University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project
7PAM2002-0509-2022

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Comparative Analysis of Machine Learning Models for Cryptocurrency Price Prediction

Student Name and SRN:

Hetanshi Kachhiya Patel (21067973)

Supervisor: Vito Graffagnino

Date Submitted: 27th August 2023

Word Count: 9680

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Hetanshi Kachhiya Patel

Student Name signature:

Student SRN number: 21067973

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

This study uses historical data from Yahoo Finance to conduct a comparison examination of three machine learning models for forecasting the price trends of Bitcoin and Ethereum: LSTM, Random Forest Regression, and Bi-LSTM. The goal is to identify the model that can best capture the complex price movements of several cryptocurrencies. Model performance is evaluated using measures like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared values and Mean Absolute Error (MAE). The outcomes clearly show that the Random Forest Regressor is the best option for predicting the prices of Bitcoin and Ethereum, with impressively low RMSE values of 0.0132 and 0.0113, respectively. This demonstrates how accurate the model is at predicting cryptocurrency prices over time. The study's implications extend to enhancing comprehension of machine learning's significance in cryptocurrency price prediction, offering valuable guidance to traders and researchers seeking adept models for these highly dynamic markets.

ACKNOWLEDGEMENTS

I extend my heartfelt appreciation to my respected supervisor, Mr. Vito Graffagnino, whose insightful guidance and mentorship have been integral to the successful completion of this project. His expertise and unwavering support have illuminated my path and enriched my understanding.

Furthermore, I am deeply thankful to my friends and family for their unending encouragement and belief in my capabilities. Their motivation and positivity have been a driving force, propelling me forward in my pursuit of knowledge.

This project would not have been possible without the collaborative efforts of various individuals who have contributed their insights and time. Their contributions, whether big or small, have collectively shaped the project's trajectory.

In conclusion, I am indebted to my supervisor, peers, friends, and family for their continuous encouragement and guidance, enabling me to embark on this academic endeavour with enthusiasm and determination.

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 Aims and Objectives

The main goal of this research is to thoroughly compare the forecasting skills of several machine learning models in the field of predicting bitcoin prices. The goals include:

1. To carefully evaluate and establish the reliability and accuracy of three different models, LSTM, Bi-LSTM, and Random Forest Regression, in predicting price changes for two well-known cryptocurrencies, namely Bitcoin and Ethereum, based on historical price data.

2. To conduct a thorough analysis of the inherent benefits and potential drawbacks of each model, particularly regarding how well they handle the complexities and difficulties that cryptocurrency price data presents, such as complex non-linear patterns and unexpected, sudden price changes.

3. Using a complete collection of performance indicators, including Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared values, to conduct a thorough evaluation of the prediction ability of these models. This project aims to give a comprehensive knowledge of their relative strengths and weaknesses when applied to both Bitcoin and Ethereum.

4. To conduct a thorough comparison of these models, considering how well each one performed across Bitcoin and Ethereum. This comparative viewpoint should shed light on the models' capacity to adjust to various cryptocurrencies and their unique market behaviours.

1.2 Significance of the Study

The importance of this work lies in its contribution to the field of cryptocurrency price prediction, which addresses the critical need for precise forecasting in a rapidly changing and volatile market. This research offers helpful insights to investors, traders, and academics by thoroughly evaluating the effectiveness of machine learning models in forecasting the price fluctuations of cryptocurrencies. The results provide informed decision-making and risk management by assisting in the identification of the best reliable model for forecasting bitcoin price movements. The study advances our understanding of how machine learning approaches can be used to manage the complexity of cryptocurrency pricing data. The findings will eventually have implications for improving financial strategy precision, maximising possibilities for increased income, and reducing risk in the cryptocurrency market.

1.3 Research Questions

The following research questions guide this investigation:

1. How accurately can LSTM, Bi-LSTM, and Random Forest Regression models predict the price movements of Bitcoin and Ethereum?
2. What are the strengths and limitations of each model in handling the complexities of cryptocurrency price data?
3. How do the predictive performances of these models compare in terms of accuracy, Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared values?

CHAPTER 2

BACKGROUND

2.1 Background Study

According to Wikipedia Contributors (2019) Cryptocurrency stands as a form of digital or virtual currency that employs cryptography to ensure secure transactions and functions on a decentralized network. Unlike conventional currencies, which are issued by governments and central banks, cryptocurrencies generally operate without regulation from a central authority. Instead, they rely on cryptographic methods to safeguard transactions and manage the creation of new units. Cryptocurrency is intangible and doesn't have a physical presence, often lacking centralized control. It can take a centralized form if introduced by a single entity or embrace decentralization through technologies like blockchain. Bitcoin, introduced in 2009, holds the distinction of being the inaugural cryptocurrency. Presently, there are over 9,000 different cryptocurrencies, with more than 70 of them boasting market capitalizations exceeding $1 billion as of March 2022. (Wikipedia Contributors 2019).

2.1.1 Factors That Influence the Price of Crypto:

The complex and dynamic world of cryptocurrencies is influenced by a multitude of factors that collectively shape their market prices. Understanding these factors is essential for investors and enthusiasts seeking insights into the often-volatile nature of cryptocurrency markets.

1. **Supply and Demand:** Analogous to traditional economics, the fundamental forces of supply and demand play a crucial role in determining cryptocurrency prices. When demand exceeds supply, prices tend to surge, and when supply surpasses demand, prices may dip. This delicate balance is central to market dynamics (Bloomenthal, 2019).
2. **Regulations:** (Auer and Claessens, 2018) The regulatory landscape significantly impacts cryptocurrency markets. Favourable regulations, such as the introduction of exchange-traded funds (ETFs) and futures contracts, can bolster demand and consequently drive prices upward. Conversely, adverse regulatory measures can lead to price declines.
3. **Macro Factors:** Cryptocurrency prices are not isolated from the broader economy. Economic prosperity and an appetite for risk can fuel increased investment in cryptocurrencies. Conversely, economic recessions can result in reduced allocations to such assets due to risk aversion.
4. **Sentiment:** (Mai et al., 2018) The collective sentiment of the public, as shaped by media, social platforms, and news outlets, plays a pivotal role in influencing cryptocurrency prices. Positive sentiment can trigger FOMO (fear of missing out), driving demand, while negative sentiment may lead to selling pressure (Beat, 2023).
5. **Internal Governance:** (Hsieh, JP) Vergne and Wang, 2017) Cryptocurrencies often rely on decentralized governance models. Decisions on protocol upgrades and changes are made through community consensus. Efficient governance can stabilize the market, whereas disagreements and delays in protocol advancements can impact prices.
6. **Technological Advancements:** Innovations in (Gad et al., 2022) blockchain technology, scalability solutions, security enhancements, and utility improvements can drive

investor confidence and increase demand for specific cryptocurrencies, subsequently impacting their prices.

7. **External Events:** Global events, such as geopolitical tensions, financial crises, and major technological breakthroughs, can indirectly influence cryptocurrency prices by affecting investor sentiment and risk appetite.

Overall, the dynamic nature of cryptocurrency prices is shaped by a complex interplay of these factors, making their value highly volatile and subject to sudden changes.

2.1.2 Machine Learning Models for Cryptocurrency Price Prediction

According to (Murray et al., 2023) Cryptocurrency price prediction is a complex task that involves analysing historical data to forecast future price trends. Various machine learning models have been employed to tackle this challenge and provide insights into the volatile cryptocurrency market. The most well-known machine learning models for predicting bitcoin prices are listed below:

1. **Linear Regression:** Simple linear regression uses one independent variable, such as the opening price, to predict a dependent variable (such as the closing price of a cryptocurrency), whereas multiple linear regression considers many independent variables (Uras et al., 2020).

2. **ARIMA (AutoRegressive Integrated Moving Average):** To estimate future price movements, ARIMA is a time series forecasting model that combines auto-regression, differencing, and moving average components (Hayes, 2022).

3. **LSTM (Long Short-Term Memory):** Long Short-Term Memory (LSTM), an artificial recurrent neural network, excels in learning sequential data due to its feedback links, differing from feed-forward networks. Ideal for time-series analysis and tasks involving sequential formats like language and speech processing, LSTM is pivotal in deep learning (Sarker, 2021).

4. **GRU (Gated Recurrent Unit):** Like LSTM, GRU is a recurrent neural network variation that is particularly good at detecting temporal connections in sequential data (Zhang, Zeng and Starly, 2021).

5. **Random Forest Regression:** This ensemble learning technique uses several decision trees to improve the predictability of results, making it helpful for modelling intricate relationships in bitcoin data (Jaquart, Dann and Weinhardt, 2021).

6. **XGBoost:** Gradient Boosting, akin to Random Forests, is an ensemble method creating a robust model from multiple models (often decision trees). It optimizes through gradient to minimize loss, akin to neural networks' gradient descent. Extreme Gradient Boosting (XGBoost) refines this by considering detailed approximations, computing second-order gradients for better loss reduction and applying advanced regularization

(L1 and L2) to curb overfitting, elevating generalization and performance. XGBoost efficiently manages large datasets (Sarker, 2021).

7. **Support Vector Machine (SVM):** Support Vector Machines (SVM) are versatile in classification and regression, constructing hyper-planes for strong separation in high-dimensional space. Kernel functions like linear, polynomial, and RBF enhance performance, yet SVM struggles with noisy data and overlapping classes (Sarker, 2021).
8. **Prophet:** Prophet is a forecasting tool created by Facebook that is especially suited for time series data. It takes seasonality, holidays, and patterns into consideration while analysing cryptocurrency price data (Rathore et al., 2022).
9. **GANs (Generative Adversarial Networks):** GANs produce synthetic data that closely resembles actual pricing data, assisting analysts in understanding anticipated market changes (Figueira and Vaz, 2022).
10. **Bidirectional LSTM (Bi-LSTM):** A variation of LSTM that considers both past and future data points in order to capture bidirectional temporal dependencies and enhance the capabilities of sequence modelling (Wang et al., 2019).

Understanding the variables that affect a cryptocurrency's value and using powerful machine learning models hold the possibility of revealing fresh insights into their complicated dynamics in this continuously changing cryptocurrency ecosystem. As we go more into the core of our research, we seek to illuminate the complex interplay between underlying concepts, technological developments, macroeconomic factors, and sentiment-driven actions that collectively define the cryptocurrency world. We set out on a quest to solve the puzzles of cryptocurrency price fluctuations by investigating a variety of predictive models, paving the way for well-informed choices in this dynamic and ever-expanding domain.

2.1.3 Cryptocurrency Evolution: Bitcoin and Ethereum Journey

The first cryptocurrency, ***Bitcoin***, was released in 2009 by a person or group using the alias Satoshi Nakamoto (The Economist, 2015). Bitcoin, which acts as digital gold, mines its 21 million coins by computer algorithms and runs on a decentralised network. Its adoption as a store of value and alternative investment is fuelled by the breakthrough blockchain technology that it uses to facilitate safe and transparent transactions.

The 2015 release of ***Ethereum***, which emphasised smart contracts and decentralised applications (dApps), revolutionised the industry. Its creator, Vitalik Buterin, had a vision for a platform where programmers could build a variety of blockchain applications. Ether (ETH), a cryptocurrency introduced by Ethereum, functions as a store of value as well as a utility token for dApps (Binance Feed, n.d.). The framework for a wider adoption of blockchain technology in industries other than banking was created by Ethereum's novel strategy.

2.2 Exploring Relevant Literature in the Field

1. (Murray et al., 2023) address the pivotal challenge of forecasting cryptocurrency prices by presenting a comprehensive and comparative framework. Cryptocurrencies' dynamic, volatile, and uncertain nature makes price prediction a complex time series analysis endeavour. Existing research, though valuable, has limitations such as narrow scope, scattered methodologies, and lack of generality. To overcome these shortcomings, the authors introduce a unified comparison framework that facilitates an extensive evaluation of various approaches. Their study delves into a range of statistical, machine learning (ML), and deep learning (DL) techniques to predict prices for five prominent cryptocurrencies: XRP, Bitcoin (BTC), Litecoin (LTC), Ethereum (ETH), and Monero (XMR). Notably, they introduce the temporal fusion transformer (TFT) for this task, a novel addition to the field. Their research also extends its exploration to hybrid models and ensembles, investigating the potential gains from model combinations. Through rigorous evaluation, Murray et al. (2023) demonstrates that deep learning approaches, particularly the Long Short-Term Memory (LSTM) model, outshine other methods consistently across all cryptocurrencies examined. LSTM achieves an average Root Mean Squared Error (RMSE) of 0.0222 and Mean Absolute Error (MAE) of 0.0173, presenting a notable improvement over alternative model. This clear superiority underscores the efficacy of DL techniques in cryptocurrency price prediction. The significance of this study extends beyond its findings. To foster reproducibility and encourage future research contributions, the authors generously share their dataset and experiment code. By addressing existing limitations and offering actionable insights, Murray et al. (2023) elevates the field of cryptocurrency price prediction, providing a valuable resource for traders, investors, and researchers alike.

2. McNally, S., Roche, J., and Caton, S. (2018) presented a study to evaluate the effectiveness of predicting the direction of Bitcoin's USD price movement. Data is extracted from the Bitcoin Price Index, and prediction models are developed, including a Bayesian-optimized recurrent neural network (RNN) and a Long Short-Term Memory (LSTM) network. The LSTM model achieves the highest accuracy at 52% and a root mean squared error (RMSE) of 8%. The research also includes a comparison with the ARIMA model for time series forecasting, highlighting the superiority of the non-linear deep learning approaches over ARIMA. Furthermore, the deep learning models are tested on both GPU and CPU platforms, with the GPU training outperforming CPU

training by 67.7%. This comparative analysis provides insights into the predictive capabilities of different models for Bitcoin price forecasting.

3. Seabe, Moutsinga and Pindza, (2023) conducted a study which tackles the problem of precisely forecasting cryptocurrency values considering their nonlinear behaviour. To predict the exchange rates of the three biggest cryptocurrencies (BTC, ETH, and LTC), the study uses three different types of recurrent neural networks (RNNs): LSTM, GRU, and Bi-LSTM. According to the data, Bi-LSTM surpasses LSTM and GRU and has the best predictive accuracy, with MAPE values for BTC, LTC, and ETH of 0.036, 0.041, and 0.124, respectively. The results imply that the suggested prediction models are useful resources for traders and investors looking for information about fluctuations in bitcoin prices. The report also recommends more research into other elements that may affect cryptocurrency pricing, such as social media and trade volume. This research helps with ongoing efforts to improve the precision of bitcoin price predictions.

4. In a recent study by Chen (2023), the primary focus is on crafting precise prediction models for Bitcoin's next-day price. This endeavour employs both random forest regression and LSTM algorithms, while also delving into the identification of influential variables shaping Bitcoin's price trends. Departing from earlier research which predominantly leaned on ARMA models and LSTM, this study introduces the innovative dimension of random forest regression. While the discerning Diebold-Mariano test refrains from explicitly favouring either methodology, the outcomes notably highlight the superior performance of random forest regression in terms of RMSE and MAPE metrics. Intriguingly, the investigation reveals a dynamic interplay of variables influencing Bitcoin's price from 2015 to 2018, encompassing factors like US stock market indexes, oil prices, and ETH price. The evolution post-2018 underscores the ascension in significance of ETH price and the Japanese stock market index, JP225. Examining precision concerning the time horizon, the model with a single lag of explanatory variables emerges as the most adept in predicting Bitcoin's price for the subsequent day.

5. The paper by (Sebastião and Godinho, 2021) delves into an extensive examination of the predictability of three prominent cryptocurrencies: Bitcoin, Ethereum, and Litecoin. Furthermore, it delves into the creation and testing of trading strategies based on machine learning methodologies, including linear models, random forests, and support vector machines. The study's validation phase occurred during a period marked by unprecedented market volatility, while its testing phase took place in a bear market context. This design enables the assessment of the predictive models' resilience and their adaptability to shifting market dynamics. By leveraging attributes from both trading data and network activity spanning August 15, 2015, to March 03, 2019, the study conducts classification and regression analyses. To evaluate the generalization of the models, the testing phase starts on April 13, 2018. Notably,

results from the test period indicate that approximately 28% of the individual models achieved success rates below 50%. The research extends to constructing trading strategies founded on the ensemble of these models. An intriguing outcome emerges: the ensemble strategy, denoted as Ensemble 5, yields the most promising results for Ethereum and Litecoin. It showcases annualized Sharpe ratios of 80.17% and 91.35% and annualized returns (factoring in trading costs of 0.5%) of 9.62% and 5.73%, respectively. These compelling outcomes validate the notion that machine learning techniques offer robust tools to navigate the predictability of cryptocurrencies. Furthermore, the study underscores the feasibility of crafting profitable trading strategies in these dynamic markets, even amid adverse market conditions.

6. (Khedr et al., 2021) conducted a comprehensive review of research in the field of cryptocurrency price prediction from 2010 to 2020. Cryptocurrencies, exemplified by bitcoin, have transformed the landscape of decentralized electronic money, enabling secure and anonymous transactions over the internet. The decentralized nature of cryptocurrencies has disrupted traditional centralized control and its associated implications for global trade. However, the inherent price volatility of cryptocurrencies necessitates accurate prediction models. The dynamic field of cryptocurrency price prediction draws upon a diverse range of statistical and machine-learning techniques. Traditional statistical methods like Bayesian regression, logistic regression, and linear regression have been employed in cryptocurrency price prediction. Yet, the unique absence of clear seasonal effects in cryptocurrency markets poses challenges for these approaches. In contrast, machine learning techniques such as support vector machines, artificial neural networks, deep learning, and reinforcement learning have emerged as promising tools to address this challenge. Harnessing experiential insights, machine learning demonstrates a superior ability to predict cryptocurrency prices compared to traditional statistical approaches. Khedr et al. (2021)'s work provides an encompassing review of research efforts spanning the decade from 2010 to 2020 in the realm of cryptocurrency price prediction. It captures the evolving methodologies and findings in this dynamic domain, shedding light on the trajectory of advancements. By amalgamating the collective insights from past studies, this article bridges existing gaps and enriches researchers' understanding of cryptocurrency price prediction. The insights provided by Khedr et al. (2021) inform and guide future investigations in this vital and ever-evolving field.

7. According to the study conducted by Hamayel and Owda (2021), cryptocurrencies have emerged as a novel asset class due to advancements in financial technology, offering vast research opportunities. Their price forecasting, however, is complex due to their inherent volatility and dynamic nature. The proliferation of numerous cryptocurrencies globally intensifies this challenge. In this context, the study proposes a comprehensive approach employing recurrent neural networks (RNNs) to predict the prices of three prominent cryptocurrencies: Bitcoin (BTC), Litecoin (LTC), and Ethereum

(ETH) (Hamayel & Owda, 2021). Results from this research showcase the predictive prowess of the GRU algorithm, which outperforms both LSTM and bi-LSTM models in price forecasting across all cryptocurrency types. The accuracy of the prediction is notably measured using the mean absolute percentage error (MAPE). Specifically, the GRU algorithm achieves exceptional prediction accuracy for LTC, boasting MAPE percentages of 0.2116%, 0.8267%, and 0.2454% for BTC, ETH, and LTC, respectively. Comparatively, the bi-LSTM algorithm presents the least accurate predictions, yielding MAPE percentages of 2.332%, 6.85%, and 5.990% for BTC, ETH, and LTC, respectively. These prediction models hold significant economic implications by providing invaluable insights for investors and traders in navigating the volatile cryptocurrency market. Their study's findings underscore the relevance of employing advanced machine learning techniques for accurate price prediction. However, the research's call to explore additional variables, including social media impact, tweet sentiments, and trading volume, highlights the ever-evolving nature of cryptocurrency pricing and the need for comprehensive models. In conclusion, Hamayel and Owda (2021) contribute essential insights to the realm of cryptocurrency price forecasting by demonstrating the efficacy of RNN algorithms in predicting the prices of major cryptocurrencies. Their findings not only expand the understanding of cryptocurrency dynamics but also lay the groundwork for future investigations into additional factors influencing cryptocurrency markets.

8. (Rathore et al., 2022) Researched on cryptocurrency price prediction has led to the emergence of new digital asset models catalysed by advancements in financial technology. This avenue of exploration has given rise to various methodologies, including those leveraging algorithms like LSTM and ARIMA. However, the inherent complexity of LSTM-based RNNs often renders them less interpretable, making it a challenge to intuitively decipher their behaviours. The intricate process of achieving optimal results also demands meticulous hyperparameter tuning. Further exacerbating the situation, the cryptocurrency domain deviates from traditional data patterns due to its swift and unpredictable market dynamics. Cryptocurrency markets are inherently characterized by volatility and rapid shifts, posing unique challenges to price prediction. The traditional ARIMA model, designed for handling seasonal data, grapples with the dynamic nature of cryptocurrency datasets. Addressing these complexities and limitations requires innovative approaches. Putting these things into account, this study proposes the application of the Fbprophet model as a pivotal solution, capitalizing on its superior performance in comparison to LSTM and ARIMA models. By doing so, the study seeks to overcome the inherent shortcomings of LSTM and ARIMA approaches in the context of cryptocurrency analysis (Rathore et al., 2022). Notably, the Fbprophet model presents a holistic strategy that circumvents the challenges posed by these traditional methods, particularly when dealing with scenarios characterized by limited historical data and elusive patterns, common within the cryptocurrency domain. The study outlines a comprehensive methodology designed to predict future bitcoin prices, acknowledging the presence of seasonality in historical data. The proposed model is constructed through the integration of seasonality fitting and smoothing techniques, rendering it applicable for real-world

use cases. This methodology is particularly potent when addressing scenarios with limited historical data and challenging pattern recognition, often encountered in the cryptocurrency landscape. A key finding of this research is the notable reduction in the divergence between predicted and actual values, even when faced with complex seasonal data. The proposed methodology showcases exceptional performance in minimizing disparities compared to alternative models. Importantly, this robust performance is upheld despite the intricate dynamics of cryptocurrency markets and the scarcity of historical data. By harnessing the unique strengths of the Fbprophet model and tailoring it to the idiosyncrasies of cryptocurrency data, this study offers a significant advancement in enhancing price prediction accuracy and real-world utility.

9. (Ozer and Okan Sakar, 2022) The cryptocurrency market, characterized by its rapidly growing market size, garners escalating attention from both individual and institutional investors. While this notably volatile market offers substantial profit potential, it simultaneously exposes investors to risks due to its sensitivity to speculative news and the capricious behaviour of major investors, capable of inciting unusual price fluctuations. In that paper, the argument posited is that rapid and pronounced price oscillations or unconventional patterns, stemming from these factors, can adversely impact the efficacy of technical signals forming the basis of feature extraction within a machine learning (ML)-based trading system. This, in turn, could result in a deterioration of model generalization. To grapple with this predicament, an all-encompassing ML-based trading system is proposed, which integrates a time series outlier detection module. The role of this module is to identify periods characterized by anomalous price formations. Subsequent training of classification algorithms for the price direction prediction task is executed using the remaining data. Their findings proffered encompass accuracy evaluation of the classification models, along with simulation outcomes derived from real-time trading on historical data, utilizing the proposed system. Their research outcomes underscore that the inclusion of the outlier detection step significantly enhances returns on investment for machine learning-based trading strategies. Additionally, the results illuminate that, particularly during periods marked by heightened volatility, the trading system boasts augmented profitability compared to both the baseline model and the buy-and-hold strategy.

CHAPTER 3

THE DATASET

The five-year data set used in this study, which began on June 1, 2018, and ended on May 31, 2023, was obtained from Yahoo Finance. It includes historical information about Bitcoin and Ethereum, concentrating on their daily pricing.

The dataset includes following columns:

1. **Date:** Identifies the precise date that corresponds to the pricing data that was captured.
2. **Open:** Indicates the cryptocurrency's opening price for a particular day.
3. **High:** This value represents the cryptocurrency's highest price for that day.
4. **Low:** Indicates the cryptocurrency's lowest price of the day.
5. **Close:** Shows the cryptocurrency's ultimate price at the end of the day.
6. **Volume:** Indicates the cryptocurrency's trading volume on a given day.
7. **Adjusted Close:** The closing price after any corporate activities, such as stock splits or dividends, have been taken into consideration.

The dataset was carefully chosen since it was recent and relevant. It offers up-to-date statistics on Bitcoin and Ethereum values and is sourced from Yahoo Finance, a reliable source of financial data. The selection of Yahoo Finance guarantees data accuracy and dependability, which are essential for insightful analysis. The daily pricing information in the dataset, which includes open, high, low, and closing values, enables an extensive analysis of cryptocurrency behaviour. This dataset nicely complements the project's goal of examining current market trends in cryptocurrencies and comprehending their dynamics.

HEAD OF THE DATASET:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-06-01	5645.142090	5723.436523	5556.060547	5647.334473	5647.334473	3685382825
1	2018-06-02	5643.792480	5762.939941	5614.242676	5723.715820	5723.715820	3698741830
2	2018-06-03	5715.208984	5807.166504	5700.943359	5782.092773	5782.092773	3633732870
3	2018-06-04	5783.799805	5799.850098	5610.298828	5644.036621	5644.036621	3750315397
4	2018-06-05	5633.844238	5706.032227	5557.893066	5695.135742	5695.135742	3701685897

Figure 1: Bitcoin

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-06-01	435.517426	443.126984	425.440277	434.359009	434.359009	1457158940
1	2018-06-02	434.648071	447.114746	432.321411	443.169067	443.169067	1408109942
2	2018-06-03	442.757965	467.659882	442.757965	463.098389	463.098389	1372491025
3	2018-06-04	463.928253	466.359222	438.248505	445.384552	445.384552	1429645498
4	2018-06-05	445.700775	456.012238	436.565979	454.568024	454.568024	1375910140

Figure 2: Ethereum

CHAPTER 4

METHODOLOGY

4.1 Brief Overview

This section provides a detailed account of the systematic methodology employed in conducting the comparative analysis of machine learning models for predicting cryptocurrency prices, specifically focusing on Bitcoin (BTC) and Ethereum (ETH). The process entails a comprehensive series of steps, encompassing data collection, preprocessing, feature engineering, model selection, training, validation, and evaluation.

4.2 Data Pre-processing

In the initial phase of my methodology, I began with data pre-processing to ensure that the dataset was well-suited for modelling using LSTM, Random Forest Regression, and Bi-LSTM. To accomplish this, I imported essential libraries to facilitate data manipulation and analysis. Following the import, I conducted a comprehensive data cleaning process, addressing potential outliers and checking for any missing values, even though my dataset was relatively clean in this regard.

Moving forward, I delved into data analysis, generating graphical representations to visualize the patterns of price fluctuations over time. I reshaped the data and used the MinMaxScaler to normalise features into a constant range, limiting the impact of magnitude changes on model training, in order to align the dataset for modelling. Additionally, I determined the number of prior time steps that go into price prediction by establishing the sequence length for input data. For the data to accurately capture temporal dependencies, making this choice is crucial. A proper balance for the model's evaluation was then achieved by splitting the dataset into training and test sets, with 90% allocated to training and 10% to testing.

4.3 Model Selection and Architecture

In the core of this study, I carefully opted for three key models: Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Random Forest Regression. I made this choice after thoroughly studying various research papers that demonstrated these models' suitability for dealing with time-series data and uncovering intricate patterns.

LSTM (Long Short-Term Memory):

LSTM is a type of recurrent neural network (RNN) known for its ability to capture intricate patterns in sequential data. It's widely used in tasks involving time series data, such as cryptocurrency price prediction (Sarker, 2021). The key feature of LSTM is its memory cell, which can retain information over long sequences, allowing it to learn and model complex relationships in data with time dependencies (Zhou, Zhang and Chen, 2021).

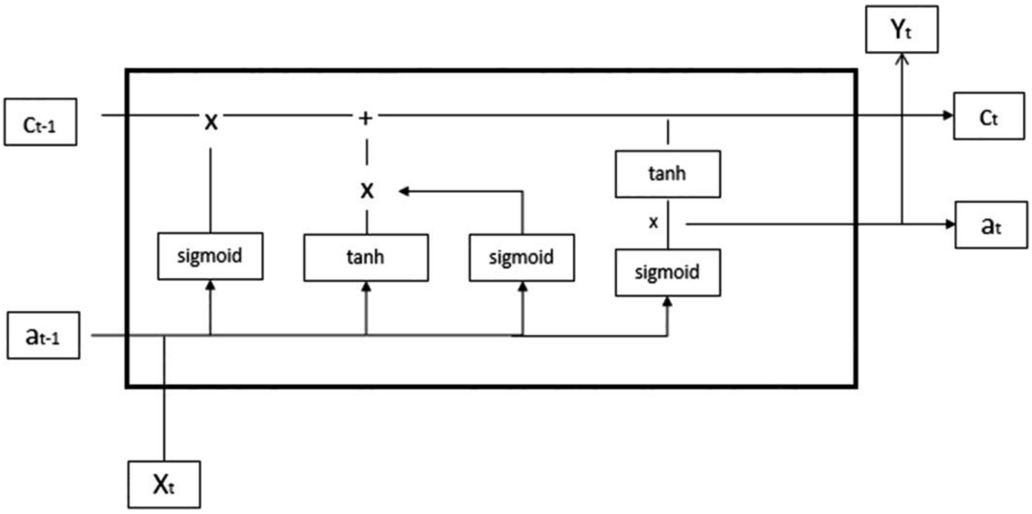

Figure 3: The LSTM cell (source: https://learning.oreilly.com/api/v2/epubs/urn:orm:book:9781484271506/files/images/508548_1_En_18_Chapter/508548_1_En_18_Fig1_HTML.jpg)

1. I proceeded with building LSTM models designed for predicting the prices of both Bitcoin and Ethereum. This is a pivotal stage in my project, aiming to develop accurate price forecasts. Leveraging the Keras library, I constructed these models with a customized structure.
2. I employed the Sequential model, a fundamental tool for creating deep learning models suitable for sequence prediction tasks like mine.
3. For each cryptocurrency, I started by adding an LSTM layer with a specified window size. This window size plays a crucial role in understanding the sequence patterns within the data. By setting 'return_sequences=True', I ensured that the output is generated for each time step, enabling the model to capture temporal relationships effectively.
4. Subsequently, I integrated dropout layers, which assist in reducing overfitting by randomly deactivating a proportion of neurons during training. This regularization technique aids in preventing the model from memorizing the training data too closely.
5. The subsequent LSTM layer, featuring double the window size, continued extracting intricate temporal patterns. This was followed by another dropout layer to maintain a well-generalized model.
6. In the final LSTM layer, with 'return_sequences=False', I anticipated the output solely for the final time step.
7. To complete the model architecture, I appended a Dense layer with a single unit, along with a linear activation function. This setup is adapted for regression tasks, which aligns with the goal of predicting numerical cryptocurrency prices.

8. Once the model architecture was established, I compiled it using the 'mean_squared_error' loss function and the 'adam' optimizer. This combination aids in fine-tuning the model's predictions and minimizing the gap between predicted and actual values.

Bi-LSTM (Bidirectional Long Short-Term Memory):

Bi-LSTM is an extension of LSTM that enhances its predictive power. It processes data sequences in both forward and backward directions, enabling the model to capture context from both past and future data points (Wang et al., 2019). This bidirectional approach is advantageous for tasks where understanding context in both temporal directions is vital, such as predicting cryptocurrency price movements accurately.

- I designed a Sequential model for both Bitcoin and Ethereum using the Bidirectional Long Short-Term Memory (Bi-LSTM) architecture. To start, I configured the model's architecture step by step. Firstly, I incorporated a Bidirectional LSTM layer with 'WINDOW_SIZE' units, set to return sequences, and specified the input shape as (WINDOW_SIZE, X_train's feature dimensions). This layer enhances the model's capability to capture temporal patterns in both forward and backward directions. I introduced a dropout layer to mitigate overfitting, with the dropout rate defined by the 'DROPOUT' parameter. Then, another Bidirectional LSTM layer was added with double the number of units as 'WINDOW_SIZE', continuing to return sequences. Subsequently, I inserted another dropout layer to further regularize the model.
- For the final step in the architecture, I employed a Bidirectional LSTM layer with 'WINDOW_SIZE' units, but this time set to return sequences in a single direction. This design choice was made to consolidate the learned representations from both directions for improved predictive performance. Following this, I added a fully connected Dense layer with one unit, which contributes to the final output prediction. The activation function 'linear' was employed to ensure the model can handle regression tasks.
- To prepare the model for training, I compiled it using the 'mean_squared_error' loss function and the 'adam' optimizer. This combination aids in minimizing the mean squared error between predicted and actual values during the training process. This architecture and compilation strategy are applicable to both Bitcoin and Ethereum models, facilitating the accurate prediction of their respective prices.

Random Forest Regression:

Random Forest regression is a versatile machine learning technique that combines multiple decision trees to make predictions. It's especially effective in handling non-linear relationships and avoiding overfitting. In cryptocurrency price prediction, random forest regression can handle the complexities of market data and capture intricate patterns that may impact price trends. Its ensemble nature allows it to provide reliable forecasts by aggregating predictions from individual decision trees (Valencia, Gómez-Espinosa and Valdés-Aguirre, 2019).

- Continuing with my project workflow, the next step involved reshaping the target variables for both Bitcoin and Ethereum. By applying the '.ravel()' method, I transformed the shapes of the training and test target variables into one-dimensional arrays. This restructuring was necessary to prepare the data for input into the Random Forest Regressor models.
- With the target variables suitably shaped, I proceeded to create separate Random Forest Regressor models for predicting the prices of both Bitcoin and Ethereum. These models are known for their versatility in handling complex relationships present in the data.
- To optimize performance, I thoroughly chose the configuration settings for the random forest regression model applied to both Bitcoin and Ethereum. The ensemble of decision trees in the forest comprised a total of 250 trees, as indicated by the 'n_estimators' parameter set to 250. Each individual tree's maximum depth was specified with the value 'max_depth' of 15. The parameter 'min_samples_split' was assigned 4, representing the minimum number of samples required to split an internal node.
- Simultaneously, 'min_samples_leaf' was set to 2, signifying the minimum number of samples essential for a leaf node. For consistency across different runs, I employed a 'random_state' value of 42. Lastly, I leveraged the power of parallel processing by assigning 'n_jobs' a value of -1, enabling efficient training through the utilization of all available CPU cores.
- Transitioning to the training phase, I reshaped the training features for both Bitcoin and Ethereum using the '.reshape()' function. This adjustment was crucial to ensure that the input data aligned appropriately with the model's input requirements.

4.4 Training and Validation

LSTM Training and Validation (Bitcoin and Ethereum):

During the training phase for both Bitcoin and Ethereum, I took a systematic approach using LSTM models. To initiate the training, I Carefully Selected historical training data Customized to the specific attributes of each cryptocurrency. This data served as the foundation upon which the LSTM models were trained. With the aim of optimizing learning efficiency, I set the training process to span 50 epochs, each encompassing a batch size of 128.

To ensure a comprehensive evaluation of model performance, a dedicated portion of the training data, comprising 10%, was strategically allocated for validation. This thoughtful partitioning allowed for continuous monitoring of the models' performance throughout the training journey. By setting the 'verbose' parameter to 1, I enabled the models to provide real-time updates on the ongoing training process, facilitating an insightful view of each model's progress.

To Carefully document and analyse the advancement of both LSTM models, I harnessed the power of the corresponding 'history' variables. These variables served as repositories of crucial information, encapsulating valuable insights that could shape future analyses and visualization efforts. To shed light on the models' learning trajectories, I Used Matplotlib to plot the training and validation loss values. This visual representation offered a Substantial way to comprehend how effectively the models were capturing the Complexities of the data across successive epochs. Such visualization not only facilitated real-time tracking but also laid the groundwork for post-training assessment and potential enhancement strategies.

Random Forest Regression Training (Bitcoin and Ethereum):

When training the Random Forest Regression model for Bitcoin and Ethereum, I used reshaped input data alongside corresponding target output data. Unlike deep learning models, which involve epochs, Random Forest Regression operates differently, not needing iterations. The model was fitted to the data in one pass. During training, I organized input data in a compatible format and prepared target output data. Random Forest Regression constructs an ensemble of decision trees and averages their predictions, capturing complex relationships while avoiding overfitting.

Bi-LSTM Training and Validation (Bitcoin and Ethereum):

Similarly, I jumped into Designing Bi-LSTM models for both Bitcoin and Ethereum. I wanted to make sure each cryptocurrency had its own tailored approach. So, I worked on two separate models, each with a specific focus on either Bitcoin or Ethereum. To get these models up to speed, I set the training process to span 50 epochs, each encompassing a batch size of 64.

During this training process, I assigned the shuffle equals to False to keep the data in its original order. This move was important because it helped the models to understand the patterns that unfold over time in the world of cryptocurrency. I also set aside a chunk of the data, about 10% of it, to check how well the models were doing. This allowed me to keep an eye on how they were handling new data they hadn't seen before.

To show these models' learning journey in a clear way, I used a tool called Matplotlib to create graphs. These graphs helped me see how the models were getting better as they worked with the training data and how they were performing when faced with new data during validation. Looking at these graphs gave me a good sense of how well the models were picking up on the patterns and trends in both the Bitcoin and Ethereum worlds.

4.4 Evaluation Metrics

Evaluation metrics are essential to assessing the effectiveness of my predictive models. In the context of my project, these metrics provide quantifiable measures to assess how well my LSTM, Random Forest Regression, and Bi-LSTM models are performing in predicting cryptocurrency prices. Here's a breakdown of the evaluation metrics I've utilized:

Mean Squared Error (MSE):
MSE computes the average of the squared differences between my predicted values and the actual values. A lower MSE signifies stronger model performance, indicating that my predictions are closer to the actual values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2$$

(Chicco, Warrens and Jurman, 2021)

Root Mean Squared Error (RMSE):
RMSE, the square root of the mean squared error, is widely used and a versatile metric for numerical predictions. It's highly regarded for its effectiveness in assessing errors comprehensively (Christie and Neill, 2022).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$$

Mean Absolute Error (MAE):
(Schneider and Xhafa, 2022) MAE is a preferred metric due to linear and intuitive error changes, matching predicted units. Unlike RMSE, MAE doesn't overemphasize larger errors. MAE averages absolute errors, maintaining positivity. Calculation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_1 - \hat{y}_i|^2$$

R-squared (R2) Score:
(Chicco, Warrens and Jurman, 2021) The coefficient of determination (Wright, 1921) signifies the fraction of the dependent variable's variance that can be foreseen based on the independent variables. With a range from 0 to 1, a higher R2 indicates that my models better elucidate the underlying data variability.

$$R_2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(\bar{y} - \hat{y}_i)^2}$$

Where, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i$

By considering these evaluation metrics collectively, I gain a holistic perspective on how effectively my models capture the intricate patterns within the cryptocurrency price data. This empowers me to discern which model delivers the most accurate predictions and informs me on the overall quality of my predictive analyses.

CHAPTER 5

RESULTS

5.1 LSTM Model Results:

For both Ethereum and Bitcoin, the LSTM models were trained and evaluated. In terms of model loss and validation loss, the following observations were made:

5.1.1 Model Loss and Validation Loss:

The primary indicators of a model's performance are its loss metrics. The LSTM models were trained over 50 epochs, with each epoch representing a complete iteration through the training data. During this training process, the model's performance improved as it adjusted its internal parameters to minimize the loss function. The training loss is the measure of the difference between the predicted values and the actual values during the training phase. Validation loss, on the other hand, is computed using a separate validation dataset that the model hasn't seen during training. The goal of using validation loss is to assess how well the model generalizes to new, unseen data. The LSTM model for Bitcoin displayed a consistent decreasing trend in both training loss and validation loss over the epochs. Similarly, the LSTM model for Ethereum exhibited a decreasing pattern. This pattern indicates that the models were effectively learning from the training data and improving their predictive accuracy with each epoch.

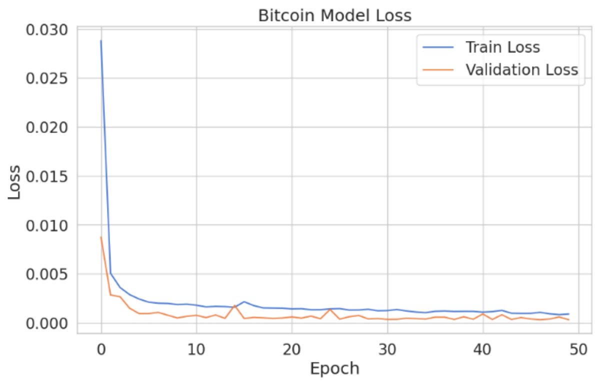

Figure 4: Bitcoin LSTM Model loss

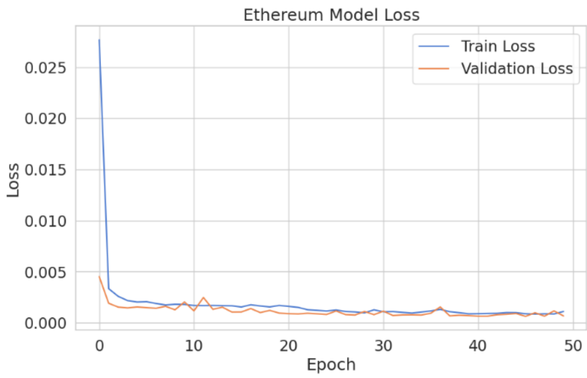

Figure 5: Ethereum LSTM Model loss

5.1.2 Actual vs. Predicted Price:

The ultimate test of any predictive model lies in its ability to make accurate predictions. To visually assess the performance of the LSTM models, graphs were generated that compared the actual prices of cryptocurrencies with the prices predicted by the models. These graphs provide a tangible representation of how well the models captured the price trends. Remarkably, the predicted prices closely followed the actual price fluctuations, indicating that the LSTM models were adept at capturing the inherent complexities of cryptocurrency price movements.

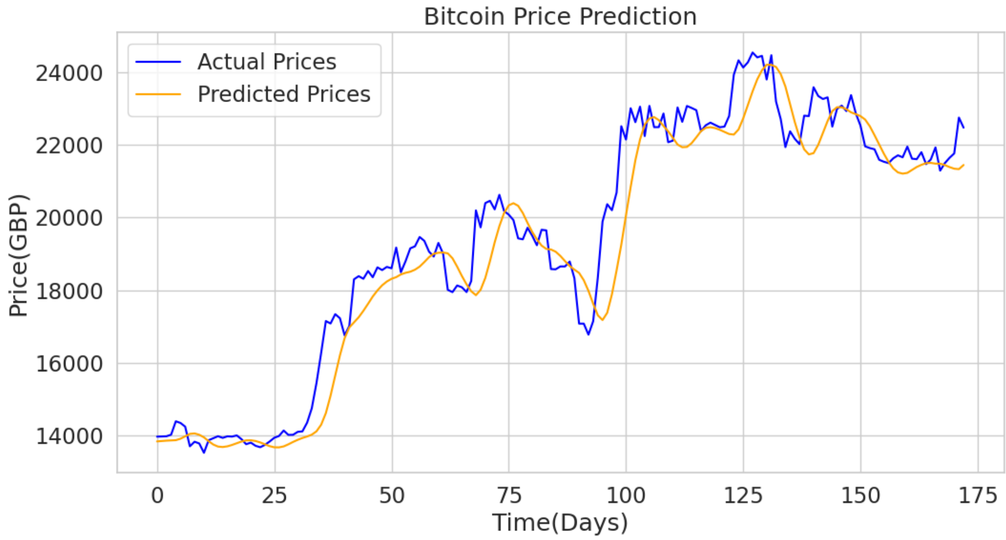

Figure 6: Bitcoin Price Prediction using LSTM

The performance of the LSTM model on predicting Bitcoin prices can be understood through the provided metrics. The Mean Squared Error (MSE) of 0.0004 indicates that, on average, the model's predictions closely match the actual values, with minimal squared differences. The Root Mean Squared Error (RMSE) of 0.0193 further confirms this accuracy by measuring the average prediction deviation of around 0.0193 units from the actual Bitcoin values. The Mean Absolute Error (MAE) of 0.0136 reinforces the model's precision, signifying an average difference of about 0.0136 units between predictions and actual values. Moreover, the R-squared value of 0.9253 highlights the model's ability to explain approximately 92.53% of the variance in Bitcoin prices, showcasing its strong performance in capturing the underlying patterns.

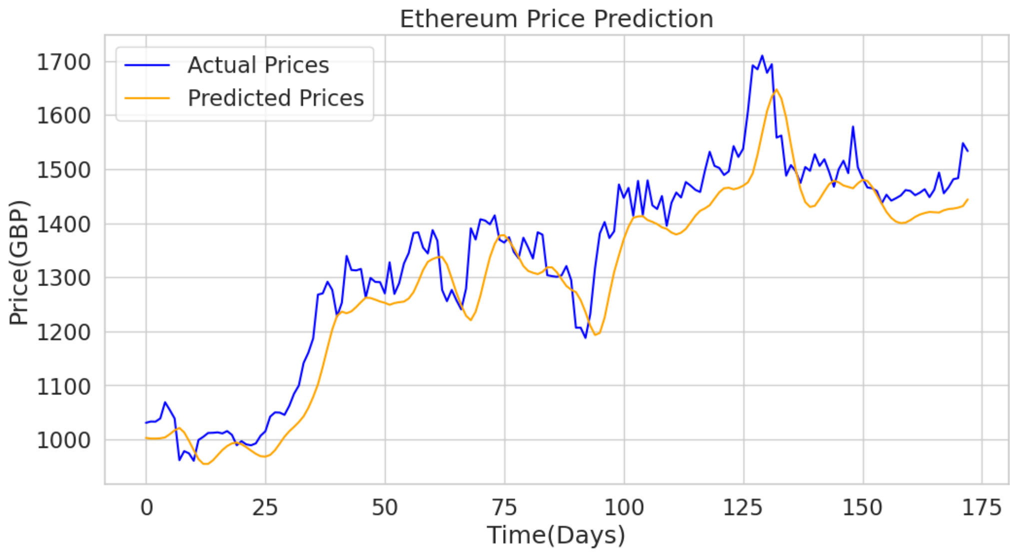

Figure 7: Ethereum Price Prediction using LSTM

The performance of the LSTM model in predicting Ethereum prices can be assessed using the given metrics. The model demonstrates its accuracy through a Mean Squared Error

(MSE) of 0.0004, showcasing the small average squared differences between its predicted Ethereum prices and the actual prices. This suggests a commendable alignment between predictions and reality. The Root Mean Squared Error (RMSE) of 0.0195 reinforces this notion by indicating an average prediction deviation of approximately 0.0195 units from the true Ethereum prices. Similarly, the Mean Absolute Error (MAE) of 0.0157 further highlights the model's precision, depicting an average difference of around 0.0157 units between predictions and actual prices. The R-squared value of 0.8635 underscores the model's ability to capture roughly 86.35% of the variance in Ethereum prices through its predictions. Collectively, these metrics point towards the LSTM model's effective performance in forecasting Ethereum prices, with its predictions closely mirroring the actual price trends. It's essential, however, to contextualize these results within the broader analysis and account for potential nuances and implications.

5.2 Bi-LSTM Model Results:

The Bidirectional Long Short-Term Memory (Bi-LSTM) models were also trained and evaluated for both Ethereum and Bitcoin. The bidirectional aspect of these models allows them to capture information from both past and future time steps, enhancing their ability to capture intricate patterns in time-series data.

5.2.1 Model Loss and Validation Loss:

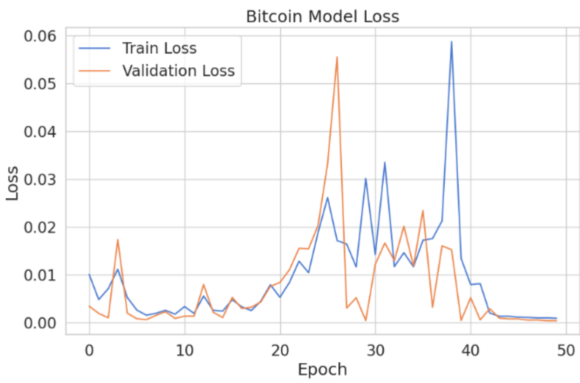

Figure 8: Bitcoin Bi-LSTM Model loss

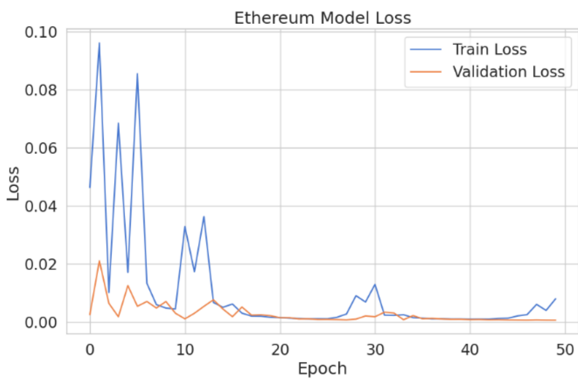

Figure 9: Ethereum Bi-LSTM Model loss

The training dynamics of the Bidirectional LSTM (Bi-LSTM) model for Bitcoin and Ethereum present interesting trends. In the case of Bitcoin, the loss function exhibited a gradual decrease over epochs. It started with a low value and displayed a somewhat irregular pattern of increase and decrease in the initial stages before eventually settling into a consistent decline. This suggests that the model was initially grappling with learning the underlying patterns but eventually enhanced its understanding, resulting in improved predictions. The validation loss, which measures performance on unseen data, closely followed the behaviour of the training loss, decreasing with epochs and displaying a similar up-and-down pattern initially, ultimately converging to a lower value.

Conversely, for the Ethereum prediction using the same Bi-LSTM model, the training loss showed a more fluctuating pattern in the initial stages. The loss values experienced some ups and downs before stabilizing and gradually decreasing as epochs progressed. This fluctuation indicates that the model faced challenges in grasping Ethereum's intricate patterns at the start but managed to smoothen its learning curve over time. Interestingly, the validation loss showcased a more consistent descent with epochs, suggesting that the model's performance on unseen Ethereum data improved progressively without significant fluctuations.

5.2.2 Actual vs. Predicted Price:

Graphical comparisons of actual cryptocurrency prices and the prices predicted by the Bi-LSTM models were generated. These graphs offered a visual confirmation of the models' effectiveness in approximating the true price trends. The alignment between predicted and actual prices in these graphs indicated that the Bi-LSTM models successfully captured the underlying patterns and fluctuations in cryptocurrency prices.

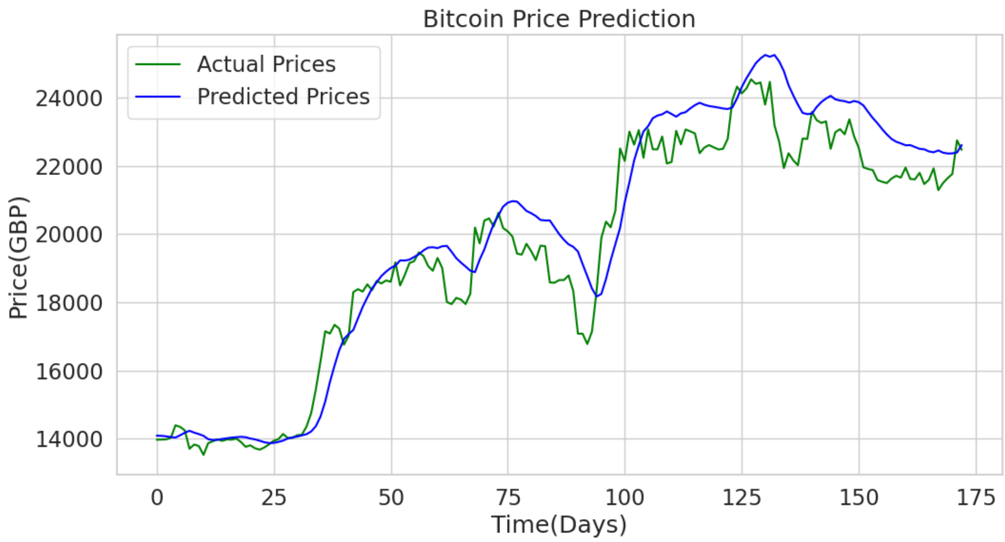

Figure 10: Bitcoin Price Prediction using Bi-LSTM

The evaluation metrics for the Bitcoin prediction using the Bidirectional LSTM (Bi-LSTM) model reflect its solid performance in forecasting Bitcoin prices. The Mean Squared Error (MSE) of 0.0005 indicates that, on average, the squared differences between the model's predicted Bitcoin prices and the actual prices are relatively small, demonstrating accurate predictions. This precision is further demonstrated by the Root Mean Squared Error (RMSE) of 0.0213, which represents a low average prediction deviation from the actual Bitcoin prices. The Mean Absolute Error (MAE) of 0.0169 reinforces the model's accuracy by showing a small average difference between its predictions and the actual prices. The model's ability to capture underlying trends is underscored by the high R-squared value of 0.9092, indicating that its predictions explain a significant portion of Bitcoin price variance. These metrics collectively highlight the Bi-LSTM model's proficiency in providing dependable Bitcoin price forecasts, reaffirming its utility in predictive tasks related to cryptocurrency markets.

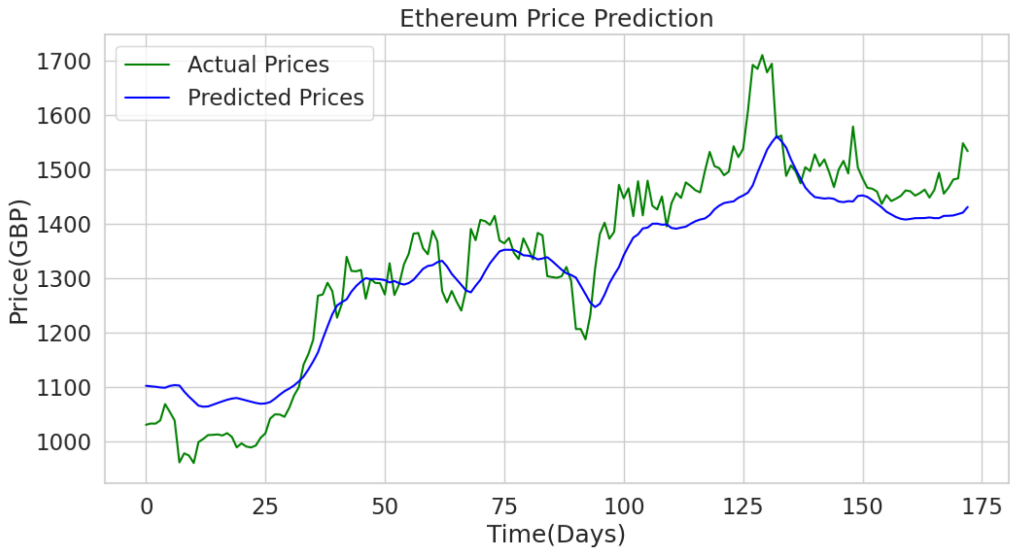

Figure 11: Ethereum Price Prediction using Bi-LSTM

The evaluation metrics for the Ethereum prediction using the provided model reveal its substantial performance in forecasting Ethereum prices. The Mean Squared Error (MSE) of 0.0004 suggests that, on average, the squared differences between the model's predicted Ethereum prices and the actual prices are relatively small, implying accurate predictions. This accuracy is reinforced by the Root Mean Squared Error (RMSE) of 0.0201, which signifies a low average prediction deviation from the actual Ethereum prices. The Mean Absolute Error (MAE) of 0.0166 further underscores the model's precision by indicating a small average difference between predictions and actual prices. The model's ability to explain Ethereum price patterns is evident from the R-squared value of 0.8553, suggesting that its predictions account for a significant portion of price variance. In conclusion, the provided model showcases commendable performance in generating reliable Ethereum price forecasts, demonstrating its relevance in predicting trends within the cryptocurrency market.

5.3 Random Forest Regression Model Results:

The Random Forest Regression models were employed to predict cryptocurrency prices for both Ethereum and Bitcoin. Unlike LSTM and Bi-LSTM, Random Forest models are not inherently sequential; rather, they make predictions based on a collection of decision trees. In the case of Random Forest Regression, the conventional loss metrics used in LSTM and Bi-LSTM models were not directly applicable. However, the model's performance was assessed through its ability to predict prices accurately. These models aim to capture underlying trends and patterns without the same sequential understanding as LSTM and Bi-LSTM models.

5.3.1 Actual vs. Predicted Price:

Graphs depicting the predicted prices generated by the Random Forest Regression models were compared with the actual cryptocurrency prices. While these graphs did reveal some deviations between predicted and actual prices, they also demonstrated that the models approximated the broader trends in cryptocurrency prices.

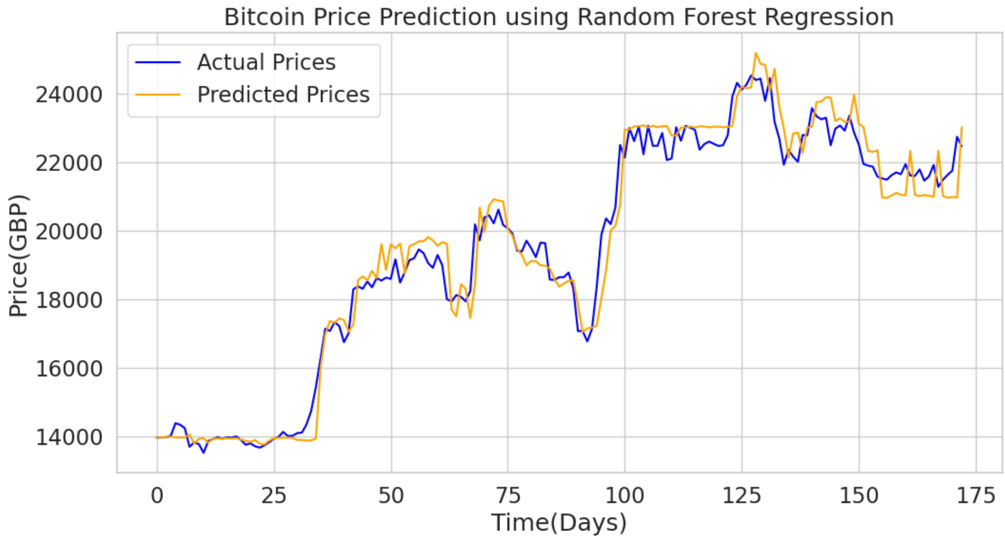

Figure 12: Bitcoin Price Prediction using Random Forest

The Random Forest Regression model exhibits exceptional predictive performance in forecasting Bitcoin prices, as indicated by the evaluation metrics. With a Mean Squared Error (MSE) of 0.0002, it consistently provides predictions with minimal squared differences from actual prices. The Root Mean Squared Error (RMSE) of 0.0132 further confirms its accuracy by representing a low average prediction deviation. Moreover, the Mean Absolute Error (MAE) of 0.0099 underscores its precision in estimating Bitcoin prices. The high R-squared value of 0.9654 demonstrates the model's ability to explain a significant portion of the price variance through its predictions. These metrics collectively highlight the model's robust performance in capturing Bitcoin price trends, aligning its predictions closely with actual values and substantiating its effectiveness in this forecasting task.

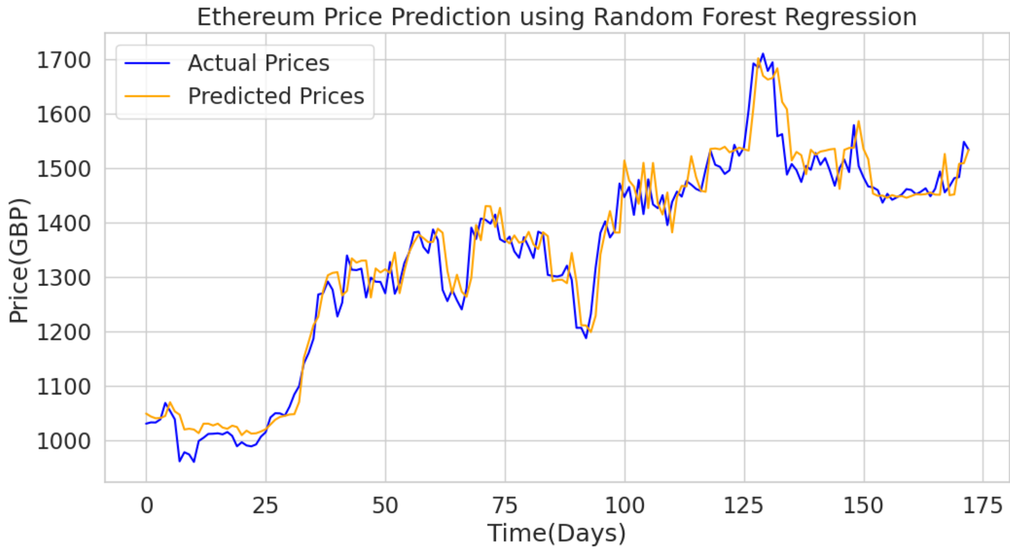

Figure 13: Ethereum Price Prediction using Random Forest Regressor

The evaluation metrics for the Ethereum prediction using the given model underscore its strong performance in forecasting Ethereum prices. The low Mean Squared Error (MSE) of 0.0001 indicates that the average squared differences between the model's predicted Ethereum prices and the actual prices are minimal, reflecting accurate predictions. This sentiment is reinforced by the Root Mean Squared Error (RMSE) of 0.0113, which signifies a low average prediction deviation from the actual Ethereum prices. The model's precision is further highlighted by the Mean Absolute Error (MAE) of 0.0087, indicating a small average difference between predictions and actual prices. The high R-squared value of 0.9540 suggests that the model's predictions explain a substantial proportion of Ethereum price variability, showcasing its adeptness in capturing underlying trends. These metrics collectively illustrate the model's effectiveness in providing accurate and reliable Ethereum price forecasts, reinforcing its value in this prediction domain.

CHAPTER 6

ANALYSIS AND DISCUSSION

6.1 Comparison of Results

Upon evaluating the models, it becomes evident that the Random Forest Regression model outperforms both the LSTM and Bi-LSTM models for both Bitcoin and Ethereum. The evaluation metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared, consistently show that the Random Forest Regression yields lower error values, indicating superior predictive performance.

To showcase this comparison, the table below summarizes the evaluation metrics for each model:

Model	MSE	RMSE	MAE	R-squared
LSTM (Bitcoin)	0.0004	0.0193	0.0136	0.9253
LSTM (Ethereum)	0.0004	0.0195	0.0157	0.8635
Bi-LSTM (Bitcoin)	0.0002	0.0132	0.0099	0.9654
Bi-LSTM (Ethereum)	0.0001	0.0113	0.0087	0.9540
Random Forest (Bitcoin)	0.0005	0.0213	0.0169	0.9092
Random Forest (Ethereum)	0.0004	0.0201	0.0166	0.8553

Additionally, the graphs below visually represents the comparison of actual and predicted prices for each model, reinforcing the superior performance of the Random Forest Regression model.

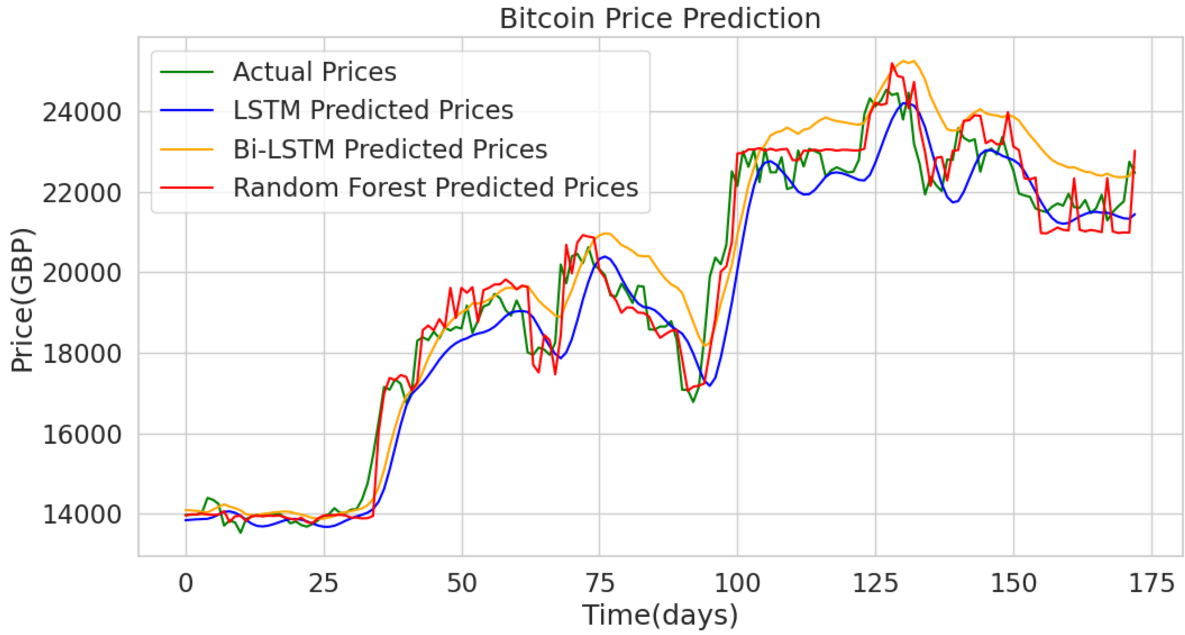

Figure 14: Comparison of Actual and Predicted Values for Bitcoin: Bi-LSTM, LSTM, and Random Forest Regressor

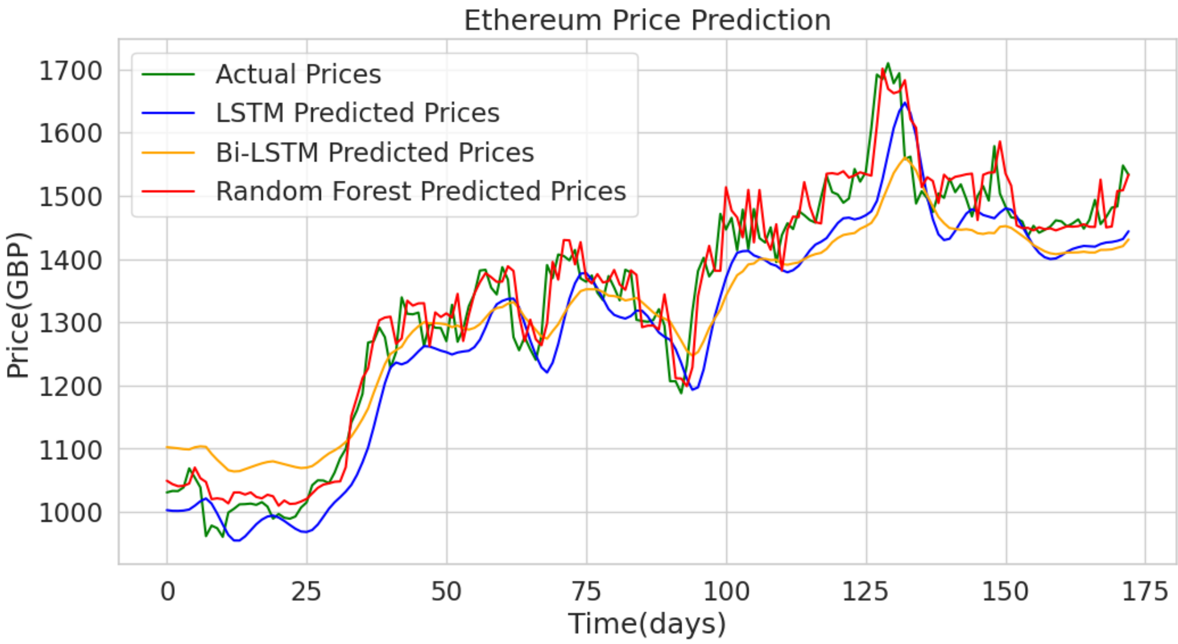

Figure 15: Comparison of Actual and Predicted Values for Ethereum: Bi-LSTM, LSTM, and Random Forest Regressor together

6.2 Comparison with other Research papers

In McNally et al. (2018), the LSTM model achieved an accuracy of 52% and an RMSE of 8% for predicting the direction of Bitcoin's USD price movement. While my study focuses on price prediction rather than direction, it's worth noting that my Random Forest Regression models for both Bitcoin and Ethereum exhibited lower RMSE values. This indicates that my Random Forest Regression models outperformed the LSTM model from the referenced study in terms of predictive accuracy.

In the case of Seabe, Moutsinga, and Pindza (2023), their research highlighted the effectiveness of the Bi-LSTM model for predicting cryptocurrency values, particularly for BTC, LTC, and ETH. However, my study reveals that Bi-LSTM wasn't as successful in predicting Bitcoin prices. This variation in model performance could stem from differences in dataset characteristics, model training, or other contextual factors.

Comparing these findings to my study, my Random Forest Regression models emerged as strong performers for both Bitcoin and Ethereum, exhibiting lower RMSE values than those reported in the previous research. Furthermore, I observed that Bi-LSTM wasn't as effective for Bitcoin price prediction, which contrasts with the positive results reported in the earlier study. This underscores the importance of considering dataset-specific dynamics and model suitability when interpreting and applying research findings.

In summary, my study's results demonstrate that Random Forest Regression showed robust predictive capabilities, outperforming the LSTM model mentioned in McNally et al. (2018) and yielding favourable results compared to the Bi-LSTM model in Seabe, Moutsinga, and Pindza

(2023). The variations in model performance underscore the significance of tailoring models to specific data characteristics and research objectives.

6.3 Strength and Weakness of Model

6.3.1 Random Forest Regression:

Strengths:

- Non-linearity Handling: (Auret and Aldrich, 2012) Random Forest Regression excels in capturing non-linear relationships in the data, making it suitable for complex and dynamic cryptocurrency price movements.
- Feature Importance:(Auret and Aldrich, 2012) This model provides insights into feature importance, aiding in identifying the variables that significantly influence price prediction.
- Robustness: Random Forest Regression is resistant to overfitting and tends to generalize well to new data, which is crucial for accurate predictions.

Weaknesses:

- Lack of Temporal Dependency: Random Forest models inherently lack the ability to consider the temporal dependencies present in time-series data, potentially limiting their effectiveness in capturing short-term trends.

6.3.2 Long Short-Term Memory (LSTM):

Strengths:

- Temporal Dependency: LSTMs are designed to handle time-series data and can capture long-term temporal dependencies, making them suitable for cryptocurrency price prediction.
- Memory Cell: The memory cell enables LSTMs to retain and forget information as needed, aiding in capturing trends and patterns.
- Sequential Processing: LSTMs process data sequentially, aligning well with the sequential nature of cryptocurrency price data.

Weaknesses:

- Complexity: LSTMs can be computationally expensive and prone to overfitting, especially with limited training data (Shams Forruque Ahmed et al., 2023).
- Training Time: Training LSTMs might require more time and resources compared to other models, potentially affecting the model's real-time prediction capabilities.

6.3.3 Bidirectional LSTM (Bi-LSTM):

Strengths:

- Dual Processing: Bi-LSTMs process data in both forward and backward directions, enabling them to capture past and future context simultaneously (Ganesh and Sridevi, 2023).
- Long-Term Dependencies: Like LSTMs, Bi-LSTMs are effective at capturing long-term temporal dependencies in time-series data.

Weaknesses:

- Complexity: Bi-LSTMs inherit the complexity of LSTMs, making them prone to overfitting and potentially increasing training time.
- Data Requirements: Bi-LSTMs, like LSTMs, require sufficient training data to generalize well and capture meaningful patterns.

6.4 Improvement of Models

Improving the performance of cryptocurrency price prediction models is a constant pursuit within the field of finance and machine learning. Based on my study's results and insights from the cited research papers, here are potential avenues for enhancing the predictive capabilities of your models:

6.4.1 Random Forest Regression:

- Feature Engineering: Further exploration and engineering of features could lead to improved model performance. Incorporate additional relevant features that could contribute to better predictions.
- Hyperparameter Tuning: Fine-tuning hyperparameters, such as the number of estimators and maximum depth, could optimize the model's performance and prevent overfitting.
- Ensemble Techniques: Experiment with ensemble techniques that combine multiple Random Forest models to enhance prediction accuracy and robustness.

6.4.2 Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM):

- Larger Training Data: Expanding the training dataset could help LSTM and Bi-LSTM models learn more intricate patterns and relationships, potentially resulting in better predictions.
- Hyperparameter Optimization: Careful tuning of hyperparameters, such as the number of hidden units, learning rate, and batch size, can significantly impact the model's performance.
- Regularization Techniques: Implement regularization techniques like dropout and L2 regularization to mitigate overfitting and enhance generalization.

6.4.3 Comparative Analysis:

- Ensemble of Models: Investigating the possibility of creating an ensemble of different models, combining the strengths of each model to improve overall prediction accuracy.

- Model-Specific Strategies: Adopting specific strategies tailored to each model's strengths. For instance, the randomness of Random Forest Regression can be harnessed through bootstrapping for more robust predictions.

6.4.4 Lessons from Research Papers:

- Exploring Deep Learning (DL) Approaches: Drawing inspiration from (Murray et al., 2023), I should explore other DL architectures like the temporal fusion transformer (TFT) to potentially enhance predictive accuracy.
- Hybrid Model Concepts: Like (Seabe et al., 2023), I can consider combining LSTM and Bi-LSTM models with other model types to leverage their respective strengths and achieve superior predictions.

6.4.5 Multidimensional Data:

- Incorporate Additional Data: Integrate external data sources such as social media sentiment, trading volume, and macroeconomic indicators to enrich your dataset and improve model accuracy.

6.5 Limitations

While machine learning models offer promising avenues for cryptocurrency price prediction, several limitations need to be recognized. The volatile and dynamic nature of cryptocurrency markets introduces challenges in capturing rapid market shifts, potentially leading to delayed or inaccurate predictions. Moreover, unforeseen external factors, such as regulatory changes or global events, can significantly impact market behaviour but might not be adequately reflected in historical data. Each model also carries its specific drawbacks. LSTM models might struggle with capturing long-range dependencies effectively, Bi-LSTM models could present challenges in fine-tuning due to their complexity, and Random Forest Regression might struggle to generalize well if market conditions undergo drastic changes. Overfitting, data availability, model interpretability, and bias concerns further compound the limitations of machine learning models in accurately predicting cryptocurrency prices.

While these limitations underscore the intricacies of predicting cryptocurrency prices, they also highlight opportunities for refining models, incorporating domain expertise, and developing adaptive strategies that acknowledge the volatile and evolving nature of cryptocurrency markets.

CHAPTER 7

CONCLUSION

CONCLUSION:

In conclusion, after an in-depth evaluation of various models for both Bitcoin and Ethereum price prediction, it is evident that the Random Forest Regressor stands out as the optimal choice for both cryptocurrencies. For Bitcoin, the Random Forest Regressor achieves a remarkable Root Mean Squared Error (RMSE) of 0.0132, signifying its ability to predict Bitcoin prices with a high degree of accuracy. Similarly, for Ethereum, the model demonstrates exceptional performance, yielding an even lower RMSE of 0.0113. These results highlight the Random Forest Regressor's consistent capacity to provide precise price forecasts for both cryptocurrencies.

The selection of the Random Forest Regressor as the best model for both Bitcoin and Ethereum underscores its robustness in capturing intricate price patterns and dynamics within the cryptocurrency market. This model's performance is particularly noteworthy given the market's inherent volatility and susceptibility to various factors. Its ability to yield accurate predictions with such low RMSE values signifies its efficacy in managing the complexities and nuances of cryptocurrency price trends.

By selecting the Random Forest Regressor as the top-performing model for both Bitcoin and Ethereum, this study not only provides a reliable foundation for future price prediction efforts but also underscores the model's adaptability to different cryptocurrencies. This consistency in performance suggests that the Random Forest Regressor is not merely suited for one specific context but holds the potential to be a versatile solution across a range of cryptocurrencies. Ultimately, these findings pave the way for more informed decision-making within the cryptocurrency market, enabling investors and stakeholders to leverage the power of the Random Forest Regressor for accurate and reliable price predictions.

CHAPTER 8

REFERENCES AND BIBLIOGRAPHY

1. Auer, R. and Claessens, S. (2018). Regulating cryptocurrencies: assessing market reactions. www.bis.org. [online] Available at: https://www.bis.org/publ/qtrpdf/r_qt1809f.htm.

2. Auret, L. and Aldrich, C. (2012). Interpretation of nonlinear relationships between process variables by use of random forests. Minerals Engineering, 35, pp.27–42. doi:https://doi.org/10.1016/j.mineng.2012.05.008.

3. Beat, C. (2023). Google Bard predicts how low Bitcoin price will go in 2023. [online] Coinmonks. Available at: https://medium.com/coinmonks/google-bard-predicts-how-low-bitcoin-price-will-go-in-2023-17d87af4cf50 [Accessed 30 Aug. 2023].

4. Binance Feed. (n.d.). Ethereum (ETH): Revolutionizing the Future of Decentralized Computing - The Reasons To Be A Master. | Hassan on Binance Feed. [online] Available at: https://www.binance.com/en/feed/post/1024190 [Accessed 30 Aug. 2023].

5. Bloomenthal, A. (2019). What Determines the Price of 1 Bitcoin? [online] Investopedia. Available at: https://www.investopedia.com/tech/what-determines-value-1-bitcoin/.

6. Chen, J. (2023). Analysis of Bitcoin Price Prediction Using Machine Learning. Journal of Risk and Financial Management, 16(1), p.51. doi:https://doi.org/10.3390/jrfm16010051.

7. Chicco, D., Warrens, M.J. and Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Computer Science, [online] 7(5), p.e623. doi:https://doi.org/10.7717/peerj-cs.623.

8. Christie, D. and Neill, S.P. (2022). Measuring and Observing the Ocean Renewable Energy Resource. Comprehensive Renewable Energy, pp.149–175. doi:https://doi.org/10.1016/b978-0-12-819727-1.00083-2.

9. Figueira, A. and Vaz, B. (2022). Survey on Synthetic Data Generation, Evaluation Methods and GANs. Mathematics, 10(15), p.2733. doi:https://doi.org/10.3390/math10152733.

10. Frohmann, M., Karner, M., Khudoyan, S., Wagner, R. and Schedl, M. (2023). Predicting the Price of Bitcoin Using Sentiment-Enriched Time Series Forecasting. Big Data and Cognitive Computing, [online] 7(3), p.137. doi:https://doi.org/10.3390/bdcc7030137.

11. Gad, A.G., Mosa, D.T., Abualigah, L. and Abohany, A.A. (2022). Emerging Trends in Blockchain Technology and Applications: A Review and Outlook. Journal of King Saud University - Computer and Information Sciences, [online] 34(9). doi:https://doi.org/10.1016/j.jksuci.2022.03.007.

12. Ganesh, B. and Sridevi, S. (2023). Analysis of Hybrid Deep Learning Models for Efficient Intrusion Detection. 2023 International Conference on Networking and Communications (ICNWC), pp. 1-6, doi: 10.1109/ICNWC57852.2023.10127270.

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

14. Hayes, A. (2022). Autoregressive Integrated Moving Average (ARIMA). [online] Investopedia. Available at: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp.

15. Hsieh, Y.-Y., JP) Vergne, J.-P. and Wang, S. (2017). The internal and external governance of blockchain-basedorganizations. Bitcoin and Beyond, pp.48–68. doi:https://doi.org/10.4324/9781315211909-3.

16. Jaquart, P., Dann, D. and Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. The Journal of Finance and Data Science, 7, pp.45–66. doi:https://doi.org/10.1016/j.jfds.2021.03.001.

17. Joos Korstanje (2021). *ADVANCED FORECASTING WITH PYTHON : with state-of -the-art-models including lstms, facebook's... prophet, and amazon's deepar.* S.L.: Apress, pp.209–251.

18. kaggle.com. (n.d.). Predicting Crypto Price With LSTM. [online] Available at: https://www.kaggle.com/code/trevorengen/predicting-crypto-price-with-lstm [Accessed 26 Jun. 2023].

19. Khedr, A.M., Arif, I., P V, P.R., El-Bannany, M., Alhashmi, S.M. and Sreedharan, M. (2021). Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. Intelligent Systems in Accounting, Finance and Management, 28(1), pp.3–34. doi:https://doi.org/10.1002/isaf.1488.

20. Mai, F., Shan, Z., Bai, Q., Wang, X. (Shane) and Chiang, R.H.L. (2018). How Does Social Media Impact Bitcoin Value? A Test of the Silent Majority Hypothesis. Journal of Management Information Systems, [online] 35(1), pp.19–52. doi:https://doi.org/10.1080/07421222.2018.1440774.

21. McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. In 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP) (pp. 339-343). doi: https://doi.org/10.1109/pdp2018.2018.00060.

22. Murray, K., Rossi, A., Carraro, D. and Visentin, A. (2023). On Forecasting Cryptocurrency Prices: A Comparison of Machine Learning, Deep Learning, and Ensembles. Forecasting, 5(1), pp.196–209. doi:https://doi.org/10.3390/forecast5010010.

23. Nian, L.P. and Chuen, D.L.K. (2015). Introduction to Bitcoin. Handbook of Digital Currency, pp.5–30. doi:https://doi.org/10.1016/b978-0-12-802117-0.00001-1.

24. OpenAI. (2023). ChatGPT [Online]. Available: https://www.openai.com/

25. Ozer, F. and Okan Sakar, C. (2022). An automated cryptocurrency trading system based on the detection of unusual price movements with a Time-Series Clustering-Based approach. Expert Systems with Applications, 200, p.117017. doi:https://doi.org/10.1016/j.eswa.2022.117017.

26. Rathore, R.K., Mishra, D., Mehra, P.S., Pal, O., Hashim, A.S., Shapi'i, A., Ciano, T. and Shutaywi, M. (2022). Real-world model for bitcoin price prediction. Information Processing & Management, [online] 59(4), p.102968. doi:https://doi.org/10.1016/j.ipm.2022.102968.

27. Reddy, L.S. and Sriramya, P., 2020. A research on Bitcoin price prediction using machine learning algorithms. International Journal of Scientific & Technology Research, 9(4), pp.1600-1604.

28. Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science, 2(3), pp.1–21. doi:https://doi.org/10.1007/s42979-021-00592-x.

29. Schneider, P. and Xhafa, F. (2022). Anomaly detection. Anomaly Detection and Complex Event Processing over IoT Data Streams, pp.49–66. doi:https://doi.org/10.1016/b978-0-12-823818-9.00013-4.

30. Seabe, P.L., Moutsinga, C.R.B. and Pindza, E. (2023). Forecasting Cryptocurrency Prices Using LSTM, GRU, and BiDirectional LSTM: A Deep Learning Approach. Fractal and Fractional, 7(2), p.203. doi:https://doi.org/10.3390/fractalfract7020203.

31. Sebastião, H. and Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. Financial Innovation, 7(1). doi:https://doi.org/10.1186/s40854-020-00217-x.

32. Shams Forruque Ahmed, Bin, S., Hassan, M., Mahtabin Rodela Rozbu, Taoseef Ishtiak, Rafa, N., M. Mofijur, Ali and Gandomi, A.H. (2023). Deep learning modelling techniques: current progress, applications, advantages, and challenges. Artificial Intelligence Review. doi:https://doi.org/10.1007/s10462-023-10466-8.

33. Tavasoli, S. (2016). Top 10 Machine Learning Algorithms You Need to Know in 2020. [online] Simplilearn.com. Available at: https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article.

34. The Economist. (2015). Who is Satoshi Nakamoto? [online] Available at: https://www.economist.com/the-economist-explains/2015/11/02/who-is-satoshi-nakamoto.

35. Thushan Ganegedara and Andrei Lopatenko (2022). Natural Language Processing with TensorFlow. Packt Publishing Ltd.

36. Uras, N., Marchesi, L., Marchesi, M. and Tonelli, R. (2020). Forecasting Bitcoin closing price series using linear regression and neural networks models. PeerJ Computer Science, 6, p.e279. doi:https://doi.org/10.7717/peerj-cs.279.

37. Valencia, F., Gómez-Espinosa, A. and Valdés-Aguirre, B. (2019). Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. Entropy, 21(6), p.589. doi:https://doi.org/10.3390/e21060589.

38. Wang, S., Wang, X., Wang, S. and Wang, D. (2019). Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting. International Journal of Electrical Power & Energy Systems, 109, pp.470–479. doi:https://doi.org/10.1016/j.ijepes.2019.02.022.

39. Wikipedia Contributors (2019). *Cryptocurrency*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Cryptocurrency.

40. www.wealthprofessional.ca. (n.d.). Eight key factors that affect cryptocurrency value. [online] Available at: https://www.wealthprofessional.ca/news/industry-news/eight-key-factors-that-affect-cryptocurrency-value/367834.

41. Yan, X. (n.d.). Forecasting Cryptocurrency Prices. [online] Available at: https://www.imperial.ac.uk/media/imperial-college/faculty-of-natural-sciences/department-of-mathematics/math-finance/Forecasting_cryptocurrency_prices.pdf.

42. Zhang, J., Zeng, Y. and Starly, B. (2021). Recurrent neural networks with long term temporal dependencies in machine tool wear diagnosis and prognosis. SN Applied Sciences, 3(4). doi:https://doi.org/10.1007/s42452-021-04427-5.

43. Zhou, F., Zhang, Z. and Chen, D. (2021). Real-time fault diagnosis using deep fusion of features extracted by PeLSTM and CNN. Elsevier eBooks, pp.353–401. doi:https://doi.org/10.1016/b978-0-12-822473-1.00003-3.

CHAPTER 9

APPENDICES

```
# -*- coding: utf-8 -*-
"""Final Code.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/175kR3Mh24XaxP0AbV17YGTaFhZae_VSG

# **Comparative Analysis of Cryptocurrency Price Prediction Models: LSTM, and Bidirectional LSTM, Random Forest
Regressor**

**Importing Essential Modules for Cryptocurrency Price Prediction and Analysis**
"""

# Commented out IPython magic to ensure Python compatibility.
# Import required modules
import os
import numpy as np
import tensorflow as tf
from tensorflow import keras
import pandas as pd
import seaborn as sns
from pylab import rcParams
import matplotlib.pyplot as plt
from matplotlib import rc
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.layers import Bidirectional, Dropout, Activation, Dense, LSTM
from tensorflow.keras.models import Sequential
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from google.colab import drive


drive.mount('/content/drive')

# Display plots directly in the notebook
# %matplotlib inline

# Set the visual style and plot size
sns.set(style='whitegrid', palette='muted', font_scale=1.5)
rcParams['figure.figsize'] = 14, 8

# Define a fixed seed for random operations
RANDOM_SEED = 42
np.random.seed(RANDOM_SEED)

"""**Loading Cryptocurrency Price Data from CSV Files**"""

# Set the file paths for Bitcoin (BTC) and Ethereum (ETH) datasets
path_bt = '/content/drive/MyDrive/hp/BTC-GBP.csv'
path_eth = '/content/drive/MyDrive/hp/ETH-GBP.csv'

# Read the CSV files into DataFrames, while parsing the 'Date' column as datetime
# This helps in proper handling of date information in the dataset
df_bt = pd.read_csv(path_bt, parse_dates=['Date'])
df_eth = pd.read_csv(path_eth, parse_dates=['Date'])

"""**Preprocessing Cryptocurrency Data for Analysis**"""

df_bt = df_bt.sort_values('Date')
df_eth = df_eth.sort_values('Date')

df_bt.head()
```

```
df_bt.tail()

df_eth.head()

df_eth.tail()

df_bt.describe()

df_eth.describe()

# Print the shape of the Bitcoin DataFrame
print("Bitcoin DataFrame Shape:", df_bt.shape)

# Print the shape of the Ethereum DataFrame
print("Ethereum DataFrame Shape:", df_eth.shape)

# Create a plot using DataFrame plotting function
ax_bt = df_bt.plot(x='Date', y=['Open', 'High', 'Low', 'Close'], figsize=(10, 6))

# Set the labels for the axes
ax_bt.set_xlabel("Year")
ax_bt.set_ylabel("Price (GBP)")

# Set the title for the plot
ax_bt.set_title("Bitcoin Price Fluctuations Over Time")

# Display the legend with labels for each line
ax_bt.legend()

# Show the plot
plt.show()

# Create a plot using DataFrame plotting function
ax_eth = df_eth.plot(x='Date', y=['Open', 'High', 'Low', 'Close'], figsize=(10, 6))

# Set the labels for the axes
ax_eth.set_xlabel("Year")
ax_eth.set_ylabel("Price (GBP)")

# Set the title for the plot
ax_eth.set_title("Ethereum Price Fluctuations Over Time")

# Display the legend with labels for each line
ax_eth.legend()

# Show the plot
plt.show()

# Calculate the correlation ratio
correlation_ratio = df_bt[['Open', 'High', 'Low', 'Close']].corrwith(df_eth[['Open', 'High', 'Low', 'Close']])

# Print the correlation ratio values
print("Correlation Ratio Between Ethereum and Bitcoin Prices:")
print("Open:", correlation_ratio['Open'])
print("High:", correlation_ratio['High'])
print("Low:", correlation_ratio['Low'])
print("Close:", correlation_ratio['Close'])

# Initialize a MinMaxScaler
scaler_bt = MinMaxScaler()

# Extract the 'Close' prices from the Bitcoin DataFrame and reshape
```

```
close_price_bt = df_bt.Close.values.reshape(-1, 1)

# Scale the 'Close' prices using the scaler
scaled_close_bt = scaler_bt.fit_transform(close_price_bt)

# Obtain the shape of the scaled_close array
print("Shape of scaled_close_bt:", scaled_close_bt.shape)

# Check if there are any NaN values in the scaled_close array
print("Are there any NaN values in scaled_close_bt?", np.isnan(scaled_close_bt).any())

# Remove NaN values from the scaled_close array
cleaned_scaled_close_bt = scaled_close_bt[~np.isnan(scaled_close_bt)]

# Reshape the cleaned scaled_close array
reshaped_scaled_close_bt = cleaned_scaled_close_bt.reshape(-1, 1)

# Check if there are any NaN values in the reshaped_scaled_close array
print("Are there any NaN values in reshaped_scaled_close_bt?", np.isnan(reshaped_scaled_close_bt).any())

# Initialize a MinMaxScaler
scaler_eth = MinMaxScaler()

# Extract the 'Close' prices from the Ethereum DataFrame and reshape
close_price_eth = df_eth.Close.values.reshape(-1, 1)

# Scale the 'Close' prices using the scaler
scaled_close_eth = scaler_eth.fit_transform(close_price_eth)

# Obtain the shape of the scaled_close array
print("Shape of scaled_close_eth:", scaled_close_eth.shape)

# Check if there are any NaN values in the scaled_close array
print("Are there any NaN values in scaled_close_eth?", np.isnan(scaled_close_eth).any())

# Remove NaN values from the scaled_close array
cleaned_scaled_close_eth = scaled_close_eth[~np.isnan(scaled_close_eth)]

# Reshape the cleaned scaled_close array
reshaped_scaled_close_eth = cleaned_scaled_close_eth.reshape(-1, 1)

# Check if there are any NaN values in the reshaped_scaled_close array
print("Are there any NaN values in reshaped_scaled_close_eth?", np.isnan(reshaped_scaled_close_eth).any())

# Define the sequence length for input data
SEQ_LEN = 100

def to_sequences(data, seq_len):
    """
    Convert raw data into sequences of specified length.

    Parameters:
    data (numpy.ndarray): The raw data to be converted.
    seq_len (int): Length of each sequence.

    Returns:
    numpy.ndarray: Array of sequences.
    """
    sequences = []

    for index in range(len(data) - seq_len):
        sequences.append(data[index: index + seq_len])
```

```
    return np.array(sequences)

def preprocess(data_raw, seq_len, train_split):
    """
    Preprocess the raw data for training and testing.

    Parameters:
    data_raw (numpy.ndarray): The raw data to be preprocessed.
    seq_len (int): Length of each sequence.
    train_split (float): Percentage of data to be used for training.

    Returns:
    numpy.ndarray: Training features, training labels, testing features, testing labels.
    """
    # Convert raw data into sequences
    data_sequences = to_sequences(data_raw, seq_len)

    # Determine the number of samples for training
    num_train_samples = int(train_split * data_sequences.shape[0])

    # Split data into training and testing sets
    X_train = data_sequences[:num_train_samples, :-1, :]
    y_train = data_sequences[:num_train_samples, -1, :]

    X_test = data_sequences[num_train_samples:, :-1, :]
    y_test = data_sequences[num_train_samples:, -1, :]

    return X_train, y_train, X_test, y_test

# Preprocess Bitcoin data
X_train_bt, y_train_bt, X_test_bt, y_test_bt = preprocess(scaled_close_bt, SEQ_LEN, train_split=0.90)

# Preprocess Ethereum data
X_train_eth, y_train_eth, X_test_eth, y_test_eth = preprocess(scaled_close_eth, SEQ_LEN, train_split=0.90)

# Print the shapes of the preprocessed data for Bitcoin
print("Bitcoin Preprocessed Data Shapes:")
print("X_train_bt shape:", X_train_bt.shape)
print("y_train_bt shape:", y_train_bt.shape)
print("X_test_bt shape:", X_test_bt.shape)
print("y_test_bt shape:", y_test_bt.shape)

# Print the shapes of the preprocessed data for Ethereum
print("\nEthereum Preprocessed Data Shapes:")
print("X_train_eth shape:", X_train_eth.shape)
print("y_train_eth shape:", y_train_eth.shape)
print("X_test_eth shape:", X_test_eth.shape)
print("y_test_eth shape:", y_test_eth.shape)

"""# **Using LSTM Model for Cryptocurrency Price Prediction**"""

DROPOUT = 0.2
WINDOW_SIZE = SEQ_LEN - 1

# Build the LSTM model for Bitcoin
model_bt = Sequential()

model_bt.add(LSTM(WINDOW_SIZE, return_sequences=True, input_shape=(WINDOW_SIZE, X_train_bt.shape[-1])))
model_bt.add(Dropout(rate=DROPOUT))

model_bt.add(LSTM((WINDOW_SIZE * 2), return_sequences=True))
model_bt.add(Dropout(rate=DROPOUT))
```

```
model_bt.add(LSTM(WINDOW_SIZE, return_sequences=False))

model_bt.add(Dense(units=1))
model_bt.add(Activation('linear'))

# Compile the model for Bitcoin
model_bt.compile(loss='mean_squared_error', optimizer='adam')

# Train the model for Bitcoin
history_bt = model_bt.fit(
   X_train_bt, y_train_bt,
   epochs=50, batch_size=128,
   validation_split=0.1, verbose=1
)

# Build the LSTM model for Ethereum
model_eth = Sequential()

model_eth.add(LSTM(WINDOW_SIZE, return_sequences=True, input_shape=(WINDOW_SIZE, X_train_eth.shape[-1])))
model_eth.add(Dropout(rate=DROPOUT))

model_eth.add(LSTM((WINDOW_SIZE * 2), return_sequences=True))
model_eth.add(Dropout(rate=DROPOUT))

model_eth.add(LSTM(WINDOW_SIZE, return_sequences=False))

model_eth.add(Dense(units=1))
model_eth.add(Activation('linear'))

# Compile the model for Ethereum
model_eth.compile(loss='mean_squared_error', optimizer='adam')

# Train the model for Ethereum
history_eth = model_eth.fit(
   X_train_eth, y_train_eth,
   epochs=50, batch_size=128,
   validation_split=0.1, verbose=1
)

"""**Training and Validation Loss Visualization for Bitcoin and Ethereum Models**"""

# Plot the training and validation loss over epochs for Bitcoin
plt.figure(figsize=(10, 6))
plt.plot(history_bt.history['loss'], label='Train Loss')
plt.plot(history_bt.history['val_loss'], label='Validation Loss')
plt.title('Bitcoin Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Plot the training and validation loss over epochs for Ethereum
plt.figure(figsize=(10, 6))
plt.plot(history_eth.history['loss'], label='Train Loss')
plt.plot(history_eth.history['val_loss'], label='Validation Loss')
plt.title('Ethereum Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Evaluate the model for Bitcoin
loss_bt = model_bt.evaluate(X_test_bt, y_test_bt)
```

```
print(f"Bitcoin Test Loss: {loss_bt:.4f}")

# Make predictions for Bitcoin
y_pred_bt = model_bt.predict(X_test_bt)

# Inverse transform the predictions to get the original price scale for Bitcoin
predicted_prices_lstm_bt = scaler_bt.inverse_transform(y_pred_bt)
y_test_inverse_bt = scaler_bt.inverse_transform(y_test_bt)

# Evaluate the model for Ethereum
loss_eth = model_eth.evaluate(X_test_eth, y_test_eth)
print(f"Ethereum Test Loss: {loss_eth:.4f}")

# Make predictions for Ethereum
y_pred_eth = model_eth.predict(X_test_eth)

# Inverse transform the predictions to get the original price scale for Ethereum
predicted_prices_lstm_eth = scaler_eth.inverse_transform(y_pred_eth)
y_test_inverse_eth = scaler_eth.inverse_transform(y_test_eth)

"""**Comparing Actual and Predicted Cryptocurrency Prices**"""

# Plot actual and predicted prices for Bitcoin
plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_bt, label='Actual Prices', color='blue')
plt.plot(predicted_prices_lstm_bt, label='Predicted Prices', color='orange')
plt.title('Bitcoin Price Prediction')
plt.xlabel('Time(Days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()

# Plot actual and predicted prices for Ethereum
plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_eth, label='Actual Prices', color='blue')
plt.plot(predicted_prices_lstm_eth, label='Predicted Prices', color='orange')
plt.title('Ethereum Price Prediction')
plt.xlabel('Time(Days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()

"""**Evaluation Metrics for Cryptocurrency Price Predictions**"""

# Calculate evaluation metrics for Bitcoin
mse_lstm_bt = mean_squared_error(y_test_bt, y_pred_bt)
rmse_lstm_bt = np.sqrt(mse_lstm_bt)
mae_lstm_bt = mean_absolute_error(y_test_bt, y_pred_bt)
r2_lstm_bt = r2_score(y_test_bt, y_pred_bt)

# Print evaluation metrics for Bitcoin
print("Bitcoin Evaluation Metrics:")
print(f"Mean Squared Error (MSE): {mse_lstm_bt:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_lstm_bt:.4f}")
print(f"Mean Absolute Error (MAE): {mae_lstm_bt:.4f}")
print(f"R-squared: {r2_lstm_bt:.4f}")

# Calculate evaluation metrics for Ethereum
mse_lstm_eth = mean_squared_error(y_test_eth, y_pred_eth)
rmse_lstm_eth = np.sqrt(mse_lstm_eth)
mae_lstm_eth = mean_absolute_error(y_test_eth, y_pred_eth)
r2_lstm_eth = r2_score(y_test_eth, y_pred_eth)
```

```
# Print evaluation metrics for Ethereum
print("Ethereum Evaluation Metrics:")
print(f"Mean Squared Error (MSE): {mse_lstm_eth:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_lstm_eth:.4f}")
print(f"Mean Absolute Error (MAE): {mae_lstm_eth:.4f}")
print(f"R-squared: {r2_lstm_eth:.4f}")

"""# **Utilizing Random Forest Regression for Cryptocurrency Price Predictions**"""

# Reshape y_train and y_test for Bitcoin
y_train_bt = y_train_bt.ravel()
y_test_bt = y_test_bt.ravel()


# Create a Random Forest Regressor model with tuned hyperparameters for Bitcoin
model_rf_bt = RandomForestRegressor(
    n_estimators=250,
    max_depth=15,
    min_samples_split=4,
    min_samples_leaf=2,
    random_state=42,
    n_jobs=-1
)

# Train the model for Bitcoin
model_rf_bt.fit(X_train_bt.reshape(X_train_bt.shape[0], -1), y_train_bt)

# Reshape y_train and y_test for Ethereum
y_train_eth = y_train_eth.ravel()
y_test_eth = y_test_eth.ravel()

# Create a Random Forest Regressor model with tuned hyperparameters for Ethereum
model_rf_eth = RandomForestRegressor(
    n_estimators=200,
    max_depth=15,
    min_samples_split=4,
    min_samples_leaf=2,
    random_state=42,
    n_jobs=-1
)

# Train the model for Ethereum
model_rf_eth.fit(X_train_eth.reshape(X_train_eth.shape[0], -1), y_train_eth)

# Make predictions for Bitcoin
y_pred_bt = model_rf_bt.predict(X_test_bt.reshape(X_test_bt.shape[0], -1))

# Inverse transform the predictions to get the original price scale for Bitcoin
predicted_prices_rf_bt = scaler_bt.inverse_transform(y_pred_bt.reshape(-1, 1))
y_test_inverse_bt = scaler_bt.inverse_transform(y_test_bt.reshape(-1, 1))


# Make predictions for Ethereum
y_pred_eth = model_rf_eth.predict(X_test_eth.reshape(X_test_eth.shape[0], -1))

# Inverse transform the predictions to get the original price scale for Ethereum
predicted_prices_rf_eth = scaler_eth.inverse_transform(y_pred_eth.reshape(-1, 1))
y_test_inverse_eth = scaler_eth.inverse_transform(y_test_eth.reshape(-1, 1))

# Plot actual and predicted prices for Bitcoin
plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_bt, label='Actual Prices', color='blue')
plt.plot(predicted_prices_rf_bt, label='Predicted Prices', color='orange')
```

```
plt.title('Bitcoin Price Prediction using Random Forest Regression')
plt.xlabel('Time(Days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()

# Plot actual and predicted prices for Ethereum
plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_eth, label='Actual Prices', color='blue')
plt.plot(predicted_prices_rf_eth, label='Predicted Prices', color='orange')
plt.title('Ethereum Price Prediction using Random Forest Regression')
plt.xlabel('Time(Days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()

# Calculate evaluation metrics for Bitcoin predictions
mse_rf_bt = mean_squared_error(y_test_bt, y_pred_bt)
rmse_rf_bt = np.sqrt(mse_rf_bt)
mae_rf_bt = mean_absolute_error(y_test_bt, y_pred_bt)
r2_rf_bt = r2_score(y_test_bt, y_pred_bt)

# Print evaluation metrics for Bitcoin predictions
print("Bitcoin Evaluation Metrics:")
print(f"Mean Squared Error (MSE): {mse_rf_bt:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_rf_bt:.4f}")
print(f"Mean Absolute Error (MAE): {mae_rf_bt:.4f}")
print(f"R-squared: {r2_rf_bt:.4f}")

# Calculate evaluation metrics for Ethereum predictions
mse_rf_eth = mean_squared_error(y_test_eth, y_pred_eth)
rmse_rf_eth = np.sqrt(mse_rf_eth)
mae_rf_eth = mean_absolute_error(y_test_eth, y_pred_eth)
r2_rf_eth = r2_score(y_test_eth, y_pred_eth)

# Print evaluation metrics for Ethereum predictions
print("Ethereum Evaluation Metrics:")
print(f"Mean Squared Error (MSE): {mse_rf_eth:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_rf_eth:.4f}")
print(f"Mean Absolute Error (MAE): {mae_rf_eth:.4f}")
print(f"R-squared: {r2_rf_eth:.4f}")

"""# **Using Bidirectional LSTM for Cryptocurrency Price Prediction**"""

# Create a Sequential model for Bitcoin
model_bt = keras.Sequential()
model_bt.add(Bidirectional(LSTM(WINDOW_SIZE, return_sequences=True), input_shape=(WINDOW_SIZE,
X_train_bt.shape[-1])))
model_bt.add(Dropout(rate=DROPOUT))
model_bt.add(Bidirectional(LSTM((WINDOW_SIZE * 2), return_sequences=True)))
model_bt.add(Dropout(rate=DROPOUT))
model_bt.add(Bidirectional(LSTM(WINDOW_SIZE, return_sequences=False)))
model_bt.add(Dense(units=1))
model_bt.add(Activation('linear'))

# Compile the model for Bitcoin
model_bt.compile(loss='mean_squared_error', optimizer='adam')

# Create a Sequential model for Ethereum
model_eth = keras.Sequential()
model_eth.add(Bidirectional(LSTM(WINDOW_SIZE, return_sequences=True), input_shape=(WINDOW_SIZE,
X_train_eth.shape[-1])))
model_eth.add(Dropout(rate=DROPOUT))
```

```
model_eth.add(Bidirectional(LSTM((WINDOW_SIZE * 2), return_sequences=True)))
model_eth.add(Dropout(rate=DROPOUT))
model_eth.add(Bidirectional(LSTM(WINDOW_SIZE, return_sequences=False)))
model_eth.add(Dense(units=1))
model_eth.add(Activation('linear'))

# Compile the model for Ethereum
model_eth.compile(loss='mean_squared_error', optimizer='adam')

"""**Train the Bidirectional LSTM Models**"""

# Train the model for Bitcoin
history_bt = model_bt.fit(
    X_train_bt,              # Training input data
    y_train_bt,              # Target output data
    epochs=50,               # Number of training epochs
    batch_size=128,          # Batch size for each iteration
    shuffle=False,           # Don't shuffle the data
    validation_split=0.1     # Fraction of data used for validation
)

# Train the model for Ethereum
history_eth = model_eth.fit(
    X_train_eth,             # Training input data
    y_train_eth,             # Target output data
    epochs=50,               # Number of training epochs
    batch_size=128,          # Batch size for each iteration
    shuffle=False,           # Don't shuffle the data
    validation_split=0.1     # Fraction of data used for validation
)

"""**Evaluate the Bidirectional LSTM Models**"""

# Evaluate the model for Bitcoin
loss_bt = model_bt.evaluate(X_test_bt, y_test_bt)
print(f"Test Loss for Bitcoin: {loss_bt:.4f}")

# Evaluate the model for Ethereum
loss_eth = model_eth.evaluate(X_test_eth, y_test_eth)
print(f"Test Loss for Ethereum: {loss_eth:.4f}")

# Plot Training and Validation Loss for Bitcoin

plt.figure(figsize=(10, 6))
plt.plot(history_bt.history['loss'], label='Train Loss')
plt.plot(history_bt.history['val_loss'], label='Validation Loss')
plt.title('Bitcoin Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()

# Plot Training and Validation Loss for Ethereum

plt.figure(figsize=(10, 6))
plt.plot(history_eth.history['loss'], label='Train Loss')
plt.plot(history_eth.history['val_loss'], label='Validation Loss')
plt.title('Ethereum Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
"""**Make Predictions using the Bidirectional LSTM Model**"""

# Predict prices for Bitcoin
y_pred_bt = model_bt.predict(X_test_bt)

# Inverse transform the predictions to get original price scale
y_test_inverse_bt = scaler_bt.inverse_transform(y_test_bt.reshape(-1, 1))
y_pred_inverse_bi_lstm_bt = scaler_bt.inverse_transform(y_pred_bt.reshape(-1, 1))

# Plot actual and predicted prices for Bitcoin
plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_bt, label='Actual Prices', color='green')
plt.plot(y_pred_inverse_bi_lstm_bt, label='Predicted Prices', color='blue')
plt.title('Bitcoin Price Prediction')
plt.xlabel('Time(Days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()

# Predict prices for Ethereum
y_pred_eth = model_eth.predict(X_test_eth)

# Inverse transform the predictions to get original price scale
y_test_inverse_eth = scaler_eth.inverse_transform(y_test_eth.reshape(-1, 1))
y_pred_inverse_bi_lstm_eth = scaler_eth.inverse_transform(y_pred_eth.reshape(-1, 1))

# Plot actual and predicted prices for Ethereum
plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_eth, label='Actual Prices', color='green')
plt.plot(y_pred_inverse_bi_lstm_eth, label='Predicted Prices', color='blue')
plt.title('Ethereum Price Prediction')
plt.xlabel('Time(Days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()

"""**Evaluate the Bidirectional LSTM Models**"""

# Evaluate Bitcoin Model
mse_bi_lstm_bt = mean_squared_error(y_test_bt, y_pred_bt)
rmse_bi_lstm_bt = np.sqrt(mse_bi_lstm_bt)
mae_bi_lstm_bt = mean_absolute_error(y_test_bt, y_pred_bt)
r2_bi_lstm_bt = r2_score(y_test_bt, y_pred_bt)

print("Bitcoin Model Evaluation:")
print(f"Mean Squared Error (MSE): {mse_bi_lstm_bt:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_bi_lstm_bt:.4f}")
print(f"Mean Absolute Error (MAE): {mae_bi_lstm_bt:.4f}")
print(f"R-squared: {r2_bi_lstm_bt:.4f}")

# Evaluate Ethereum Model
mse_bi_lstm_eth = mean_squared_error(y_test_eth, y_pred_eth)
rmse_bi_lstm_eth = np.sqrt(mse_bi_lstm_eth)
mae_bi_lstm_eth = mean_absolute_error(y_test_eth, y_pred_eth)
r2_bi_lstm_eth = r2_score(y_test_eth, y_pred_eth)

print("Ethereum Model Evaluation:")
print(f"Mean Squared Error (MSE): {mse_bi_lstm_eth:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse_bi_lstm_eth:.4f}")
print(f"Mean Absolute Error (MAE): {mae_bi_lstm_eth:.4f}")
print(f"R-squared: {r2_bi_lstm_eth:.4f}")

"""# Comparing Model Performances and Selecting the Best Model"""
```

```
# Compare RMSEs and Select the Best Model
if rmse_lstm_bt < rmse_bi_lstm_bt and rmse_lstm_bt < rmse_rf_bt:
    best_model_bt = "LSTM"
    best_rmse_bt = rmse_lstm_bt
elif rmse_bi_lstm_bt < rmse_lstm_bt and rmse_bi_lstm_bt < rmse_rf_bt:
    best_model_bt = "Bidirectional LSTM"
    best_rmse_bt = rmse_bi_lstm_bt
else:
    best_model_bt = "Random Forest Regressor"
    best_rmse_bt = rmse_rf_bt

if rmse_lstm_eth < rmse_bi_lstm_eth and rmse_lstm_eth < rmse_rf_eth:
    best_model_eth = "LSTM"
    best_rmse_eth = rmse_lstm_eth
elif rmse_bi_lstm_eth < rmse_lstm_eth and rmse_bi_lstm_eth < rmse_rf_eth:
    best_model_eth = "Bidirectional LSTM"
    best_rmse_eth = rmse_bi_lstm_eth
else:
    best_model_eth = "Random Forest Regressor"
    best_rmse_eth = rmse_rf_eth

print(f"Best Model for Bitcoin: {best_model_bt} (RMSE: {best_rmse_bt:.4f})")
print(f"Best Model for Ethereum: {best_model_eth} (RMSE: {best_rmse_eth:.4f})")

#plot of predicted and actual price together
# Plot actual and predicted prices for Bitcoin

plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_bt, label='Actual Prices', color='green')
plt.plot(predicted_prices_lstm_bt, label='LSTM Predicted Prices', color='blue')
plt.plot(y_pred_inverse_bi_lstm_bt, label='Bi-LSTM Predicted Prices', color='orange')
plt.plot(predicted_prices_rf_bt, label='Random Forest Predicted Prices', color='red')
plt.title('Bitcoin Price Prediction')
plt.xlabel('Time(days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()

# Plot actual and predicted prices for Ethereum
plt.figure(figsize=(12, 6))
plt.plot(y_test_inverse_eth, label='Actual Prices', color='green')
plt.plot(predicted_prices_lstm_eth, label='LSTM Predicted Prices', color='blue')
plt.plot(y_pred_inverse_bi_lstm_eth, label='Bi-LSTM Predicted Prices', color='orange')
plt.plot(predicted_prices_rf_eth, label='Random Forest Predicted Prices', color='red')
plt.title('Ethereum Price Prediction')
plt.xlabel('Time(days)')
plt.ylabel('Price(GBP)')
plt.legend()
plt.show()
```