938 and the standard deviation was 2.092208. Both algorithms perform well while forecasting Bitcoin prices. LSTM model took approx. 42 M via a four-core CPU and 1 M while using a two-core GPU. ARIMA model was good for the short term, but the LSTM model was more efficient while predicting long-span Bitcoin prices. (Hua, 2020).

The authors did a comparative study between LSTM and ARIMA models for short-term prediction of Bitcoin Prices. They forecast Bitcoin prices for the next day by using the static forecast method. To check the accuracy of prediction they split the data into train and test datasets. In the first training sample ARIMA model forecast better than the LSTM algorithm. But in the second training sample LSTM algorithm predicts better than the ARIMA algorithm. According to the authors, the accuracy of the LSTM model was 99.73 percent. Which was much better than the ARIMA model. ARIMA model predicts accurately because of an upward trend in data. Otherwise, ARIMA would not be able to predict correctly. (Latif et al., 2023).

The authors did a Bitcoin Price prediction based on machine-learning techniques and they did public sentiment analysis also. For sentiment analysis, they used Twitter and Reddit posts. They could use Facebook and LinkedIn data for better public opinion collection. The root-meansquare-error of the LSMT model with a single feature was 198.448 and with multiple features was 197.515 whereas the root-mean-square-error of the ARIMA model was 209.263 which states that the LSTM model performed accurately. The result shows that the LSTM algorithm predicts more accurately than the ARIMA algorithm for a long span. (Raju & Tarif, 2020)

The authors predicted the rates of three CTCs, namely Ethereum, BTC, and Litecoin. To predict cryptocurrency prices, they used machine learning algorithms like gated recurrent unit (GRU), LSTM, and Bi-LSTM. From the results of all three algorithms, they found that the GRU model predicts more accurately when compared with the LSTM and Bi-LSTM models. GRU presents MAPE per. of 0.24, 0.8, and 0.2 approx. for Bitcoin, Litecoin, and Ethereum respectively. BiLSTM presents MAPE per. of 5.99, 6.85, and 2.33 for Bitcoin, Litecoin, and Ethereum respectively. According to the authors, the GRU model is best for forecasting cryptocurrencies. (Hamayel & Owda, 2021)

To predict Bitcoin prices author uses data from the Bitcoin Price Index. They used machine learning techniqueslike Bayesian optimized recurrent neural network (RNN) and a Long-ShortTerm-Memory (LSTM) network. In this study, the authors found that the LSTM model performed well. The LSTM model achieves the highest accuracy of 52 percent. And RMSE was 8 percent of the LSTM model. According to the authors, the ARIMA model performed poorly when compared with both deep learning models. The LSTM model outperformed the RNN model. However, the LSTM takes more time to train compared to the RNN. (McNally et al., 2018)

By using deep learning and machine learning techniques like LSTM, ARIMA, XGBoost, and Prophet authors predict Bitcoin prices. After Building models, authors compared metrics like Root-Mean-Square-Error, Mean-Absolute-Error, and R2. The authors also used sentiment analysis combined with the LTMS model to predict Bitcoin prices. After comparing the metrics of all models, they concluded that the LTMS model with sentiment analysis performed better than others. (Ramani et al., 2023) Lots of work done in this domain. Still, a few challenges are there in this domain.

Lots of external environmental factors and sentiment affect Bitcoin prices. Some of the authors included sentiment analysis by using posts on social media. (Mishal et al., 2022) However, posts on social media may be biased. So, it is required to build a hybrid LTMS model with sentiment analysis which could use authentic posts for sentiment analysis. While predicting Bitcoin prices volume of Bitcoin trading a day should be considered it will improve prediction accuracy.

3. Aim and Objectives

The main aim of this research is to predict Bitcoin prices by using time series forecasting techniques. The goal of this research is to obtain a machine-learning algorithm model with high prediction accuracy that helps investors invest money in cryptocurrency or stocks.

The objectives of the research are outlined as follows.

- To analyze the hidden patterns in the Bitcoin dataset via EDA and visualization for a better understanding of the data trend.
- To determine the optimum pre-processing technique required for the LSTM and ARIMA model.
- To propose LSTM and ARIMA models to forecast Bitcoin price.
- To evaluate and compare the performance of both models based on evaluation metrics.

4. Significance of Study

A Bitcoin price-predicting model is useful to investors or stakeholders in the cryptocurrency space. It can provide information to investors so they can form their decision about buying, selling, or holding Bitcoin. Accordingly, they can change their investment strategies. This model can anticipate potential market fluctuation so investors can minimize their risk. It can provide optimal entry and exit points, which can enhance investor's strategies. Financial institutes can use prediction model to regulate their policies, and they can form cryptocurrencyrelated policies. The Bitcoin price prediction model can contribute to the overall growth and stability of the cryptocurrency world. The public can think the Bitcoin as an investment plan if the model predicts Bitcoin price accurately.

In this research, I am going to compare Machine learning algorithms like ARIMA and LSTM models. By evaluating the results of both models using various metrics, I can conclude which model predicts Bitcoin prices accurately. So people can use the perfect model while predicting cryptocurrency.

5. Scope of Study

5.1 In-scope of study

I am going to use historical data for predicting this model, so the model can capture trends, patterns, and seasonality in the Bitcoin prices. This study can provide short to medium-span forecasting, it can provide predictions for the next day, week, or month. It could be helpful for intra-day trading because it may provide optimal entry and exit points within a day. This study can help to assess the volatility of cryptocurrency prices over time.

5.2 Out-scope of study

This study may not be accurate for long-term prediction, especially when it tries to predict events that are out of the scope of historical data. There may be some unpredicted events or unexpected market shifts that are not present in train or historical data. This study may not work properly with the unpredictable nature of the financial market also called black swan events. This model may not cover global political factors and investor sentiments. This study may not cover price fluctuation because of Artificial or manual manipulation in the market.

6. Research Methodology

6.1 Workflow

Data Reading and Understanding → Exploratory Data Analysis and Visualization → Data Manipulation and Pre-Processing → Data Transformation and Augmentation → Building ARIMA and LSTM Models → Bitcoin Price Prediction → Evaluation and Comparison.

6.2 Dataset

I am going to use the Bitcoin dataset which is publicly available on the Kaggle website. The duration of the BTC-USD dataset is from 17/09/2014 to 25/03/2022. Prices in BTC_USD data are available in the form of US Dollars. The given data points are daily. There are a total of 2747 data points available with seven columns. The first column is about the date of each data point. The second column is open i.e., the opening price of BTC on each day. The third column is High which states the maximum price for each day. The 4 th column states the minimum price of each day. The 5 th column is about the Closing price of BTC on each day. The 6 th column is about the adj. closing price and the Seventh column is about the number of BTC traded every day. Therefore, this dataset provides open, high, low, and close (OHLC) data on Bitcoin. I am going to use the same dataset for both the LSTM and ARIMA models. Also, I am going to use the closing price of Bitcoin as the target variable.

6.3 Exploratory Data Analysis and Visualization

In this, I am going to explore the Bitcoin data. By using Python Notebook, we can explore facts like Information about the data and descriptions of the data. Also, I am going to find missing values, outliers, data duplicates, Nan values, and data types of columns. If there is any outlier then it is good practice to cap it rather than eliminating such records. I am going to treat missing values and Nan values based on mean, mode, and median as the case. For visualization, I am going to use both Matplotlib and Seaborn libraries. When I require an attractive and detailed chart, I will use Seaborn otherwise I will use Matplotlib.

6.4 Data Pre-Processing and Transformation

6.4.1 Data Pre-Processing

I am going to split the data sequentially not random. Because it is time series data in which the next data point is related to the previous one. We can split data as 60/40, 70/30, 80/20, or 90/10. I would prefer an 80/20 split. Where training data is 80% and 20% is testing

6.4.2 Data Transformation

Min-max scaling is a common technique used in machine learning, including in LSTM (Long Short-Term Memory) models, to normalize input data. Normalization is important because it helps improve the performance and training stability of neural networks. I am going to use minmax-scaling because it gives a consistent input range, faster convergence, numerical stability, improved generalization, and better weight initialization.

LSTMs expect a 3D input shape for the input data. The dimensions of this 3D shape are typically: (number of samples, time steps, number of features). So, it is necessary to convert the 2D data (like a time series or a sequence of observations) into the required 3D format by just reshaping the data. I will use functions like 'reshape' in Python.

6.5 ARIMA and LSTM Model

6.5.1 ARIMA Model

ARIMA stands for Auto-Regressive Integrated Moving Average. It is a popular and widely used time series predicting model that uses autoregression, differencing, and moving averages (MA) components. I chose the ARIMA model because of its Flexibility, Simple Interpretability, Effective handling of trends and seasonability, well-established statistical principles, forecasting accuracy, and availability of software.

6.5.2 LSTM Model

The long form of LSTM is Long Short-Term Memory, and it's a part of the recurrent neural network framework. LSTM models are designed to address some of the limitations of traditional recurrent neural networks in capturing and learning long-span dependencies in cyclic data.

I chose the LSTM Model for Bitcoin prediction because it captures long-term dependencies, handles exploding gradient problem, learns hidden patterns efficiently, does parallel processing, adapt different time lags, handles irregularity of sample effectively, does not require manual featuring, and can handle multivariate features. So, I prefer the LSTM model rather than FbProphet, and other time series forecasting algorithms

6.6 Evaluation Metrics

When evaluating LSTM (Long Short-Term Memory) models, ARIMA (Auto-Regressive Integrated Moving Average), or any time series forecasting algorithm, it's essential to use appropriate metrics to assess their performance. Some commonly used evaluation metrics for Time series forecasting models are:

- Mean Absolute Error : Represents the avg. absolute diff. between the predicted values and the actual values.
- Mean Squared Error : Measures the avg. of the squared gap between predicted and actual values.
- Root Mean Squared Error : It is the sqrt of the MSE and provides a measure of the avg. of the errors in the units as the original data.
- Mean Absolute Percentage Error : It expresses the average per. diff. between the forecasted and actual values.
- R2 Score : It measures the proportion of the variance in the target variable (target) that is predictable from the independent variables (predictions).

I am going to compare the ARIMA and LSTM Models based on mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MSPE), and based on the accuracy of the model. Because they are commonly used metrics that can be employed to assess the performance of time series predicting algorithms. It is always good to use multiple metrics for better understanding and better comparison between two algorithms. So, I decided to use multiple evaluation metrics for comparing models.

7. Required Resources

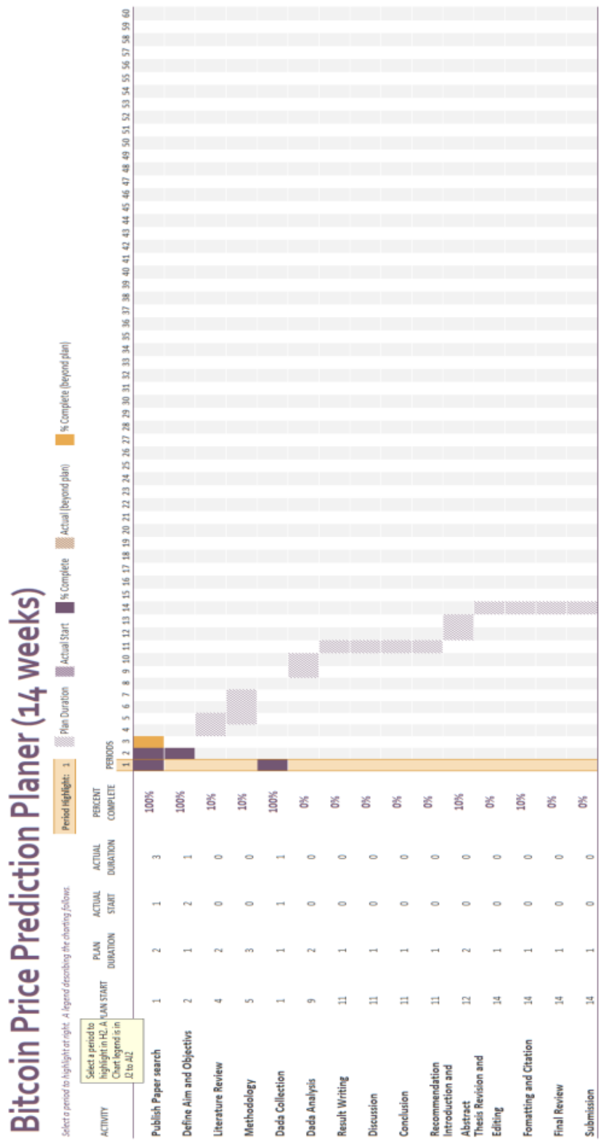
7.1 Hardware Requirement

- Central Processing Unit (CPU): A standard multicore CPU
- Graphics Processing Unit (GPU): Graphics like NVIDIA
- Random Access Memory (RAM): At least 4 GB RAM

7.2 Software Requirement

- Python with the updated version.
- Deep Learning Frameworks like TensorFlow and Keras.
- Jupyter Notebook with the updated version.
- Libraries like Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, Math, Stat
- Cloud-based platforms like Google Colab

8. Research Plan



References

Dinshaw, C., Jain, R., & Hussain, S. A. I. (2022). Statistical Scrutiny of the Prediction Capability of Different Time Series Machine Learning Models in Forecasting Bitcoin Prices. Proceedings of 4th International Conference on Cybernetics, Cognition and Machine Learning Applications, ICCCMLA 2022, Ml, 329–336. https://doi.org/10.1109/ICCCMLA56841.2022.9989057

Hamayel, M. J., & Owda, A. Y. (2021). A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms. AI (Switzerland), 2(4), 477–496. https://doi.org/10.3390/ai2040030

Hua, Y. (2020). Bitcoin price prediction using ARIMA and LSTM. E3S Web of Conferences, 218. https://doi.org/10.1051/e3sconf/202021801050

Huang, H., & Ye, T. (2022). Asset Trading Strategies Based on LSTM - Bitcoin and Gold as an Example. CAIBDA 2022 - 2nd International Conference on Artificial Intelligence, Big Data and Algorithms, 650–655.

Latif, N., Selvam, J. D., Kapse, M., Sharma, V., & Mahajan, V. (2023). Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices. Australasian Accounting, Business and Finance Journal, 17(1), 256–276. https://doi.org/10.14453/aabfj.v17i1.15

McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018, 339–343. https://doi.org/10.1109/PDP2018.2018.00060

Mishal, M. H., Jannat Rakhi, N., Rashid, F., Hamid, K., Morol, M. K., Al Jubair, A., & Nandi, D. (2022). Prediction of Cryptocurrency Price using Machine Learning Techniques and Public Sentiment Analysis. Proceedings of 2022 25th International Conference on Computer and Information Technology, ICCIT 2022, 657–662. https://doi.org/10.1109/ICCIT57492.2022.10055524

Raju, S. M., & Tarif, A. M. (2020). Real-Time Prediction of BITCOIN Price using Machine Learning Techniques and Public Sentiment Analysis. http://arxiv.org/abs/2006.14473
Ramani, K., Jahnavi, M., Reddy, P. J., Venkatachakravarthi, P., Meghanath, P., & Imran, S. K. (2023). Prediction of Bitcoin Price through LSTM, ARIMA, XGBoost, Prophet and Sentiment Analysis on Dynamic Streaming Data. 2023 9th International Conference on Advanced Computing and Communication Systems, ICACCS 2023, 1, 1514–1518. https://doi.org/10.1109/ICACCS57279.2023.10113014

Sener, E., & Demir, I. (2022). Wavelet-Denoised ARIMA and LSTM Method to Predict Cryptocurrency Price. ISMSIT 2022 - 6th International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings, 577–582. https://doi.org/10.1109/ISMSIT56059.2022.9932727

Final Thesis Report.pdf

ORIGINALITY REPORT

0%	0%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	On		