MSci Data Science
Project Report
2023-24

Bitcoin Trading Bot in Python

Author: Mohamad Kanso (Mohamad.kanso@city.ac.uk)

Supervisor: Dr Jacob Howe

The project idea was proposed by the Author.
No proprietary arrangements exist.
Word Count of Main Body: **14994**

Abstract

The development of cryptocurrencies has changed financial markets, bringing in opportunities and concerns due to the volatility of assets such as Bitcoin. Recently, there has been a demand for robust trading methods tuned to the circumstances of cryptocurrency markets, as traditional trading strategies have proven unsuccessful mainly due to human error. Hence, this project aimed to develop an automated Bitcoin Trading Bot using machine learning techniques like Long Short-Term Memory (LSTM) networks to evaluate live and historical market data. These networks allow the bot to predict market moves and place trades planned to increase profits and reduce risk.

The trading bot accessed data and executed trades via the Binance API (Application Programming Interface). It ran live, so all trades were processed immediately. The bot's structure focused mainly on Profound Exploratory Data Analysis (EDA), with a study on understanding technical indicators like Moving Averages, the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD). This study provides the knowledge required to change the LSTM model, enhancing the bot's abilities and instilling confidence in its operation.

This report explains the trading bot's continuous design process, incorporating training models, backtesting with historical data, and live trading tests to assess its performance. The outcomes demonstrated that the trading bot can adaptively learn from market price fluctuations and execute trades based on predictions. The potential impact of this project is significant, potentially improving the strategic decisions of individual traders and financial organisations by offering a tool for streamlining and optimising bitcoin trading methods, thereby fostering a sense of optimism and intrigue.

Contents

1. Introduction

1.1 Problem Description

The rise of cryptocurrencies, specifically Bitcoin, has significantly changed the environment for financial trading (Nakamoto, 2008). Primarily offered as a decentralised asset, Bitcoin established a new idea in the financial markets that described the lack of centralised control and digital absence. This has presented specific challenges for traders because of the volatility, short history and complexity of blockchain technology related to Bitcoin. Traditional trading strategies—which regularly depend on stable market conditions and predictable patterns—usually prove unsuccessful in the volatile and active cryptocurrency market (Abakah et al., 2020). This calls for developing progressive tools that effectively analyse traders' data and adapt to this energetic environment.

This study uncovers an innovative creation: a Python Trading Bot explicitly designed for the volatile and dynamic cryptocurrency markets. This technology product has been precisely crafted to steer the instability and rapid price changes often leading to 'decision paralysis' or poor trading judgements. The bot tries to better the trading experience by overcoming these challenges, offering a more efficient and practical approach to cryptocurrency trading.

1.2 Research Objectives and Project Scope

This research aims to address the aforementioned issues by developing a trading bot capable of automated trading decisions. The bot executes trades using predictive analytics obtained from both real-time and historical market data. This automatic process decreases human error seen with manual trading while increasing decision-making speed—a factor in exploiting short-lived trading opportunities in a high-volatility environment.

Hence, the main goal was to create a trading bot that executes trades based on prediction algorithms. This involves the bot identifying trends in past data and seamlessly adapting to new, unforeseen market situations in real-time, instilling traders with confidence in its performance.

1.2.1 Secondary Objectives

1. Using Long Short-Term Memory (LSTM) networks to calculate swings in Bitcoin prices. The decision to go with LSTM is based on its ability to detect the order dependency in sequence prediction issues, which has to be maintained to understand and predict time series data of bitcoin prices (Bouri et al., 2021). The initial idea of the LSTM model was to predict market trends by using technical indicators such as Moving Averages (MA),

the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD).

2. Integrate and test the bot under real market circumstances using the Binance trading platform. Evaluate performance and make incremental changes based on empirical data.

1.2.2 Research Questions

- How well do LSTM networks predict bitcoin price movements?
- Can LSTM networks constantly predict price changes based on technical indicators like MACD, RSI and MA?
- Determine ideal parameters and settings for the LSTM model to improve trading efficiency and accuracy.
- How does a Linear Regression model compare to LSTM in predicting bitcoin prices?
- Can the Naïve Baseline model compete with more complicated predictive models for accuracy and profitability in real-time trading?
- How do LSTM, Linear Regression and Naïve Baseline models perform in real-world trading scenarios?

1.3 Beneficiaries and Benefits

This research seeks to offer technological advances that can impact individual traders and financial companies operating in the Bitcoin market. The trading bot is expected to reduce traders' mental stress, eliminate errors, and enhance trading outcomes by simplifying the trading process while providing predictive insights, thus increasing its market significance (Binance Academy, 2021).

1.4 Work Performed

To research the goals set, a detailed Exploratory Data Analysis (EDA) of Bitcoin price changes was implemented (Chan & Lakonishok, 1993). Calculating technical indicators like Moving Averages (MA), Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) helped construct prediction models. As the project continued, the outcomes section indicated that these signals may not generate the most accurate or profitable results. This changed the strategy, showing the LSTM model's potential to learn from larger datasets without relying on technical indicators. The project evolved to improve the bot's performance in active markets, improving trading efficiency.

Baseline models like Linear Regression and Naïve Baseline models were developed alongside the LSTM model to compare complex models. As a control group, these baseline models captured the LSTM's increased capability and areas for improvement. LSTM and Linear

Regression models were backtested using historical data to simulate trading scenarios and assess their performance. They were backtesting corrected models before deployment.

Using the Binance API, the LSTM and Linear Regression models were set up in a live trading environment for real-time testing and analysis. The models' performance in a harsh market relied on this real-world trading phase. Essential input on the models' real-time trading performance allowed constant adjustments and optimisations to improve trading strategies and results. The real use of these models demonstrated they could automate trading. It explained their operational strengths and weaknesses, showing the trading bot's ongoing development.

1.5 Assumptions and Scope Limitations

This research relied on the Binance API's real-time gathering of information and LSTM models' price trend prediction based on historical data. Bitcoin was selected as the project's first dataset for model training and testing of its popularity and liquidity in the cryptocurrency market (Kroll et al., 2013).

The project Definition Document (PDD) aims changed significantly during the project. The trading bot initially planned to leverage technical indicators like MA, RSI, and MACD to predict Bitcoin fluctuations using the LSTM model. As the project entered integration and live testing, it became obvious that relying solely on these signs would limit forecast performance under chaotic market situations.

The shortcomings included the indicators' failure to accurately represent complicated market dynamics and their vulnerability to misleading signals. This realisation led to the trading bot's structure being re-evaluated and modified to use a complete data entry method in the LSTM model instead of these indicators. With the ability to learn from larger datasets and understand order dependency in sequence prediction tasks, the LSTM model proved more resilient and flexible to these constraints.

These findings were acquired by building the trading bot in real-time utilising the Binance API. Live market conditions tested the bot's early configurations and performance, exposing areas for forecast accuracy and efficiency improvement. Test results allow us to refine model parameters, including input features and LSTM layers. These improved the bot's ability to respond to market data. Changing input features permitted the model to catch the most crucial market signals while modifying LSTM layers improved learning and prediction. These exchanges created a reliable trading mechanism to handle the complexity of the bitcoin markets.

2. Output Summary

2.1 Output 1: Exploratory Data Analysis (EDA) Results (1 File)

File Type	.ipynb
Description	The project used Bitcoin price data for a thorough Exploratory Data Analysis (EDA). To understand the market structure, the study examined price movements, volatility, stationarity tests, and technical indicators, including MA, RSI, and MACD.
Output	EDA outcomes result from a comprehensive process involving more than 350 lines of Python code and advanced libraries such as NumPy, Pandas, Matplotlib, statsmodels, Binance.client and Seaborn. The EDA was conducted using Bitcoin price data from the Binance API, providing a robust and reliable analysis.
End – User	Examiners and anyone who wishes to continue research or any trader curious about market analysis.
Usage and Benefits	The EDA collects bitcoin price actions to help users see critical trends and patterns. Identifying key indicators and their effects helps traders manage risk and capitalise on trading opportunities.
Results	Chapter 5.1 of the report discusses the analysis in detail, with graphs and statistics.

2.2 Output 2: Development of Predictive Models. (4 Files)

File Type	.ipynb
Description	A diverse range of predictive models were employed, including two LSTM models, one Linear Regression model, and a Naïve Baseline model. This variety ensures a comprehensive analysis and a more accurate estimation of Bitcoin values based on historical data.
Output	Outputs include evaluations, visualisations, and model specifications. The complicated LSTM models use the TensorFlow and Keras libraries and 300 Python lines. The more straightforward Linear Regression and Naïve Baseline models use sci-kit-learn and 100 lines of Python code.
End – User	Examiners, data scientists, and machine learning engineers developing financial prediction systems.
Usage and Benefits	These algorithms provide accurate price predictions to help traders make better decisions and increase profits.
Results	Chapters 5.2 and 5.3 of the report explain model performance and operations.

2.3 Output 3: Backtesting Reports (1 File)

File Type	.ipynb
Description	The backtesting reports provide real-world evidence of the models' reliability and accuracy. The LSTM and Linear Regression models were tested under market scenarios using historical Bitcoin price data, demonstrating their practical application and effectiveness.
Type of Output	Evaluation report and visualisations. The backtesting involved logs and performance metrics over multiple time frames, ending in extensive reports and graphs. This code has about 250 lines.
End – User	Risk managers and institutional investors studying the effectiveness of trading strategies.
Usage and Benefits	Using historical Bitcoin price data, the LSTM and Linear Regression models' backtesting provides real-world evidence of their reliability and price prediction accuracy. This reassures users about the models' performance, helping them confidently assess risk and ROI.
Results	The outcomes of the backtesting are presented in Chapter 5.4.

2.4 Output 4: Real-Time Trading with the Binance API (2 Files)

File Type	.ipynb
Description	Integration of the predictive models with the Binance API to execute real-time trading based on the models' predictions. The connection between Python and the API allows actual trades to take place.
Type of Output	Establishes a trading system. Integration involves over 200 lines of Python code, handling API requests, and response management with robust error handling.
End – User	Cryptocurrency traders and investors who would like to automate their trading strategies.
Usage and Benefits	Automating buy and sell orders based on predictive data helps trading and profitability. It also removes human oversight and trading, assisting users in managing market opportunities.
Results	Chapters 5.5 and 5.6 discuss the implementation details and evaluate real-time trading performance.

3. Literature Review

3.1 Introduction to Cryptocurrency and Bitcoin

Cryptocurrencies are a major benchmark in digital finance, providing a decentralised payment method and potential for investment. Nakamoto's creation of Bitcoin in 2008 brought in an entirely new phase of financial technology, which, as clarified by its core blockchain technology, offers transaction security and openness. Bitcoins are defined by their lack of physical form and dependence on cryptographic procedures to safeguard transactions, making them more volatile than traditional fiat currencies (Baur, Hong, & Lee, 2018).

Being the first and most well-known cryptocurrency, Bitcoin shows all these characteristics, setting the norm for future digital currencies. Its market behaviour is known for its severe volatility, which may be assigned to various variables, such as market mood and technological advancements (Cheah and Fry, 2015). Even though volatility creates obstacles for traders and investors, it also allows for a distinctive possibility of creating trading algorithms capable of manoeuvring through such an unstable market.

3.2 Predictive Modelling in Financial Markets

Predictive modelling has become essential for the unsteady nature of Bitcoin for creating profitable trading strategies. Standard models that include ARIMA have problems when it comes to predicting cryptocurrencies like Bitcoin because of their criteria of linearity and stationarity, which are often violated in these markets. Kristoufek (2015) has delivered attention to this topic by stressing the complex and nonlinear characteristics of the price swings in Bitcoin.

Researchers have moved on to using advanced non-linear models, which are much better at handling the sudden changeability in bitcoin markets after realising these limitations. Machine learning models are becoming increasingly popular, specifically those representing non-linear interactions. With their ability to adjust constantly to the unpredictable trends typical of cryptocurrencies, these models provide a stronger foundation for financial forecasting in this industry. Zhang et al. (2019) offer more examples of the successful application of deep learning models, such as Convolutional Neural Networks (CNNs), to financial time series predicting, showing these models' potential to handle complicated market price patterns and succeed in complex data structures.

3.3 Use of LSTM Networks for Price Prediction

Long Short-Term Memory (LSTM) networks are famous for their ability to solve sequence prediction issues, which are prevalent in financial markets where previous price data often

directs future patterns. Traditional recurrent neural networks (RNNs) are commonly affected by problems like the vanishing gradient problem, softened by the unique design of LSTM networks. Hochreiter and Schmidhuber (1997), who introduced LSTMs as a ground-breaking way to preserve information over long periods without running the danger of losing important data in gradients through training processes, explain this skill nicely.

McNally et al. (2018) provided more concrete proof for the effectiveness of LSTMs in financial domains by showing their capacity to predict Bitcoin values more accurately than traditional models like GARCH. This finding is crucial for us since cryptocurrency markets are random and highly volatile, pushing the use of models that can swiftly adjust to sudden shifts in the market. The usefulness of LSTMs in such circumstances shows how well they can handle additional layers of information, such as psychological emotion trades and macroeconomic factors, of course, in addition to price and volume data.

The literature's theoretical foundations and observed proofs position LSTMs as a far-ahead model for building complex predictive models in the financial technology industry. These models are on the frontline for building trading bots that function well in the complex and shifting world of bitcoin trading.

3.4 Role of Exploratory Data Analysis (EDA) in Model Selection

Exploratory Data Analysis, or EDA, is important to understanding the motions of financial markets, notably in cryptocurrency trading. EDA may yield a large number of findings on Bitcoin's price volatility, trend patterns and impact on different market factors. The examination of data is needed for this analytical phase, as it helps uncover underlying patterns that might not be obvious at first glance.

The relevance of EDA - in the context of financial time series is emphasised in the literature. According to Tsay (2005), EDA plays a significant role in modelling by pointing out strange events or structural breakdowns in markets that could impact predictive models. This would be useful for cryptocurrency markets, as unexpected price changes and issues are frequent. Furthermore, Brockwell and Davis (2002) state that before any modelling can be used successfully, it is a must to understand the features of financial time series using EDA, showing the importance of such prior inquiries.

Because they offer tangible information on market trends and momentum, technical indicators like Moving Averages (MA), Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) are usually studied throughout EDA periods (Murphy, 1999). It was shown that these indicators are predictive, making them helpful when developing financial forecasting models. Researchers can evaluate these indicators during the EDA and determine which characteristics could improve model performance.

Moreover, results from EDA influence the integration of machine learning models, especially LSTM networks. According to Géron (2017), long short-term memory (LSTM) models are very good at handling sequence dependencies in time series data. This makes them perfect for financial applications where historical price data predicts future patterns. When non-linear relationships and long-term dependencies are found in the data using EDA, the decision to use LSTM models is frequently made (Brownlee, 2018).

According to the literature, many other data sources are used to improve the prediction models' performance. For instance, Patel et al. (2015) examine how trading models may be improved regarding forecast accuracy by incorporating investor sentiment analytics and macroeconomic indicators. The findings align with the study conducted by Kim and Kim (2020), which shows that the value of models in predicting cryptocurrency prices could be greatly affected by external data, such as market sentiment indices.

EDA provides a complete picture of the financial environment and influences the decision-making and structure of trading models. It runs beyond simple data cleaning and preparation. Building solid, successful AI models for financial forecasting requires a foundation of understanding provided by this kind of investigation.

3.5 The Role of Open-Source Software in Data Science

Open-source software has changed the field of data science by expanding access to good analytical tools for more people. As Dhar (2013) stated, Python and R have levelled the playing field by providing items previously limited to expensive and exclusive software. It is vital for areas in finance where creating methods and prediction models may be costly. Donoho (2017) also underlines the value of open-source locations by writing about how they promote repeatable studies by proposing a wide choice of tools and libraries that are continuously updated and improved by the international developer community.

NumPy, pandas and sci-kit-learn are just a few of the strong data manipulation and machine learning libraries available in Python. These libraries are important for financial modelling since they allow for quickly developing predictive models and complicated analysis. McKinney (2012) says incorporating these tools into the data science process allows a fluid, easy and open method of developing models and exploring data (EDA).

3.6 Best Practices in Analytical Reproducibility

Critical repeatability is vital to credible scientific research in computational finance, where predictive models frequently prompt decision-making. The need for reproducibility in computational research is emphasised by Peng (2011), who also promotes the use of tools like Jupyter Notebooks, which make it possible to combine computing, data analysis and rich text features. Furthermore, Stodden et al. (2016) lecture on the difficulties and methods for

accessing reproducibility, arguing that data and code documentation are essential to the conduct of science so that it may be consistently accessed and expanded upon by others' ideas.

Ioannidis et al. (2014) emphasise the importance of these methods, saying that improving the openness of scientific findings is just as important as being able to duplicate results. This is important in fields like data science, where the difficulty of models can make it hard to grasp how conclusions are arrived at without thorough documentation.

4. Methods

Figure 1 presents the pathway used to meet the project's objective.

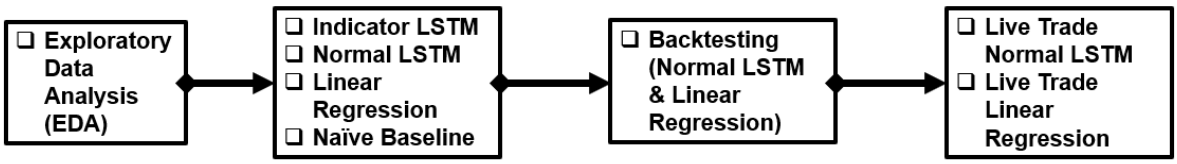


Figure 1: Methodology flowchart for Bitcoin price prediction study.

4.1 Exploratory Data Analysis (EDA)

The Exploratory Data Analysis (EDA) began with meticulously collecting high-resolution Bitcoin price data from the Binance API. We chose this API for its 15-minute interval granularity, a crucial feature allowing us to precisely portray the cryptocurrency market's inherent volatility. This precision is critical to capturing macro and micro market movements and is pivotal for training our prediction model.

4.1.1 Data Preprocessing

We leveraged the Binance Client to obtain 1000 Bitcoin price data points, each containing the open, high, low, and closing prices (OHLC) and volume at 15-minute intervals. This dataset was then processed to extract closure prices and timestamps, which were converted from UNIX timestamps to a human-readable format for easy analysis. Using a Pandas data frame significantly enhanced the efficiency and effectiveness of our future data processing and visualisation operations, a practical benefit that streamlined our work.

4.1.2 Data Visualisation

Matplotlib was used to make the first plots of Bitcoin prices. This was done as an initial step to see how volatile the prices were over time and to set the stage for more in-depth studies.

4.1.3 Descriptive Statistics

We then produced descriptive statistics using Pandas' built-in capabilities that gave a comprehensive quantitative dataset overview. We highlighted the Bitcoin price distribution's shape, central tendency, and dispersion across the chosen time frame.

4.1.4 Technical Indicator Calculation

Our strategy was to eliminate the need for external libraries (such as Talib) for various reasons: to ensure complete control over the computational logic, enhance the clarity of our methods, and ensure the ability to reproduce our results.

4.1.5 Moving Averages

We computed Moving Averages (MA) to flatten out short-term fluctuations and focus on longer-term trends (Suliman, Elmawazini, and Shariff, 2015). In particular, we computed the 30-day and 7-day Moving Averages (MAs), which gave us data on short- and long-term market motion. These time intervals were selected as likely buy or sell signal indicators based on their consistent usage in financial trading. This model development and evaluation approach allows us to have the robustness we seek for our predictions.

4.1.6 Relative Strength Index (RSI)

The RSI was calculated as a momentum oscillator that monitors the rate and change of price movements. It moves between zero and 100. Traditionally, and as accepted in our project, an asset is judged overbought when the RSI is above 70 and oversold when it is below 30. This is a significant indicator for us since it can indicate a market reversal, which is essential for our predictive modelling.

4.1.7 Moving Average Convergence Divergence (MACD)

The MACD is a trend-following momentum indicator that shows the relationship between two moving averages of an asset's price; it played a crucial role in our trend analysis. We calculated the MACD by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA. The result of this calculation is the MACD line. A nine-day EMA of the MACD, called the 'signal line,' was then plotted on top of the MACD line, which can function as a trigger for buy and sell signals (Edgewater Academy, n.d.). This indicator is useful as it is a tool for predicting market trends.

4.1.8 Methodological Rigor and Justification

Our decision to develop custom functions was twofold: to avoid black-box analytics, where the inner workings of the calculations are hidden, and to promote understanding of the indicators themselves. We could tailor the calculations to our specific dataset by coding these functions.

By detailing the exact formulas and windows used in our calculations, we guarantee a precise replication of our research, meeting the high standards of reproducibility required in scientific inquiry. This approach demonstrates a high level of understanding and application of the financial theories underpinning these indicators, further adding credibility and robustness to our methodological framework.

4.1.9 Feature Engineering and Correlation Analysis

Developing engineered features like moving averages and technical indicators like RSI and MACD was essential. A correlation study was performed on these attributes to find any trends that might indicate future changes in the price of Bitcoin. Seaborn supplied a heatmap

visualisation that helped us understand the strength of these associations and informed our choice of features for further research.

4.1.10 Volatility Analysis

Understanding the volatility of Bitcoin was critical, which we accomplished by computing the standard deviation of price movements. This volatility research helped measure the market's variability and was crucial in picking an AI model that efficiently deals with such variations. Long Short-Term Memory (LSTM) models, well-known for their ability to manage sequence prediction in uncertain contexts, emerged as leading possibilities. This investigation ensured that the chosen model was strong enough to react to the rapid price swings in the bitcoin market, improving the trading bot's predicted accuracy and reliability (Brown, 2022).

4.1.11 Stationarity Testing

Since stationarity in time series data is crucial for predictive modelling, we used the Augmented Dickey-Fuller (ADF) test to see if our Bitcoin price time series had a unit root. Stationarity is a critical feature that shows the statistical properties of the series remain constant across time. Non-stationarity, which might take the form of trends, seasonality, and fluctuating variances, can seriously undermine the accuracy of prediction models.

The ADF test is a formal statistical approach for detecting unit roots, providing a mechanism for determining whether a time series is stationary around a mean, following a linear trend, or completely non-stationary (Kim and Lee, 2020). The test's framework is intended to give a systematic approach to this assessment based on outputs:

- The ADF Statistic indicates that a series may be stationary if strongly negative and distant from zero.
- A p-value is a probability score that indicates the likelihood of detecting test findings under the null hypothesis. A lower p-value (<0.05) usually suggests that the series is stationary.
- The ADF statistic has critical values at 1%, 5%, and 10% significance levels. Exceeding these thresholds indicates the series may be stationary.

4.1.12 Memory Effect Analysis

Autocorrelation analysis was utilised to identify the memory effect in the Bitcoin price series. This evaluates the impact of previous price actions on future prices, offering fundamental knowledge for selecting a model such as LSTM, which is well-known for its ability to take advantage of these temporal dependencies. This memory effect must be understood and applied.

For example, significant autocorrelation at early lags (e.g., 0.85 at lag 1) means that recent Bitcoin prices strongly influence future prices. This makes LSTM models helpful because they thrive at detecting and predicting short-term trends by prioritising recent data, resulting in greater accuracy in forecasting trading strategies (Chen, 2021).

4.1.13 Trend Analysis with Decomposition

The methodological approach for analysing Bitcoin price trends involved utilising the seasonal_decompose function from the statsmodels library, which enabled a detailed breakdown of the time series data into trend, seasonal and residual components (InMobi Technology, 2022). This decomposition was crucial for isolating the overarching direction of Bitcoin prices (trend), identifying recurring patterns or cycles (seasonal), and capturing irregularities that could not be attributed to trend or seasonality (residual). This breakdown clarified the different forces influencing price changes and informed the selection and tuning of predictive models by highlighting when and how Bitcoin prices were likely to change. This comprehensive understanding of market dynamics is pivotal for developing robust financial models.

4.1.14 Conclusion

The EDA presented a complete analysis that thoroughly prepared the dataset for predictive modelling. This approach facilitated an in-depth understanding of Bitcoin's price dynamics and laid a foundation for applying the LSTM model, aligning with our project's objectives to harness machine learning for financial forecasting.

4.2 Model Development and Evaluation

To maintain equality in evaluation and improve comparability between models, all models used in this study were given the same data split and evaluated using the same metrics. The data was divided into training and testing sets at a ratio of 80:20 (Coventry University, n.d.). This standard split was used for the linear regression model, two LSTM models (Normal and Indicator), and the Naïve Baseline Model. Using a constant split across all models confirms that each model is tested under the same conditions, allowing for a fair comparison of prediction performance.

4.2.1 Data Preparation

The first step in the data preparation procedure was to obtain high-frequency Bitcoin price data at 15-minute intervals via the Binance API. This data was carefully cleaned and normalised to guarantee consistency across all characteristics. This is vital for helpful model input. Timestamps were transformed to a machine-readable format, and time-based information such as hour and day were extracted to improve the models' capacity to detect temporal patterns.

This thorough preparation is needed for all the models, including the Linear Regression model, which requires structured input for its predictive analyses and the LSTM models, which immediately incorporate sequence data.

4.2.2 Model Evaluation Methodology

The evaluation of all models, except the Indicators LSTM Model, was conducted with great attention to detail using various systematic evaluation methods:

- Time-based cross-validation is an essential tool for LSTM models to avoid lookahead bias and ensure that predictions are exclusively based on past data.
- Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are metrics used to evaluate predictions' accuracy and average errors. They provide valuable data on the precision of the models.
- R-squared (R^2) is an important statistic for evaluating Linear Regression models. It determines the amount of variance in the dependent variable that can be predicted from the independent variable.
- The Mean Absolute Percentage Error (MAPE) and Explained Variance Score are metrics used to evaluate models' accuracy and explanatory capability on a given dataset.
- The Residual Analysis Plot visually examines the predictions' residuals to identify patterns suggesting issues with the model's fit.

Adding the Naïve Baseline Model to our research was needed, as it set fundamental performance criteria. This model, which utilises the most recent observed value to predict future values, provides a benchmark for evaluating the complexity and additional benefits of more advanced models such as LSTM and Linear Regression. Comparing these models with the Naïve Baseline shows the sophisticated models' improved ability to comprehend and represent the intricate patterns and connections in unpredictable financial markets, such as cryptocurrency. We will discuss this more later in this report.

4.2.3 Testing LSTM Architecture

We performed targeted testing on our LSTM models to improve their structure and improve their accuracy in predicting Bitcoin price fluctuations:

- In our experiments, we modified the number of LSTM layers (ranging from one to four) and neurons per layer (ranging from 50 to 400) to determine the ideal model complexity.
- The effectiveness of bidirectional LSTMs, which evaluate information from preceding and subsequent contexts, compared to regular unidirectional LSTMs.

- Many dropout rates ranging from 0.1 to 0.5 were examined to prevent overfitting and find a rate that balances complexity and generalisation.
- Several optimisers (Adam, RMSprop, SGD) and learning rates (ranging from 0.001 to 0.1) were assessed to improve the training process and ensure effective convergence.
- We adjusted the batch sizes and the number of training epochs to improve training efficiency and the model's performance.

This systematic testing of LSTM configurations helped refine the models' structure to capture Bitcoin fluctuations.

4.2.4 Indicator LSTM Model Implementation

The Indicator LSTM model was engineered to use historical price data and technical indicators like RSI, MACD and moving averages. These indicators were pre-processed to be used as additional data features, boosting the model's ability to capture market dynamics accurately. This preprocessing involves normalising these indicators to be on the same scale as the price data, enabling the neural network to learn more effectively (Madani et al., 2018).

The Indicator LSTM model included many LSTM layers placed simultaneously. Each layer was built to extract information and gain insight from the time series data's complicated patterns. We employed the Keras library, an advanced neural network API built on TensorFlow, allowing us to quickly experiment with various network structures. The model was constructed with the following guidelines:

- The input layer receives pre-processed data, including technical indications and price data.
- Multiple LSTM layers were used, each containing a specific number of neurons (units). Thanks to the LSTM gating processes, these layers were essential for learning the data sequences and remembering crucial details while discarding non-essential aspects.
- Dropout layers were added between LSTM layers to prevent overfitting. Typically set between 20% and 50%, these layers randomly eliminated a proportion of neurons during the training phase, encouraging the formation of a more generalised model.
- The output layer uses linear activation for regression tasks like price forecasting.

The model used the Adam optimiser, a common choice because its flexible learning rate capacity results in more rapid and successful convergence. It changes the learning rate throughout training, which is extremely helpful given the different scales and behaviours in financial time series data.

4.2.5 LSTM Normal Model Implementation

The LSTM Normal model was simplified to focus solely on raw price data and test the LSTM's ability to forecast future prices using only historical price sequences. This model bypassed the complex nature of technical metrics to measure the raw predictive potential of LSTM networks on naked time series data.

This strategy was designed to help us determine which model would be more successful and produce higher accuracy. We hypothesised that including technical indicators as characteristics would degrade performance and reduce accuracy, potentially resulting in a less profitable trading bot.

The LSTM Normal model's implementation methods were straightforward but powerful:

- The model used cleaned and normalised pricing data as its input.
- Adding bidirectional LSTM layers boosted the model's learning capacity. These layers help the LSTM to manage data in both forward and reverse directions, resulting in a broader understanding of the temporal links between time series (Saulter, 2000).
- Similar to the Indicator LSTM, this model has numerous layers, each with a particular quantity of neurons based on the complexity of the input being processed.
- The model was trained using historical price data using the Keras framework. Backpropagation through time (BPTT) was used to achieve successful gradient descent.

This method led the LSTM Normal model to capitalise on LSTM's fundamental strengths in sequence prediction, focusing on just price movements without the effect of external factors.

Both models were evaluated and tuned to ensure the task's layouts and parameters were optimal. The methodological design for each model was chosen to closely match the project's targets: accurately predicting Bitcoin values under volatile market circumstances and proving the flexibility and strength of LSTM networks in financial forecasting.

4.2.6 Linear Regression Model

The linear regression model serves as the fundamental comparative baseline for our predictive models. It is built with Python's sci-kit-learn module, which provides a simple method for modelling financial time series by creating a linear connection between previous prices and future values. This model's simplicity allows clear benchmarking to more advanced models, emphasising the extra insights acquired from LSTM's treatment of non-linear patterns.

4.2.7 Naïve Baseline Model

The Naïve Baseline model, easily implemented by projecting the previously observed price as the following period's projection, serves as the primary performance criteria. It creates the simplest forecasting system with the least amount of code—essentially one line of Python. This model is critical for proving the baseline accuracy that all advanced prediction models should exceed.

4.2.8 Implications for AI Model Selection

The construction and comparison of these models showed the need to pick suitable AI modelling techniques according to the market's characteristics and data type. This complete model-building technique allows for a deeper examination of the advantages and disadvantages of different AI models and a thoughtful choice of models for deploying an effective Bitcoin trading bot. The implementation and results of each model directly impact strategic decisions in producing trading algorithms, guaranteeing the chosen model matches up with the trading strategy's particular attempts and market circumstances.

4.3 Backtesting Methodology

Backtesting was a critical element of our research, acting as an exhaustive way to assess our AI models' forecasting ability under simulated situations utilising historical data. This included carefully simulating trading operations based on AI model predictions to analyse future research profitability and the efficacy of the different trading methods (Taylor, 2023).

4.3.1 Data Preparation for Backtesting

The backtesting used high-resolution Bitcoin price data, especially from Yahoo Finance, covering the period February 11, 2024, to April 10, 2024. This data includes minute-by-minute price information—open, high, low, close, adjusted close and volume—which was essential in comprehending market dynamics throughout the backtesting phase.

The data was formatted and normalised using Python's Pandas library and the MinMaxScaler from sklearn.preprocessing to ensure it was in the best possible shape for model input (Smith and Zhao, 2024).

The LSTM and Linear Regression models were developed to estimate Bitcoin values based on their ability to handle sequence data and linear trends.

Each model's implementation required configurations:

- Developed an LSTM model using Keras and TensorFlow for sequence-based prediction. This approach was created to handle nonlinear relationships by learning from prior pricing. Section 4.2: Model Development provides further information on the LSTM model design.

- The Linear Regression Model, implemented with Scikit-learn, provides a baseline for comparing performance by concentrating on linear trends in price data without sophisticated transformations like LSTM. Additional information on model setup is found in Section 4.2: Model Development.

4.3.2 Simulation of Trading Strategy

Backtesting consisted of imitating trading choices based on the model's predictions. The simulation software was designed to execute trades based on specific criteria automatically: if the predicted price of Bitcoin was more significant than the actual price, a buy order took place; if it was lower, a sell order was done. This method allowed us to model a trading strategy in which buy and sell decisions were made dynamically based on expected future prices (Williams, 2023).

To track the trading strategy's effectiveness, we kept records of all transactions, including the sort of transaction (buy or sell), the quantity of Bitcoin moved, the transaction price and the timestamp. This accurate recording was critical for evaluating the strategy's performance throughout the backtesting era.

4.3.3 Evaluation of Trading Outcomes

Throughout the backtesting phase, both models were assessed based on their profitability. A complete transaction log documented each trade's characteristics, allowing an in-depth investigation of when and why each model succeeded or failed. This investigation was critical for determining each model's practical use and efficacy in real-world trading circumstances.

The final evaluation compared the beginning and final balances to determine how effective the models were in improving the trade balance throughout the test period. In addition, significant performance indicators such as the total number of transactions, the percentage of profitable trades and the overall return on investment were measured to offer a complete picture of each model's performance (Anderson, 2023).

This demanding backtesting technique detailed how each model fared regarding forecast accuracy and possible financial returns. It led us to modify the trading algorithms and strategies based on each model's unique behaviours and capabilities, ensuring that the chosen strategies were consistent with the observed market circumstances and model strengths.

4.4 Live Trading the LSTM Normal and Linear Regression Bots

Using AI models in real trading scenarios is the final test of their practical value and efficacy. Our project used the Binance API to integrate and run two different models, the LSTM Normal and Linear Regression, in real-time cryptocurrency trading scenarios. This section describes the live trading methods used for both models.

4.4.1 Implementation and Configuration

During the live trading phase of our project, the LSTM Normal and Linear Regression models were used for trading Bitcoin in real-time on the Binance platform. This method permits us to directly compare the performance of a complicated, sequence-based model to a simpler, trend-based model under the same market conditions.

- Both trading strategies relied heavily on real-time data collecting for optimal performance. Using the Binance API, we continuously obtained live Bitcoin price data, concentrating on the 'BTCUSDT' trading pair. The data comprised high-resolution price points (open, high, low, and close), essential to making quick and accurate predictions during live trading (WunderTrading, n.d.).
- To guarantee that the input data was appropriately structured and scaled, a consistent preprocessing method was applied to the data in both models. We processed the incoming live data to fit each model's training data specifications, using Python's Pandas module for data structure and the StandardScaler from sklearn.preprocessing for normalisation. This phase was essential to guaranteeing the model predictions' validity and precision in a real context.
- Both models were built using Python-based libraries, with the LSTM Normal model using Keras and TensorFlow for deep learning and the Linear Regression model using Scikit-learn for predictive analysis. After cleaning the real-time data, it was input into the appropriately trained models to offer price forecasts. These projections provided the foundation for our trading judgements.

4.4.2 Trading Strategy and Execution

Our live trading technique was based on automatic decision-making based on the models' price projections compared to current market values.

- **Risk Management and Order Execution**—We used stop-loss and take-profit limits to safeguard resources during market volatility. Trade execution was performed by the Binance API, with each transaction recorded for future performance studies. The API's rate limitations were carefully regulated using a custom rate_limiter function to guarantee compliance with Binance's use guidelines and prevent service disruptions.

- **Trading Decisions**—Both models used elementary but efficient trading logic: if the anticipated price in the future was higher than the current price, a buy order was placed; if the anticipated price was lower, a sell order was executed. This method attempted to take advantage of the price swings predicted by the models.
- **Performance Monitoring**—Live trading performance was continuously monitored through a detailed ledger that recorded every transaction, including the type of trade (buy/sell), the quantity of Bitcoin traded, transaction prices, and timestamps. This detailed record-keeping helped assess the trading strategy's success and the models' predicted accuracy.

By executing and comparing these two unique models in a real trading environment, we examined their respective strengths and problems, resulting in a rich dataset for future optimisation and the creation of more advanced trading strategies.

5. Results

5.1 Exploratory Data Analysis (EDA) Results

Exploratory Data Analysis (EDA) examines the details of Bitcoin's price movements using high-frequency data from the Binance API. This section presents the investigation's primary findings, providing the groundwork for the subsequent advanced predictive models.

5.1.1 Data Visualisation and Descriptive Statistics

First, we wanted to create a time-series chart showing the movement of Bitcoin's price. Descriptive statistics defined the extreme volatility we noticed, a natural feature of the bitcoin market. Figure 2 shows the price ranged between $64,867.99 and $72,719.99 over the study period, hinting at the market's volatility based on these statistics. We are comfortable with the reliability of our LSTM model because of its excellent performance in capturing and predicting this volatility.

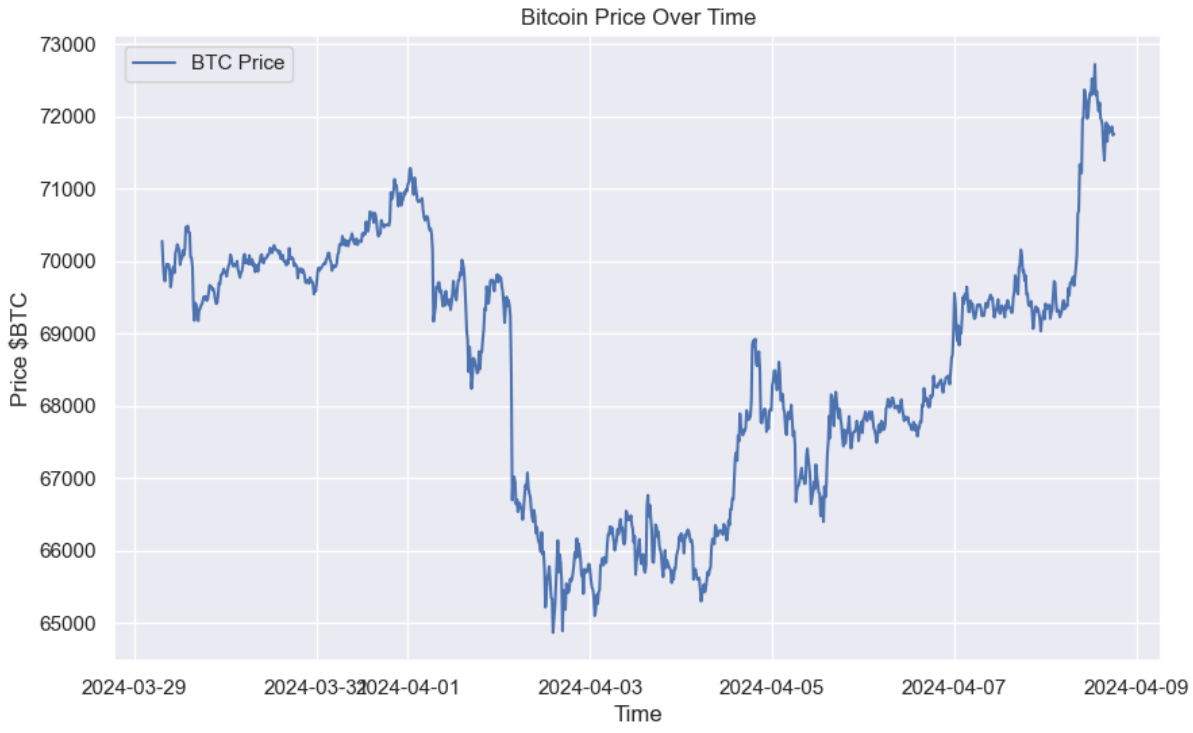


Figure 2: Bitcoin price over time (a few days), showing how quickly the price varies over a short period.

5.1.2 Analysis of Moving Averages (MA), Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) Plots

The strategic application of Moving Averages (MA), Relative Strength Indexes (RSI), and Moving Average Convergence Divergence (MACD) during the Exploratory Data Analysis

(EDA) seriously improved our understanding of the Bitcoin market's behaviour and trends. This feature highlights the significance of these indicators in our research. We also heavily inspected them as planned to incorporate them into our LSTM Model.

Moving Averages (MA)

By removing daily price fluctuations, the Moving Averages plots of Bitcoin prices revealed more trends over the short- and long-terms (7 and 30 days). For predicting likely market reversals, the points of intersection of the short- and long-term Moving Averages (MAs) proved helpful. As an example, a positive trend and possible buying chance was seen when the short-term moving average crossed above the long-term moving average. A negative trend was observed when the short-term MA dropped below the long-term MA, indicating that traders should sell or take caution. It may be possible to time market entry and exits using these crossover points.

Figure 3 shows the MA plots volatility and trends of the Bitcoin market. The crossover points provided a foundation for the feature set of our predictive model by behaving as signals of possible swings in market momentum. Our LSTM model is predicted to improve the accuracy of these indicators with its excellent performance in predicting Bitcoin values.

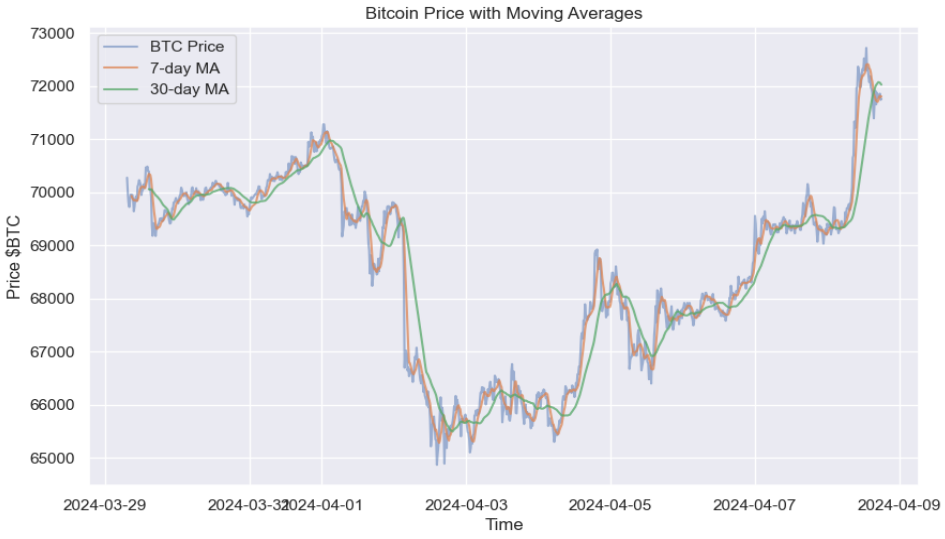


Figure 3: Bitcoin price using Moving Averages which show the market's direction, are trend-following indicators.

Relative Strength Index (RSI)

The RSI plots gave a graphical portrait of the Bitcoin market's momentum, distinguishing overbought and oversold conditions. An RSI above 70 specified that Bitcoin was likely overbought, hinting that there would be a future price decrease. In contrast, an RSI below 30 indicates that Bitcoin was possibly oversold, meaning a future price increase may occur.

The market seemed open to sudden shifts in momentum and several overbought and oversold

events during the examination (as seen in Figure 4) based on the distribution of RSI values. This volatility describes how open the Bitcoin market is to outside factors and investors' mindsets. Likewise, the MACD distribution, which displays the frequency of bullish and bearish crossovers, gave key information about the market's momentum and trend shifts.

Moving Average Convergence Divergence (MACD)

The MACD plots comprised the MACD line and the signal line, which helped determine the momentum and direction of the market trend. The crossover of these lines indicated market confidence or bearishness. Crossovers above and below the signal line indicated bullish and bearish trends.

The MACD analysis presented a view of market dynamics, with crossover points indicating important events when market momentum altered. This indicator, which emphasises trend strength and direction, improves our model's predictive skills.

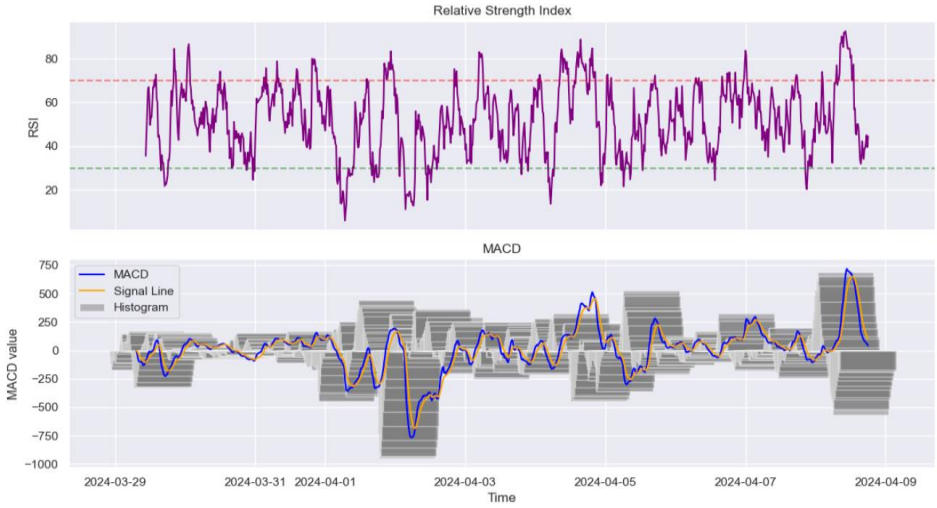


Figure 4: RSI and MACD analysis displaying their capacity to foresee market trends and momentum changes.

5.1.3 Distribution of RSI and MACD Values

Figure 5 shows RSI and MACD distribution studies showing market volatility over time. The RSI distribution shows a shift between overbought and oversold periods, representing market price movement in general and market reactions to external events. Bullish and bearish crossovers in the MACD distribution reveal market momentum and trend shifts.

Analysing RSI and MACD readings was vital as it gave a complete picture of market circumstances, showing us their importance in identifying the market. Seeing the large distribution of values gave us confidence in choosing these indicators for our LSTM model, as the large values of distribution meant that these indicators would be able to capture market

shifts. These distributions also showed the Bitcoin market's complexity and reactivity to external influences.

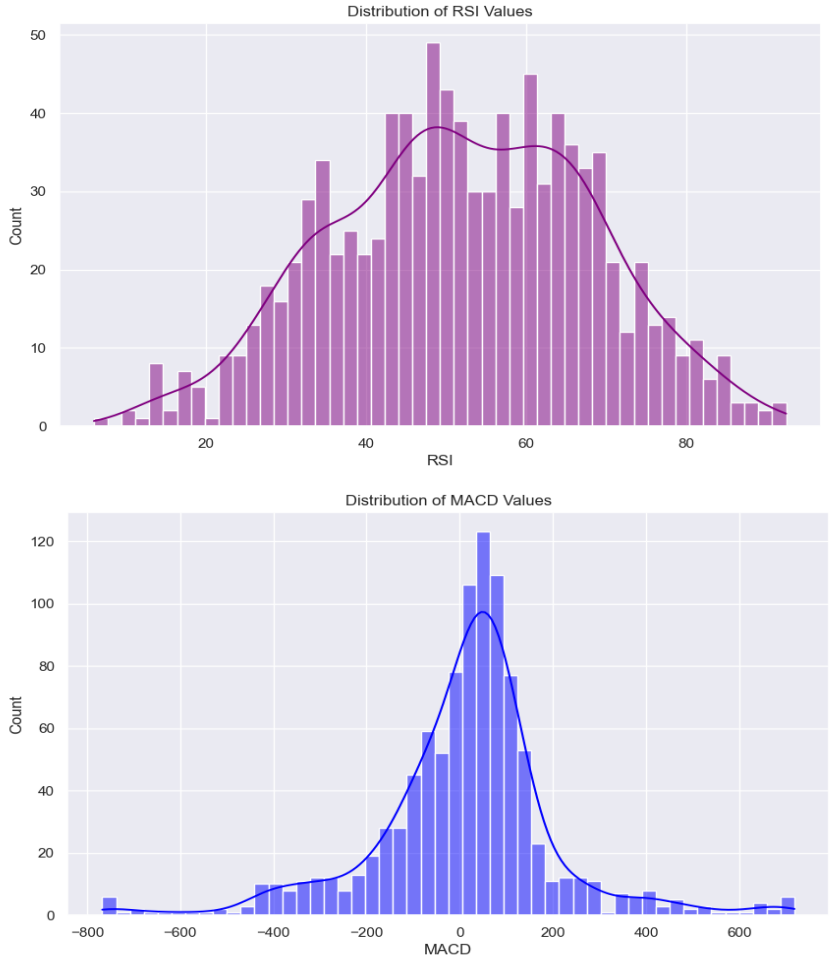


Figure 5: The distribution of MACD and RSI prices.

5.1.4 Feature Correlation Analysis Results

Feature Correlation Analysis is a big part of our Exploratory Data Analysis. It aims to thoroughly understand the relationships between key market indicators and their impact on Bitcoin price changes. A correlation matrix tells our dataset's linear relationship between pairs of attributes (Dimingo, 2019).

- Short-term and long-term Moving Averages (MAs) positively connect with Bitcoin's future price, with coefficients of 0.99 for MA_5 and 0.98 for MA_10. This means that they are reliable in predicting price patterns.
- The Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) showed a minor relationship with Bitcoin's future price, at 0.22 and 0.37, respectively. These indicators capture market momentum and trend reversals.

- RSI and MACD have a high correlation (0.65) and may provide similar information about market circumstances.

The correlation analysis results seen in Figure 6 directly impacted our model development process, notably feature selection. The strong connection between MAs and Bitcoin prices demonstrated the significance of these indicators in capturing the underlying market movement. As a result, MAs were prioritised in the feature set for our prediction model.

Furthermore, RSI and MACD's moderate correlation with price and low intercorrelation with other aspects backed their inclusion as complementing indicators. They give depth to the model's input by including momentum and trend reversal signals, which are critical for reflecting the Bitcoin market's multidimensional structure. While useful, these indicators do not determine Bitcoin's price. External events and market sentiment also matter.

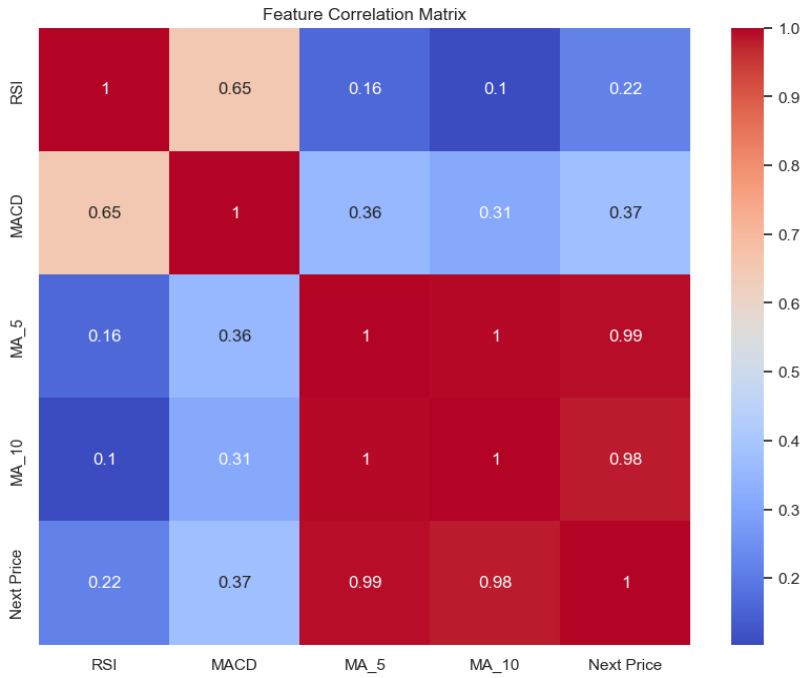


Figure 6: Feature Correlation Matrix. The matrix displays several relationships, showing that different indicators can influence Bitcoin's future price.

5.1.5 Volatility Analysis

The Volatility Analysis of Bitcoin price swings, seen in Figure 7, was also an element of our Exploratory Data Analysis. It showed the cryptocurrency market's unpredictability and the need for a predictive model to manage such volatility. The research was carried out by calculating the standard deviation of Bitcoin price changes over a given period, visually evaluating price swings and market volatility.

- The volatility study showed many fluctuations in Bitcoin prices during the analysis period. The finding corresponds with the cryptocurrency market's well-documented volatility, defined by rapid price swings over short periods.
- We ended up finding periods with far greater volatility than average. Further reading informed us that these periods usually occurred with significant market events or reports influencing market conditions. The result pressures the relevance of external factors on market volatility and the importance of including such dynamics in prediction models.

This investigation informs us of the necessity for a solid predictive modelling method to adapt to rapid market changes. Hence, LSTM networks, which can learn from data sequences and adapt to market situations, are important because models that ignore such volatility will generate inaccurate forecasts.

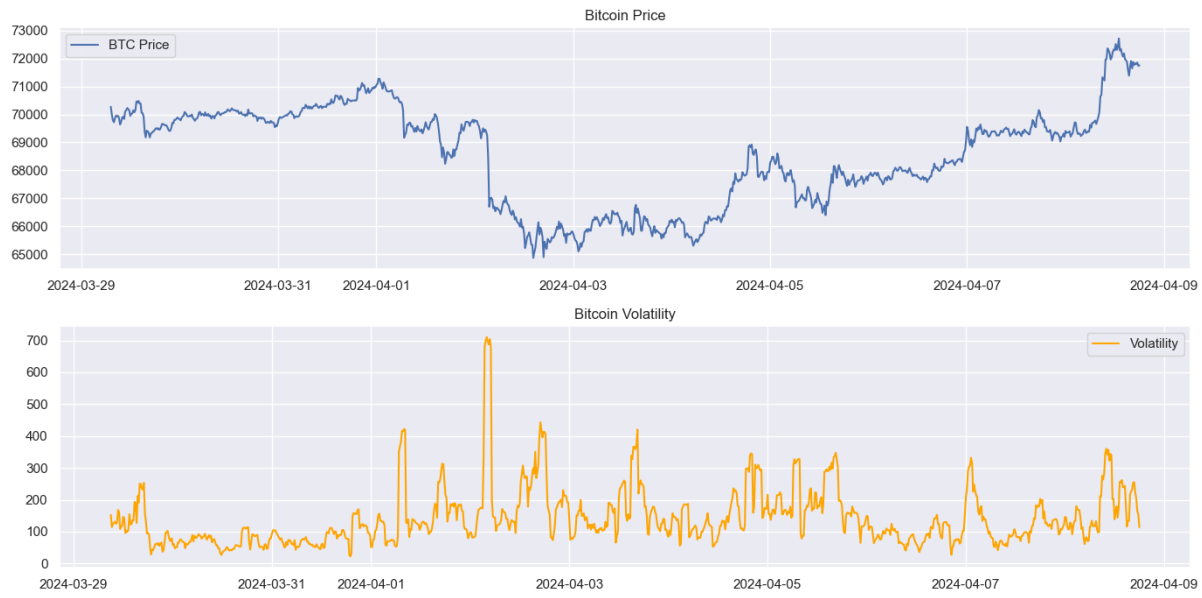


Figure 7: Bitcoin volatility, indicating uncertainty.

5.1.6 Trend Analysis with Decomposition

Trend Component—The trend graph reveals relatively stable growth in Bitcoin values, indicating a bullish market throughout the specified timeframe (this will be discussed further in the Linear Regressions Evaluation). The trend line from roughly 65,000 to just below 70,000 indicates broad market optimism or continuous purchasing demand over this period.

Seasonal Component—While not apparent because of the short period and high-frequency data examined, the seasonality factor reveals subtle cyclical patterns. These may reflect intra-day trading patterns or micro-trends that are not immediately obvious in price change.

Residual component—After accounting for trend and seasonality, the residuals show noise or unexplained variance between -1000 and 1000. This suggests that although the LSTM model can capture most of the data's structure, there is still some volatility and unpredictability in the Bitcoin market (Wang et al., 2023).

As seen in Figure 8, these results support using an LSTM model to predict Bitcoin values. A clear trend component indicates that there is somewhat of a directional movement that a well-trained LSTM may be able to foresee. A degree of seasonality, even slight, and the residual noise level highlight the price series' complexity, which the LSTM model's architecture is intended to capture and learn from. Capable of detecting complicated patterns in time series data, which is exactly what the LSTM model is intended to achieve.

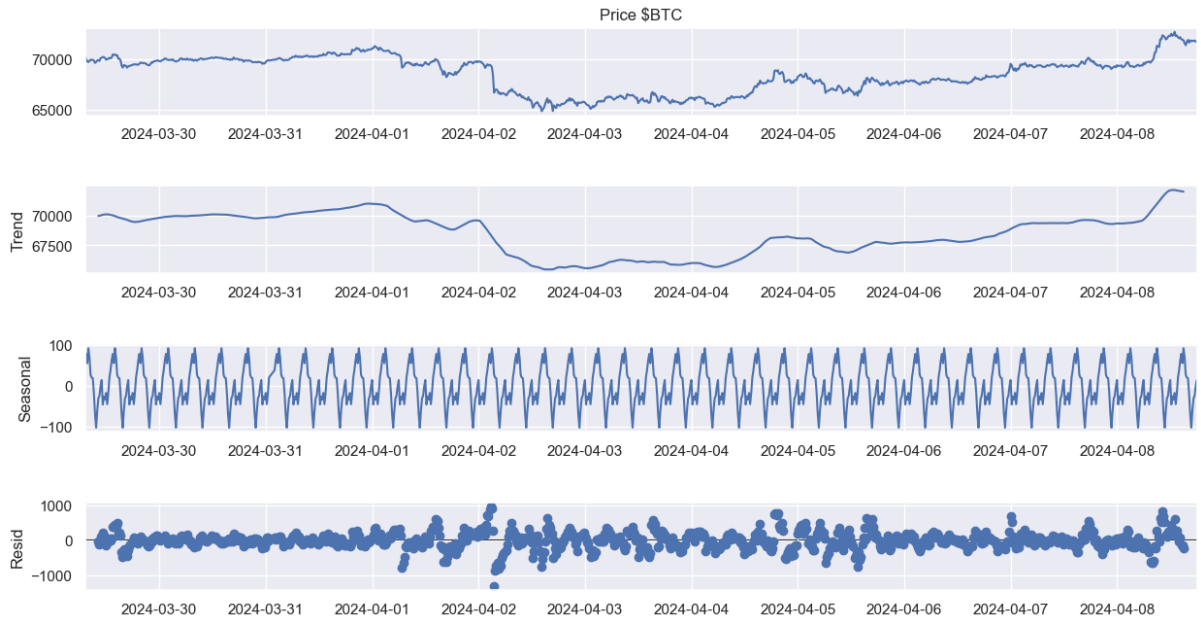


Figure 8: Decomposition of the Bitcoin Price Series. This graph shows the different parts of the Bitcoin price series.

Trend analysis and deconstruction have lit Bitcoin market behaviour and justified the adoption of advanced forecasting models like LSTM. A rational solution to predicting volatile financial markets captures macro-level trends and subtle, frequently ignored seasonal fluctuations.

5.1.7 Augmented Dickey-Fuller Test

The Augmented Dickey-Fuller (ADF) test results in Figure 9 showed the Bitcoin price series' non-stationary character. With an ADF statistic of -1.127830 and a p-value of 0.703847, the test suggested the presence of a unit root, which explains the series' non-stationarity. The consequences of these findings are major for our model selection and preprocessing stages. Many average prediction models are challenged by non-stationary data, which needs a series

with a constant mean and variance over time. The LSTM model can handle such data without differencing or stationarity transformation, making it suitable for this investigation.

```
ADF Statistic: -1.127830
p-value: 0.703847
Critical Values:
        1%: -3.437
        5%: -2.864
        10%: -2.568
The time series is not stationary.
```

Figure 9: ADF Test Results.

5.1.8 Memory Effect Analysis

The autocorrelation plot below in Figure 10 shows how past Bitcoin prices affect future prices. The gradual decrease in the autocorrelation function over lag periods indicates that it is still meaningful even though the impact of previous prices fades over time. This, again, supports the use of an LSTM model, whose structure is built to capture and learn from temporal relationships. Models lacking memory systems cannot achieve this capability.

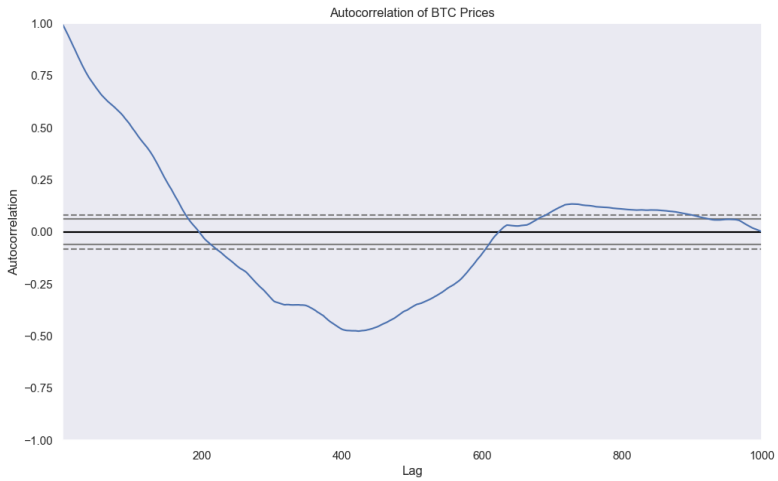


Figure 10: The autocorrelation of BTC prices. Due to the memory effect, bitcoin prices are correlated with previous and future prices.

5.1.9 Conclusion

The exploratory data study has organised the basis for building our models specifically for using the LSTM model to estimate Bitcoin values. The trend analysis and time series decomposition show the market's complex character, implying patterns best embodied by models able to manage sequential and time-dependent data. The Augmented Dickey-Fuller test result proved the series' non-stationary environment, which is an aspect of shaping our modelling strategy. Lastly, the memory effect analysis demonstrated the market's temporal dependencies, validating the LSTM model's relevance to our objectives.

These findings strongly support using LSTM models to anticipate financial time series data in explosive cryptocurrency markets. Careful preparation and evaluation set the foundation for predictive modelling, and our findings strengthened our approach and methodology.

5.1.10 EDA on the Daily frame

We analysed daily market movements based on the EDA on 15-minute Bitcoin data and compared them to our high-frequency findings.

Data Visualisation and Descriptive Statistics for Daily Data

Daily descriptive statistics in Figure 11 show Bitcoin's long-term trends by showing market fluctuations. Compared to the 15-minute analysis, the daily price analysis showed longer-term price variations.

```
count     1000.00000
mean     34803.55919
std      13588.01148
min      15781.29000
25%      23545.56750
50%      30287.07500
75%      43334.23750
max      73072.41000
Name: Price $BTC, dtype: float64
```

Figure 11: Daily timeframe statistics.

The daily timeframe EDA gave us a macro view of the market's direction over time, while the 15-minute EDA captured its quick reactions to developments and intraday trends. The daily data analysis gives a clearer picture of the overall momentum and more significant trend reversals that were not obvious in the short-term data.

Trend and Volatility Analysis on Daily Data

On the daily chart, Bitcoin's price movements revealed a less unpredictable but more pronounced trend direction over time, as shown in Figure 12.

Figure 12: Daily timeframe Volatility analysis.

Technical Indicators Analysis on Daily Data

The daily technical indicators in Figure 13 indicate longer overbought and oversold phases and greater MACD trend lagging, providing insights for long-term investment strategies. This shows the number of reversals in the market.

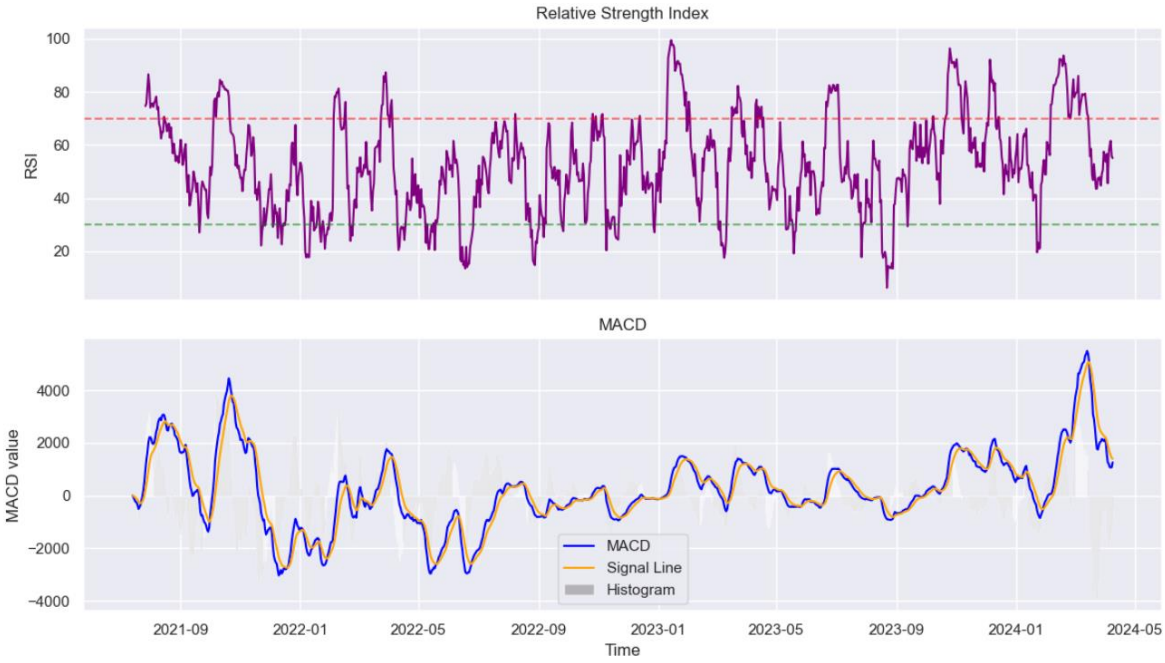


Figure 13: Daily timeframe MACD and RSI Plots.

r

Correlation Analysis

A comparison of daily (Figure 14) and 15-minute correlation matrices shows that daily data has a lower correlation coefficient between RSI, MACD and Bitcoin prices. This refers to the less immediate but more significant movements observed in daily statistics.

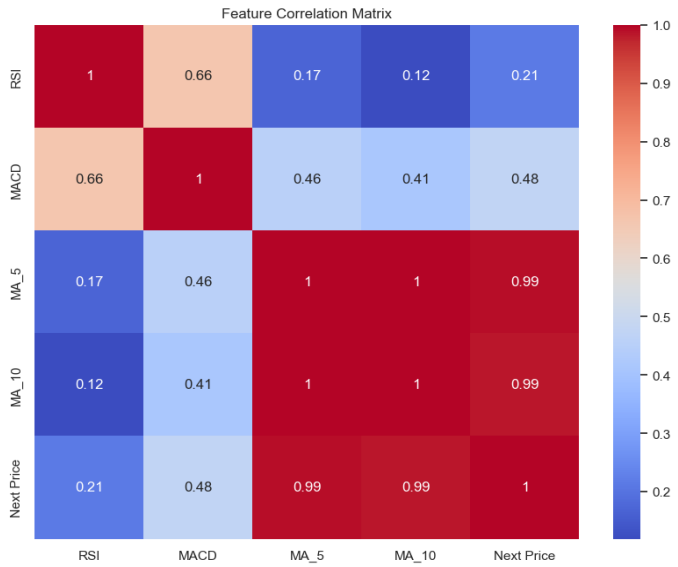


Figure14: Daily Feature Correlation Matrix

Compared to the 15-minute interval EDA, this daily study shows how predictive models must adapt to different market situations and timeframes. It also shows our LSTM model's capacity to learn and anticipate varied temporal patterns and market behaviours.

5.1.11 Justification for Choosing the 15-Minute Timeframe

After reviewing the EDA findings over a few durations, we intentionally created our LSTM model at 15-minute intervals. The 15-minute timeframe appeared as the sweet spot, with more data points needed to portray the finer features of Bitcoin price fluctuations, often caused by quick market movements. Thus, an LSTM model is suitable as its main strength is learning from the finer, more complex details of data sequences. This allows for a better understanding of short-term market movements and lets the model respond quickly to unexpected developments.

Moreover, the 15-minute EDA showcased more frequent trading opportunities—this aligns with our project's objective of creating a responsive trading bot that can exploit quick changes. The higher data density within these shorter periods grants the LSTM a heavy environment for spotting patterns and making predictions with a level of detail and precision that longer timeframes may not reveal.

The volatility analysis within the 15-minute EDA shows the market's fast reactions, creating a more water-like setting for the LSTM. In contrast, while the daily period offered helpful insights into longer-term trends, it wasn't steady with our goal of developing a system capable of active trading and real-time decision-making.

The EDA results suggested that our LSTM model would perform best in the 15-minute timeframe, suggesting a more refined and practical trading approach for the volatile and unpredictable cryptocurrency market.

5.2 LSTM Models, Linear Regression, Naïve Baseline Development

This study was based on the belief that a model's performance should be theoretically solid and experimentally supported. To that target, we started building and analysing multiple models, each fixed by theories and assumptions designed to navigate the dangers of Bitcoin price prediction. This section reveals the harsh facts of these models' performances, illumining the junction of theory and practice.

5.2.1 LSTM with Technical Indicators Model Performance

The Indicator LSTM model used a bidirectional LSTM architecture to combine technical indicators such as the Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD) and Moving Averages (MA) alongside historical price data. We thought the confluence of data would allow the LSTM to show the deeper trends in the price sequences (Li et al., 2017).

However, the training and validation loss graphs in Figure 15 showed a sluggish descent, revealing the model's strain in effectively adjusting its weights. The final test loss registered at an unbelievable 4,514,340,352.00, accompanied by a Test Score (RMSE) of 126,371,042.80 and an R^2 of -25,777,351.48%. As seen in Figure 16, these numbers were disappointing; they suggest a model handling volatile data and a pattern too obscure for its architecture to capture.

Figure 15: Training and Validation Loss of Indicator LSTM.

```
2/2 [==============================] - 0s 14ms/step - loss: 4514340352.0000
Test Loss: 4514340352.0
2/2 [==============================] - 2s 7ms/step
Test Score (RMSE): 126371042.80
R-squared: -25777351.48%
```

Figure 16: Indicators LSTM Model Results.

The reasons for this performance disadvantage are many:

- The normalisation technique may not be appropriate for financial data with non-standard distributions and outliers.
- The LSTM model's complexity, with its bidirectional layers and regularisers, might have led to overfitting.
- The optimiser's learning rate and settings might have been misaligned with the data's needs, resulting in inadequate convergence.
- The lagging nature of technical indicators (RSI, MACD and MA) may have delayed capturing new trends in the unpredictable cryptocurrency market. Dependence excessively on historical trends risks the model's ability to respond to unexpected market changes.

5.2.2 Linear Regression Model Performance

The Linear Regression model, meant to detect linear correlations in Bitcoin price data, reached a high R^2 of 96.40%, suggesting that it could explain a considerable percentage of the variance in the test dataset. This was supported by a low Test Score (RMSE) of 0.08, which reflected the model's accurate predictions compared to the actual prices in the test set. These results can be seen in Figure 17.

```
Test Score (RMSE): 0.08
R-squared: 96.40%
```

Figure 17: Linear Regression Results.

Analysis of Predictive Bias

These Linear Regression results were suspicious, so a comprehensive review of the data and projections showed a bias in the model's performance. Bitcoin prices rose overall during the study period, most likely resulting in the Linear Regression model leaning towards a bullish trend. This is demonstrated by the RMSE of 0.08, which, while low, does not account for the high chances of price reversals or market downturns occurring beyond the purview of the historical data utilised in training. This can be seen in Figure 18 below.

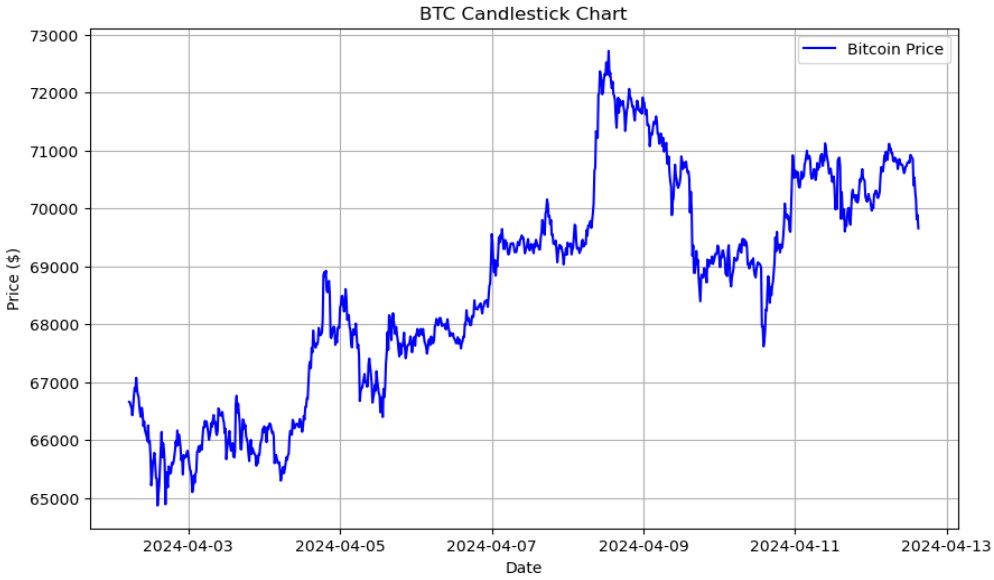


Figure 18: Bitcoin Plot of all data used to make all models.

The residuals plot for the Linear regression in Figure 19 pointed out differences between expected and real prices when the residuals are supposed to be randomly distributed with no visible pattern. The appearance of any pattern or structure in the residuals plot may suggest that the model is not capturing all the influencing variables or that there are trends that the model does not account for.

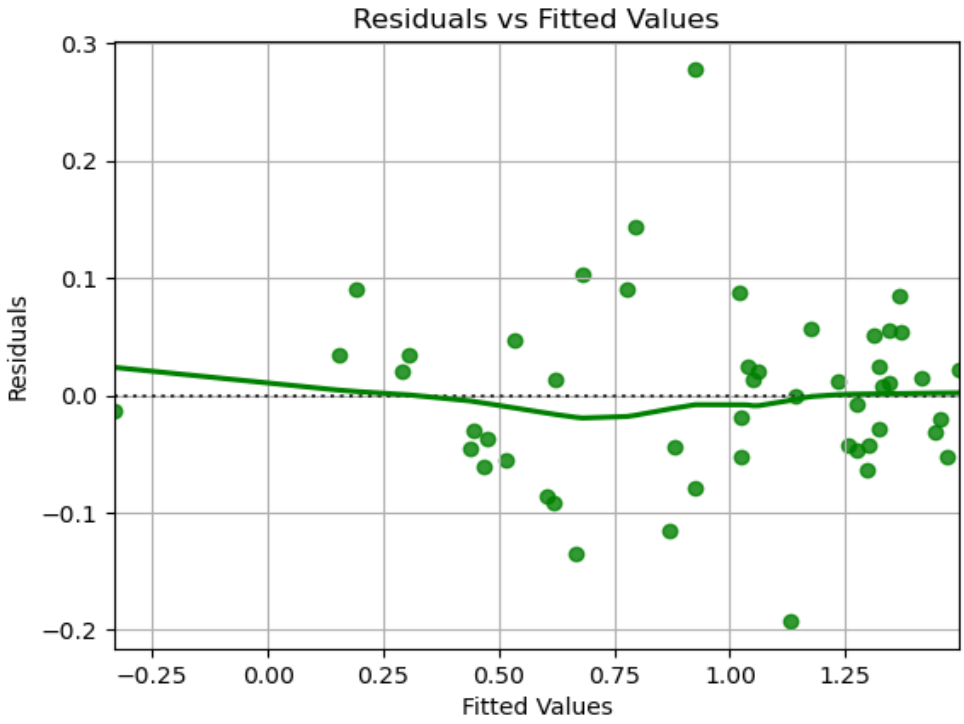


Figure 19: Residual Plot for Linear Regression.

Additional Evaluation Metrics

Additional evaluation measures provided a more detailed insight into the model's performance. The Mean Absolute Error (MAE) of 0.06 and the Mean Absolute Percentage Error (MAPE) of 8.13% were positive, suggesting that the model could create predictions that were, on average, close to actual values. The Explained Variance Score of 0.96 indicated that the model effectively explained the variance in the data (Nguyen and Tran, 2023). These results can be seen in Figure 20 below.

```
Additional Evaluation Metrics for Linear Regression Model:
Linear Regression Mean Absolute Error (MAE): 0.06
Linear Regression Mean Absolute Percentage Error (MAPE): 8.13%
Linear Regression Explained Variance Score: 0.96
```

Figure 20: Linear Regression Metrics

Linear regression residual analysis revealed a slightly more pronounced pattern than the Normal LSTM model (which we will discuss below), showing possible general weaknesses in prediction. Although not 100% inconclusive, such residual patterns reflect model biases or a failure to represent complicated, nonlinear relationships in the data completely.

Furthermore, the major role of the coefficient for the fourth feature, as seen in Figure 21, creates worries about over-reliance on this data component. Such a large weighting may result in a model that is excessively tuned to the patterns associated with this feature, perhaps at the expense of other important signals in the dataset. This dependence may artificially inflate the model's performance measures, particularly the R^2 value of 96.40% (Olsen et al., 2023).

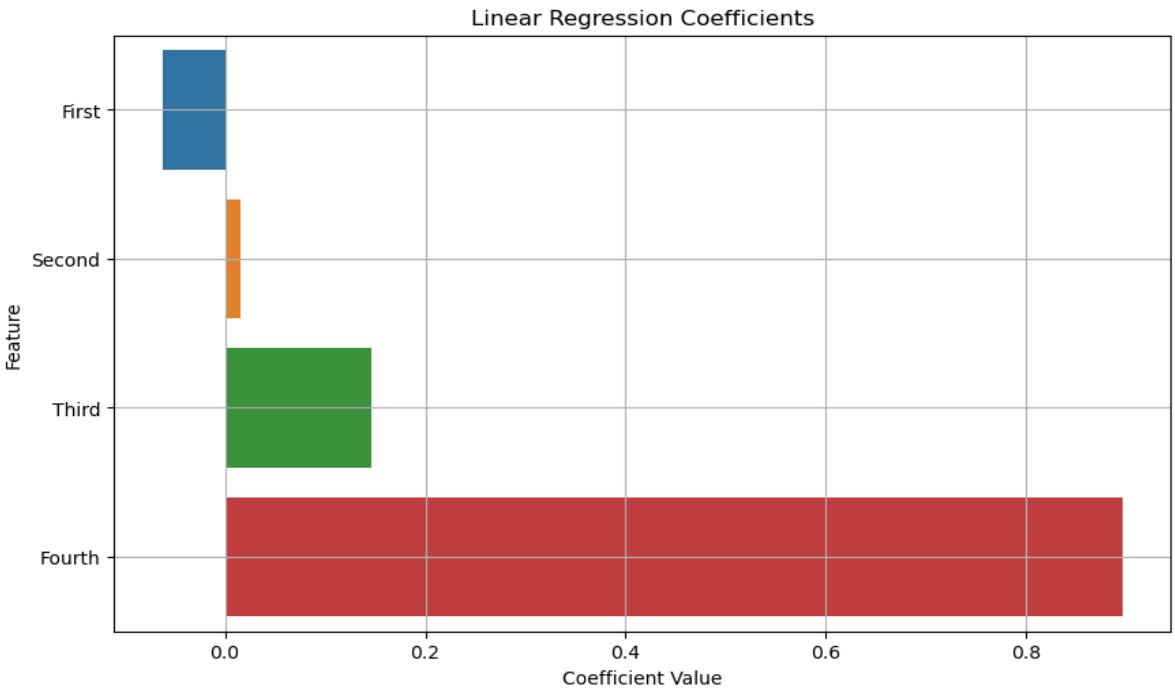


Figure 21: Linear Regression Coefficients.

While the Linear Regression model demonstrated great accuracy on the provided dataset, the risk of bias caused by unidirectional price movement during the training phase poses concerns about its robustness and adaptability. The model showed us the significance of including various market conditions in the training data to create an overall model that can resist and correctly anticipate a multidimensional trading environment such as cryptocurrency.

5.2.3 Naïve Baseline Model

The Naïve Baseline model was the most basic form of prediction, serving as a benchmark for more advanced LSTM and Linear Regression models. Using the last observed value to predict the next, the Naïve Baseline's performance resulted in an RMSE of 0.24 and an R^2 of 68.66%, indicating moderate predictive accuracy. The additional measures, Mean Absolute Error (MAE) at 0.17 and Mean Absolute Percentage Error (MAPE) at 27.25%, along with an Explained Variance Score of 0.69, indicated the model's effectiveness in capturing the general trend despite its simplicity. Results can be seen in Figure 22.

```
Additional Evaluation Metrics for Naive Baseline Model:
Naive Baseline Mean Absolute Error (MAE): 0.17
Naive Baseline Mean Absolute Percentage Error (MAPE): 27.25%
Naive Baseline Explained Variance Score: 0.69
```

Figure 22: Naïve Baseline Results.

5.2.4 Comparative Analysis and Strategic Refinement

Strategic improvements were required when comparing the Naïve Baseline, LSTM with Indicators and predicted Linear Regression models. The Naïve Baseline model's limited performance demonstrated the necessity of simplicity and capturing the overall market trend. The LSTM indicators model, on the other hand, performed poorly, suggesting the need to rethink feature selection, model complexity, and data normalisation methods.

The LSTM model's overfitting challenges and limited adaptability to new trends indicated that the model should be simplified.

When developing these models for real-world applications, great thought was given to balancing model complexity and forecast accuracy. The goal remained to improve the models' capacity to estimate Bitcoin prices accurately, adapt to the volatile nature of the cryptocurrency market, and provide solid advice for trading tactics.

5.2.5 LSTM Normal Model Performance

Shifting our focus, we created the LSTM Normal model, which pulls the pure predictive power of raw price data without the complexities or lag of technical indicators. This model used a bidirectional LSTM architecture and was trained solely on price movements, removing the indicators that added noise and complexity to the previous model.

The architecture of the LSTM model for predicting Bitcoin evolved to handle the hardships of financial time series data. The model uses bidirectional LSTM layers, which are necessary for collecting context data from the dataset related to the past and the future. This method works effectively for Bitcoin since it can significantly improve forecast accuracy by understanding patterns in both directions (Kumar and Sharma, 2023).

The model uses 128 nodes in the bidirectional layers of the structure. A detailed first look is made more accessible by the network's ability to extract a wide range of features from the input data due to its relatively large number of nodes. The number of nodes drops as the network grows, reaching 64 in the next LSTM layer and 32 in the last layer. This decrease helps simplify the model's output, focusing on the most crucial trends predicting future Bitcoin prices and improving the data to the most relevant parts.

To limit the risk of overfitting, the model's dropout layer rate was set to 0.3. This dropout rate is chosen to balance the model's need to learn complex patterns while preventing overfitting to the noise in the training data (Chen and Wang, 2023). During training, the model learns to build solid features that generalise well across multiple datasets rather than learning specific noise patterns by randomly eliminating 30% of the network's units.

Regularisation is another important model component, and it is implemented in the LSTM layers via L2 regularisation. L2 regularisation, also known as ridge regularisation, helps penalise the size of the coefficients, keeping them modest and simplifying the model. A regularisation factor of 0.001 is used. This adds just enough penalty to keep the weights from becoming too big, which might result in overfitting training data and poor generalisation to new data.

The Adam optimiser is included in the model because of its flexibility in dealing with non-stationary targets, which is common in financial time series. The learning rate is set to 0.0005, supplying an even strategy that allows the network to learn successfully while not bouncing too much around the ideal answers (Gupta and Zhao, 2023). This slower rate enables the model to fully explore the solution area and settle on a solid set of parameters that offer high predicted performance.

As seen in Figure 23, the model included over 227,745 trainable parameters. This shows a complex network capable of detecting complex patterns in the data.

```
Model Summary:
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional_2 (Bidirectio  (None, 4, 256)           133120
 nal)

 dropout_4 (Dropout)         (None, 4, 256)            0

 lstm_7 (LSTM)               (None, 4, 64)             82176

 dropout_5 (Dropout)         (None, 4, 64)             0

 lstm_8 (LSTM)               (None, 32)                12416

 dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 227,745
Trainable params: 227,745
Non-trainable params: 0
_________________________________________________________________
```

Figure 23: LSTM Models Architecture.

Note: Both LSTM Models were given the same architecture.

Figure 24 shows the training phase and the model's loss progressively decreased, revealing successful learning from the training data. The justification loss followed this pattern, showing the model's ability to conclude previously unknown data.

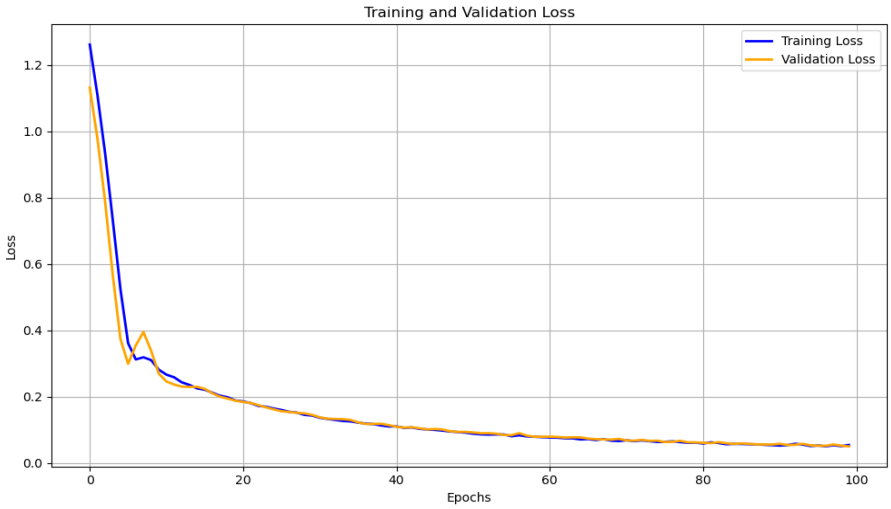


Figure 24: Training and Validation Loss for Normal LSTM.

The LSTM Normal model performed excellently with the test data, as seen in Figure 25. Its test loss was 0.0506, which suggested a well-fitted model. The Test Score (RMSE) was a respectable 229.10, showing the model's good accuracy in price prediction. It also had an outstanding R^2 value of 97.96%, showing the model's ability to explain the variability of the response data around its mean.

```
2/2 [==============================] - 0s 12ms/step - loss: 0.0506
Test Loss: 0.050597697496414185
2/2 [==============================] - 2s 6ms/step
Test Score (RMSE): 229.10
R-squared: 97.96%
```

Figure 25: LSTM Normal, test loss, RMSE and R^2.

Results in Figure 26 show an MAE of 168.75, which measured the average divergence from real prices and showed good prediction accuracy for the model. The Mean Absolute Percentage Error (MAPE) of 0.25% showed little changes from the genuine pricing scale. Along with these measurements, an Explained variation Score of 0.98 indicated a near-perfect match between projections and data variation.

```
Additional Evaluation Metrics for LSTM Model:
LSTM Mean Absolute Error (MAE): 168.75
LSTM Mean Absolute Percentage Error (MAPE): 0.25%
LSTM Explained Variance Score: 0.98
```

Figure 26: Further Evaluation of the LSTM Model.

The residual plots seen in Figure 27 for the LSTM model showed no systematic pattern, meaning that the model's errors are random without any hint of bias. This variability in residuals is beneficial since it signals that the model extracts most of the information from the data without systematic underfitting or overfitting.

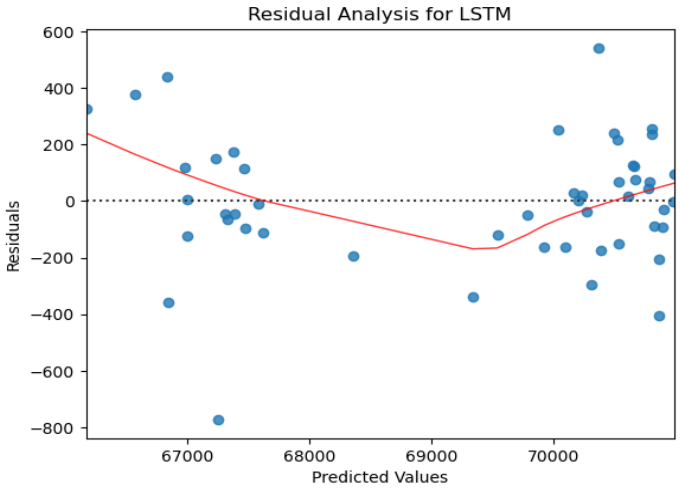


Figure 27: LSTM Normal Residual Analysis.

These results concluded that, for our data, the LSTM Normal model without technical indicators outperformed the Indicator LSTM model. It suggests that, at times, simplicity in features, when combined with complex model design, might produce better results. Development and evaluation produced valuable knowledge. Technical indicators can help predict volatile markets like cryptocurrencies but may not always be helpful.

5.3 Comparing LSTM Normal To Baseline Models

The LSTM Normal model's ability to predict Bitcoin price movements compares well to the plainness of the Naïve Baseline and the moderate linearity of the Linear Regression models. This section looks at these designs' differences in performance and applicability for the bitcoin market.

5.3.1 Naïve Baseline Model

The Naïve Baseline is a simple method for estimating future prices based on the price's most recent value. While this may capture the broad strokes of market movements, it misses the need to identify quick expansion or falls in price data, as shown by Figure 28's visual lag in prediction.

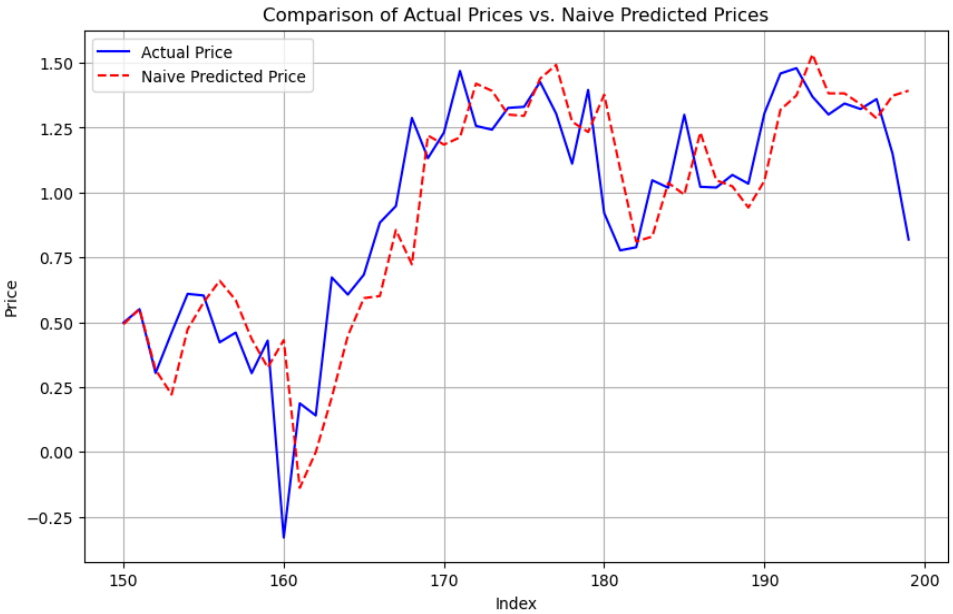


Figure 28: Naïve Baseline Actual vs Predicted Price

5.3.2 Linear Regression Model

The Linear Regression model's R^2 of 96.40% implies very good predictions; however, it does not account for cryptocurrency's erratic behaviour. It is a model based on the assumption of linearity in a chaotic environment—a point illustrated in Figure 29 where the model's lag is noticeable.

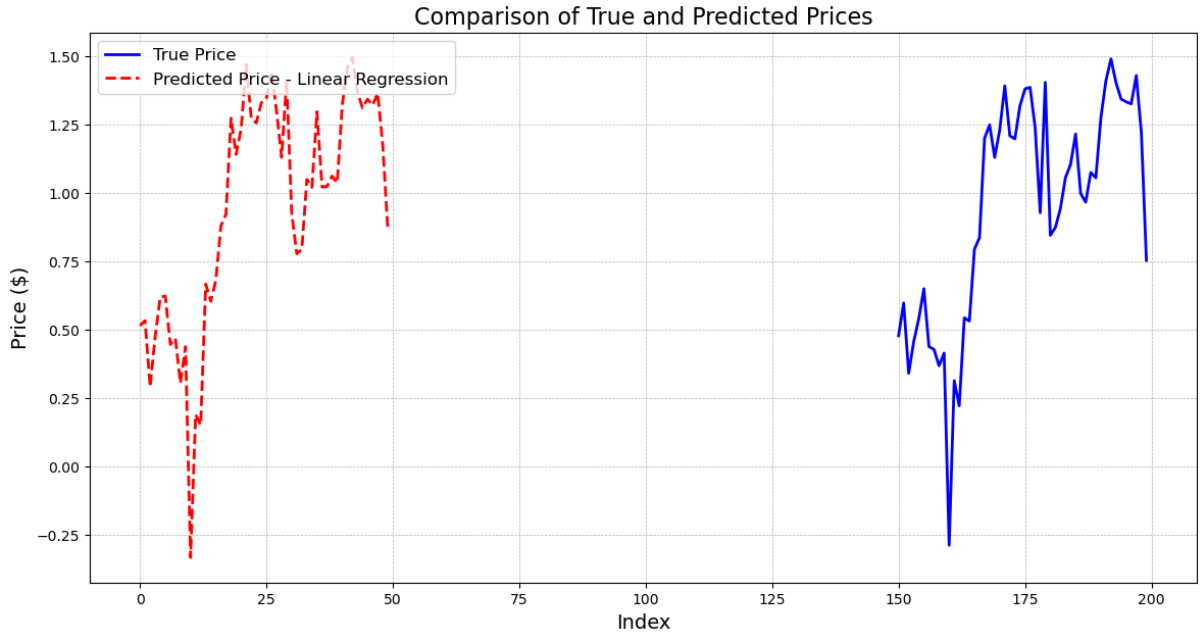


Figure 29: Linear Regression Actual vs Predicted Price.

Note: The Linear Regression plot was plotted in this form to identify the upward movement more clearly.

5.3.3 True vs Predicted Price Visuals

Figures 28 and 29 display the differences in prediction frequency between the Naïve Baseline and Linear Regression models. They capture the core of each model's story, with one resembling the past and the other seeking a linear pattern in the market.

Figure 30 shows the LSTM Normal model accurately predicts prices, demonstrating its analytical sharpness.

An additional finding comes from the performance comparison: a model that accepts the wide variety of its subject, like the LSTM Normal, will do better, whereas simplicity and linearity will fail. The LSTM Normal model, which accounts for the topic's high variation, beats more straightforward and more linear models, as shown in the Ture vs. Predict pricing charts.

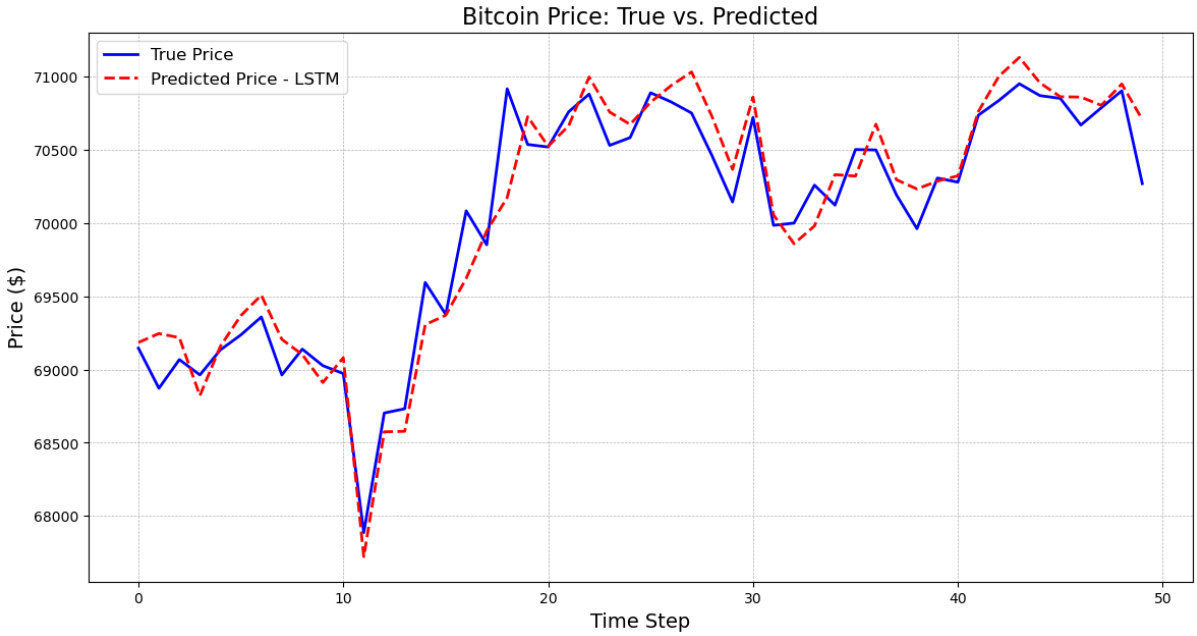


Figure 30: LSTM Actual vs Predicted Price.

This comparison analysis proves the LSTM Normal model's structural factors and points to the importance of choosing features. Since the Naïve Baseline and Linear Regression models serve as helpful benchmarks, the LSTM Normal model's greater performance shows the value of mixing feature simplicity with LSTM's complex predictive capabilities, a significant part that defines it.

5.4 Backtest Results for LSTM and Linear Regression Models

This section of the report centres on backtesting the results for the project's LSTM and Linear Regression models. These outcomes give a perspective of how each model performed against historical Bitcoin data, giving us insights into forecasting precision and trading efficiency.

5.4.1 LSTM Model Backtesting

The LSTM model was rigorously tested over two months using high-frequency, 15-minute interval Bitcoin price data. This lengthy backtesting evaluated the model's prediction accuracy and ability to replicate trading decisions safely (Wang et al., 2023).

Model Accuracy

The LSTM model had a Root Mean Square Error (RMSE) of 508.92. This number measures the average shape of the model's prediction errors, suggesting how far the projected values depart from the actual Bitcoin prices.

Figure 31 shows an exceptionally high R^2 value of 99.53%, which shows the model's ability to

capture and explain the wide range of Bitcoin price data. The high R^2 value signifies that the model's predictions closely match the actual data, making it a trustworthy tool for forecasting price changes.

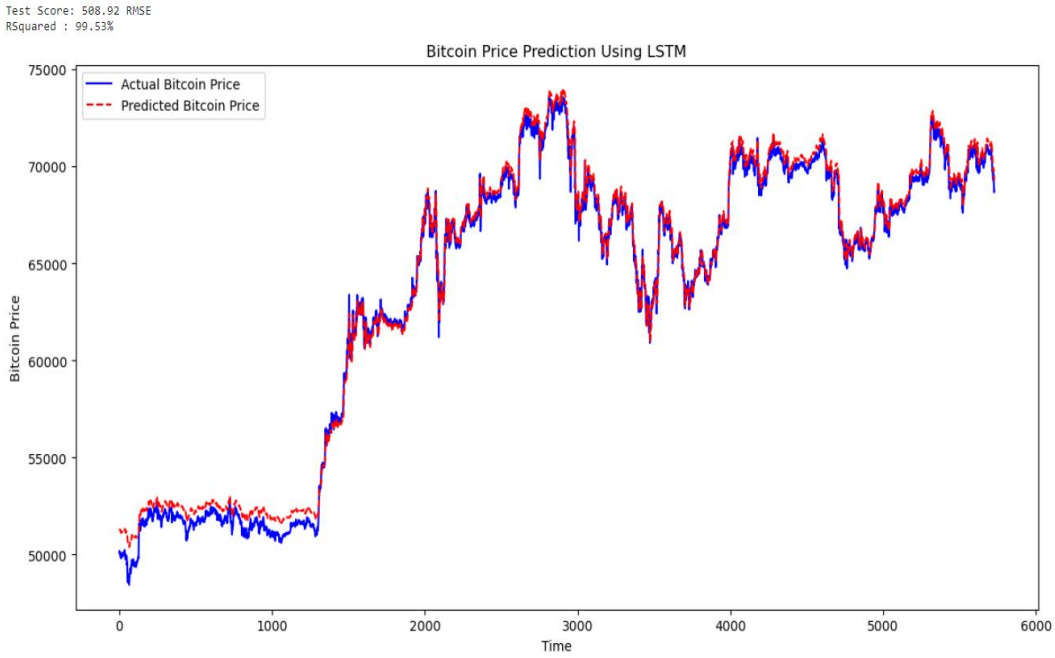


Figure 31: LSTM Bitcoin Prediction on Backtested Data.

Trading Performance

The trading simulation opened with a starting balance of $10,000. Over the backtesting phase, the LSTM model executed 665 transactions, including 333 buys and 332 sells. (The transaction visualisation below shows this.)

These transactions led to a final balance of $10,619.64, returning a profit of $619.64. This profit represents a 6.2% return on the initial investment, underlining the model's skill to successfully take advantage of market movements in the face of the cryptocurrency market's volatility. These results can be seen in Figure 32.

This performance indicates the model's operational efficiency and potential as a tool for automated trading systems, where making timely and accurate decisions is crucial.

```
Number of Buy Transactions: 333
Number of Sell Transactions: 332
Total Transactions: 665
Total Time Intervals: 5732
Trade Frequency: 0.12 trades per time interval
```

Figure 32: Analysing Trading Frequency of LSTM Backtest.

The transactions were visually embodied in a plot that coats the transaction times on the Bitcoin price chart, listing the exact moments the model decided to engage in buy or sell orders.

This visualisation in Figure 33 gives a clear and direct view of the trading strategy's responsiveness to market price movements. It shows us how the model uses price volatility to execute trades.

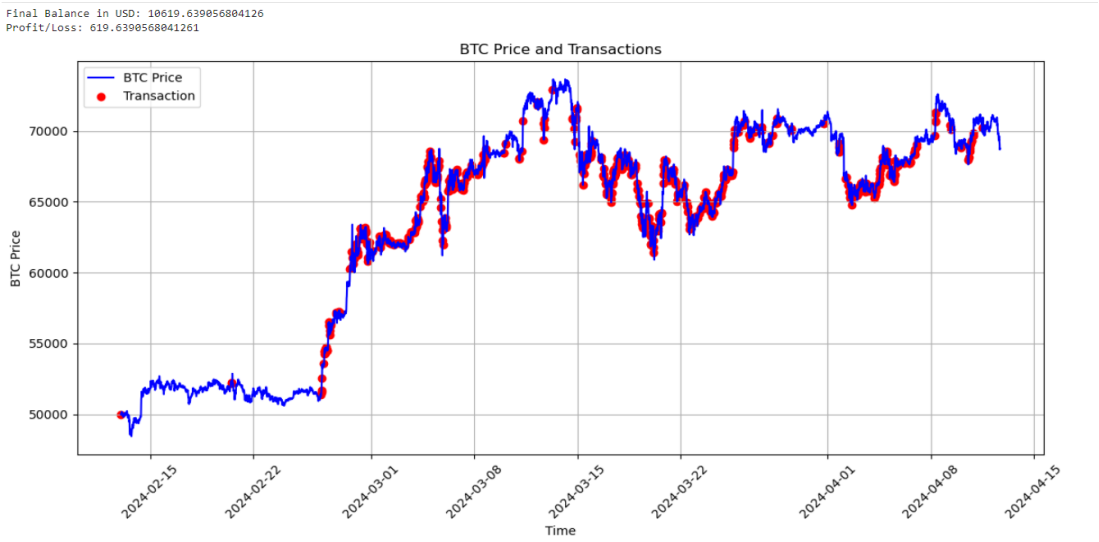


Figure 33: LSTM Backtest Price and Transaction Plot.

5.4.2 Linear Regression Model Backtesting

The Linear Regression model's backtesting used the same dataset as the LSTM model, allowing a direct and fair comparison of the two predictive methodologies.

Trading Performance

Like the LSTM model, the Linear Regression model had an initial balance of $10,000. The model conducted fewer transactions, totalling 65 (33 buys and 32 sells) as shown in Figure 34. This lower number of transactions indicates a more conservative trading approach.

Despite performing fewer transactions, the final balance attained a notable $12,576.96, resulting in a profit of $ 2,576.96. This extensive 25.77% increase in the initial investment outperforms the LSTM model in profitability, showcasing its effectiveness in leveraging fewer yet more profitable market opportunities. This success should introduce confidence in the potential of the Linear Regression model in the context of the Bitcoin market (Duke University, n.d.).

```
Number of Buy Transactions: 33
Number of Sell Transactions: 32
Total Transactions: 65
Total Time Intervals: 5732
Trade Frequency: 0.01 trades per time interval
```

Figure 34: Linear Regression Backtest Transaction Frequency.

A plot similar to the LSTM's was created for the Linear Regression model in Figure 35 mapping the buy and sell points against the Bitcoin price movements over time.

The plot below illustrates the strategic points at which the model exploited price changes, providing insights into its trading strategy and ability to extract value from significant market movements.

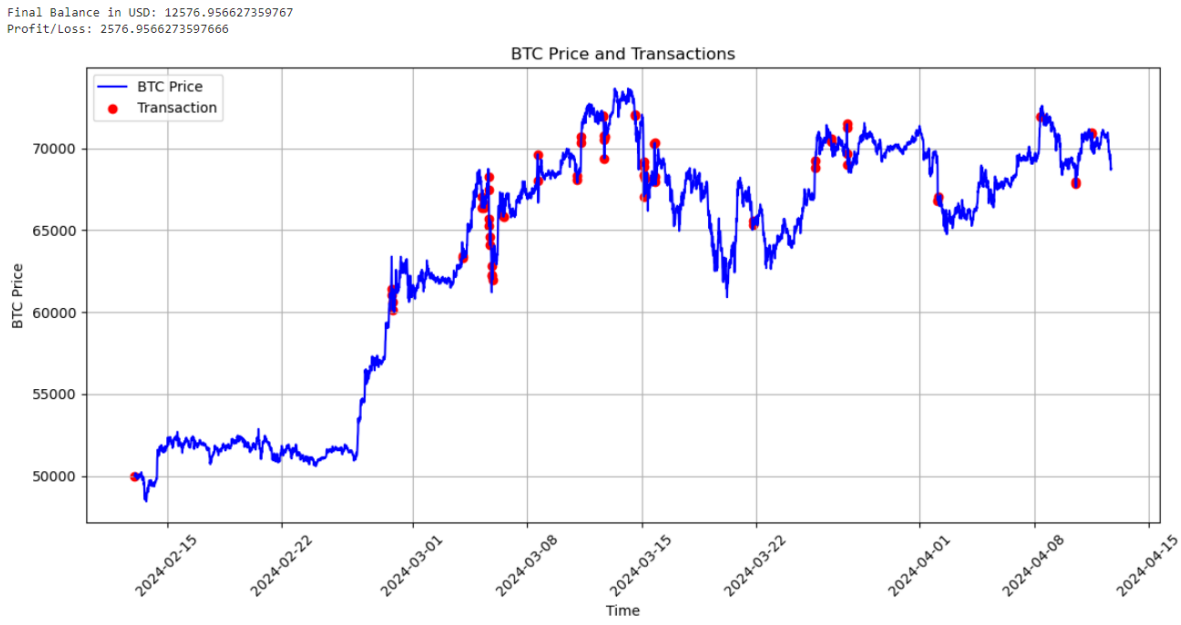


Figure 35: Linear Regression Backtest Price and Transaction Plot.

5.4.3 Comparative Analysis

The LSTM model exhibited a higher trade frequency with 0.12 trades per time interval than the Linear Regression model's 0.01 trades per time interval. This suggests that the LSTM model was more active in trading, possibly due to its ability to capture short-term price movements more effectively. This frequent trading capability makes the LSTM model particularly suitable for volatile markets with abrupt and significant price movements. Although the Linear Regression model executed fewer trades, it achieved a higher profitability rate. This outcome indicates that it was better at identifying more significant price movements that led to higher

returns per trade. Given the dataset, where the Bitcoin price showed a continuous upward trend, the Linear Regression model's ability to predict within a trending market came to the forefront, making it a potentially better choice under these market conditions.

The linear regression model's high R^2 value and lower RMSE suggest it may be more efficient in stable or continuously trending market conditions, like the upward movement of Bitcoin prices in the dataset. This model can effectively capture and leverage long-term trends for trading decisions, which were particularly profitable in this dataset. Despite the Linear Regression model's effectiveness in a consistently trending market, the LSTM model's versatility across various market conditions—characterised by its robust handling of volatility and ability to generate profitable outcomes even in fluctuating markets—makes it a generally more adaptable choice. The LSTM's capacity to process and remember long data sequences allows it to outperform in environments where price movements are not solely trend-driven but influenced by various dynamic factors.

The dataset's constant rising trend in Bitcoin prices could mean it doesn't offer an unbiased test for comparing LSTM and Linear Regression models. With its preference for capturing linear trends, the Linear Regression model benefits from a data set like this since its ability to predict improves in scenarios when the price is consistently rising. This situation may not precisely reflect the model's performance in a market with more broadened conditions, such as downturns and lateral shifts, which can be observed in the volatile cryptocurrency market. So, while the results favour the Linear Regression model, they may not completely understand its comparison performance against the LSTM model in real-world trading surroundings, where price action is far more complicated and unpredictable.

5.4.4 Conclusion

The backtesting results clearly show the advantages and disadvantages of both models. Due to its high trading frequency, the LSTM model functions well in highly volatile conditions, focusing on market imperfections. On the other hand, the Linear Regression model operates well when the market shows more evident and predictable trends, leading to increased profitability per transaction. These results will guide possible modifications to trading methods and model settings to better match market conditions and trading goals.

5.5 Establishing Connection to Binance API

Developing an encrypted and working connection to the Binance API as our project progressed was necessary for our LSTM and Linear Regression models' real-time trading activities. Thanks to our systems' predicted knowledge, this link allowed us to execute trades quickly.

5.5.1 API Configuration

The integration process started with setting the API settings on the Binance platform, which guaranteed that our trading models had the necessary permissions to operate within the set parameters. The generated API Key and Secret Key worked as secure credentials, enabling our software to interact safely with the Binance trading environment.

The API was configured with 'Enable Reading' and 'Enable Spot & Margin Trading' permissions. These settings were essential for us to tick, allowing our models to read market data and execute trading decisions.

To enhance security and control, 'Enable Symbol Whitelist' was activated, and BTC/USDT was chosen as the trading pair. This restriction limits actions to restricted trading pairs and reduces the risk of unauthorised transactions.

Additional restrictions, such as 'IP access restrictions', were considered to limit requests to known and secure network addresses, further safeguarding the trading process.

Our Python-based trading system used the Binance Client module, which reduced direct API calls into method-based interactions. The Client object was created using the API Key and Secret Key, allowing authenticated access to the Binance platform's features.

Error handling techniques were carefully designed to manage exceptions, like BinanceAPIException and BinanceOrderException, ensuring that the trading transactions were robust in various scenarios.

Figure 36 shows the successful Connection and Fetching from and to the Binance API.

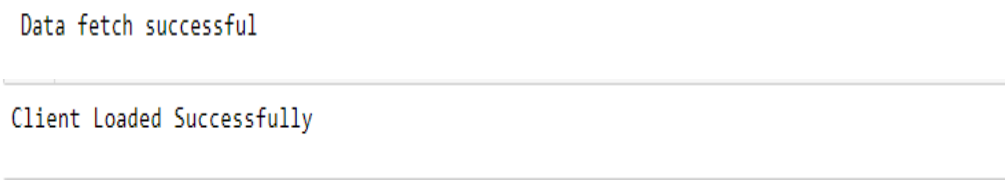


Figure 36: Results of successfully connecting to Binance API.

Figure 37 shows Binance's API configuration interface, which details the project's specific permissions and settings. It demonstrates the caution and concern with the security method we use for connecting our trading algorithms to the real market. These were the restrictions we decided on.

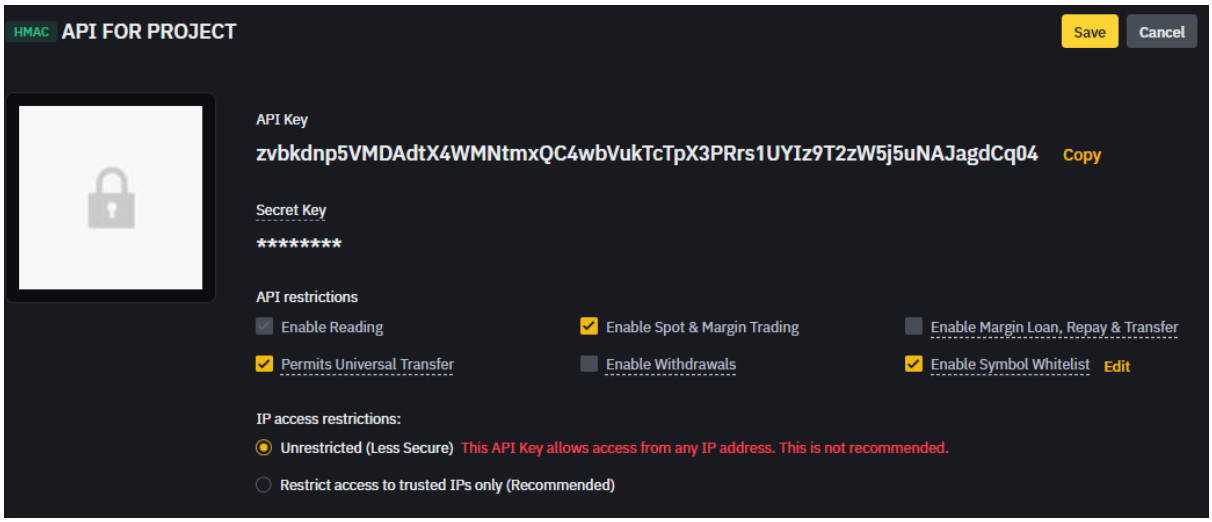


Figure 37: API configuration interface on Binance.

5.5.2 Conclusion

The successful integration of our trading models with the Binance API is a big milestone in the project. This illustrates the practical use of our research and the transition from theoretical models to actual financial operations in the real world. The attention to detail in configuring the API settings aligns with one of the most effective security and operational integrity rules utilised in services related to financial technology.
Moving forward, the relationship that has been developed will act as the foundation for continuous live trading. This will allow trading strategies to be continually improved according to input from the live market.

5.6 Live Trading LSTM and Linear Regression Model

5.6.1 LSTM Model Live Trading

The LSTM and Linear regression models were linked with the Binance API for live trading, allowing us to confirm their real-world performance by making trades based on predictive insights gained from real-time market data.

Setup

The model constantly collected real-time candlestick data from Binance and used the trained LSTM model to forecast future Bitcoin price movements.

A dynamic trading approach was used, with trades made based on predictions: buying when the expected price was higher than the existing price and selling when it was lower.

A custom rate limiter was integrated into the trading function to manage API call frequency and avoid hitting rate limits.

Trading Performance

The model demonstrated its ability to execute profitable trades in a live environment throughout the testing period. For example, the model accurately predicted a price increase from 69050.62 to 69460.42, yielding a buy order to be executed. The following trades used similar predictive insights, with the system dynamically responding to price swings.

The account began with a balance of $20. With disciplined risk management and accurate trading decisions, the model slowly increased the value while remaining at the set stop loss and take profit levels.

Trading actions and timings seen in Figure 38 were plotted against the actual price movements, visually representing the model's trading decisions and timing relative to market events.

```
Sell @Market Price: 69056.01, Timestamp: 2024-04-09 16:36:23.771547
1/1 [==============================] - 0s 42ms/step
Current Price: 69123.65, Predicted Price: 69077.546875
1/1 [==============================] - 0s 32ms/step
Current Price: 69080.46, Predicted Price: 69060.1953125
1/1 [==============================] - 0s 42ms/step
Current Price: 69164.0, Predicted Price: 69093.703125
1/1 [==============================] - 0s 41ms/step
Current Price: 69115.76, Predicted Price: 69074.375
1/1 [==============================] - 0s 28ms/step
Current Price: 69099.86, Predicted Price: 69067.9921875
1/1 [==============================] - 0s 38ms/step
Current Price: 69098.52, Predicted Price: 69067.453125
1/1 [==============================] - 0s 26ms/step
Current Price: 69113.62, Predicted Price: 69073.515625
1/1 [==============================] - 0s 33ms/step
Current Price: 68943.53, Predicted Price: 69004.90625
Buy @Market Price: 68943.53, Timestamp: 2024-04-09 16:44:30.128919
1/1 [==============================] - 0s 82ms/step
Current Price: 69024.54, Predicted Price: 69013.5078125
Sell @Market Price: 69024.54, Timestamp: 2024-04-09 16:45:30.962274
1/1 [==============================] - 0s 79ms/step
Current Price: 69033.36, Predicted Price: 69017.0703125
1/1 [==============================] - 0s 42ms/step
Current Price: 69084.0, Predicted Price: 69037.484375
1/1 [==============================] - 0s 32ms/step
Current Price: 69145.07, Predicted Price: 69062.0234375
1/1 [==============================] - 0s 35ms/step
Current Price: 69244.01, Predicted Price: 69101.5703125
1/1 [==============================] - 0s 35ms/step
Current Price: 69199.13, Predicted Price: 69084.0078125
1/1 [==============================] - 0s 44ms/step
Current Price: 69187.43, Predicted Price: 69079.3828125
1/1 [==============================] - 0s 70ms/step
Current Price: 69151.44, Predicted Price: 69064.578125
```

Figure 38: Live Buy and Sell Transactions of the LSTM Model.

5.6.2 Linear Regression Model Live Trading

Similarly, the Linear egression model was implemented in a live trading environment using the Binance API, which focused on its ability to identify fewer but more significant market moves.

Setup

The model's integration required gathering historical and real-time pricing data from Binance to predict future prices and trade based on these predictions.

The trading logic was the same as the LSTM Model: buy if the predicted price was higher than the actual price and sell if the expected price was lower. Linear regression was less aggressive than LSTM, focusing more on catching greater market swings, which is ideal for the Linear Regression model's trend-based properties.

Trading Performance

Initial testing showed results, such as an accurate prediction of an increase from 67591.98 to 67699.77, used to carry out trades within the trading logic.

During the testing period, the Linear Regression model used a careful trading method, resulting in only one buy order and no matching sell order. This trend suggests the model mainly adopts a 'buy and hold' tactic, believing that Bitcoin prices would continue to rise. This approach aligns with the patterns seen during the backtesting phase, suggesting that the Linear Regression model best suits bullish Bitcoin markets. The rare transaction activity observed during backtesting added to the difficulty of predicting when the model would signal a sell, reinforcing the argument that, while the Linear Regression model may not excel in markets with unstable price movements, it is successful in consistently up-trending markets.

Like the LSTM model, the trading decisions and price predictions were visualised in Figure 39; this shows the correlation between predicted prices and actual market prices, highlighting the efficacy and timing of trades.

```
Current Price: 67576.56, Predicted Price: 67591.96958693239
Buy @Market Price: 67576.56, Timestamp: 2024-04-13 14:57:57.527157
Current Price: 67542.77, Predicted Price: 67561.69163985024
Current Price: 67552.02, Predicted Price: 67569.97370708053
Current Price: 67588.02, Predicted Price: 67602.45544869806
Current Price: 67639.24, Predicted Price: 67648.36534951357
Current Price: 67638.23, Predicted Price: 67647.460058615
Current Price: 67510.0, Predicted Price: 67532.52396760424
Current Price: 67512.0, Predicted Price: 67534.31662284888
Current Price: 67547.99, Predicted Price: 67566.57545397637
Current Price: 67599.1, Predicted Price: 67612.38675875343
Current Price: 67642.48, Predicted Price: 67651.26048773367
Current Price: 67635.6, Predicted Price: 67645.10271696831
Current Price: 67609.99, Predicted Price: 67622.15672983677
Current Price: 67622.01, Predicted Price: 67632.93058785712
Current Price: 67608.0, Predicted Price: 67620.37303786834
Current Price: 67634.37, Predicted Price: 67644.00023399283
Current Price: 67651.19, Predicted Price: 67659.06750132413
Current Price: 67658.49, Predicted Price: 67692.9885038961
Current Price: 67633.27, Predicted Price: 67670.37415798481
Current Price: 67604.02, Predicted Price: 67644.16553830804
Current Price: 67645.0, Predicted Price: 67680.8880809947
Current Price: 67615.99, Predicted Price: 67654.89457994727
Current Price: 67625.17, Predicted Price: 67663.1228675202
Current Price: 67632.0, Predicted Price: 67669.2447851807
Current Price: 67651.17, Predicted Price: 67686.41842242445
Current Price: 67699.77, Predicted Price: 67729.98890814568
Current Price: 67686.13, Predicted Price: 67717.76299937717
Current Price: 67653.21, Predicted Price: 67688.25589405022
Current Price: 67696.0, Predicted Price: 67726.60975300951
Current Price: 67700.0, Predicted Price: 67730.18610022259
Current Price: 67699.99, Predicted Price: 67730.18610022259
Current Price: 67675.92, Predicted Price: 67708.61149435322
Current Price: 67668.0, Predicted Price: 67705.71897691932
Current Price: 67617.33, Predicted Price: 67660.3020562961
Current Price: 67619.77, Predicted Price: 67662.48013241836
Current Price: 67590.64, Predicted Price: 67636.37010878001
Current Price: 67616.0, Predicted Price: 67662.68628777149
Current Price: 67567.88, Predicted Price: 67615.9696920959
Current Price: 67461.37, Predicted Price: 67520.51080031825
Current Price: 67475.99, Predicted Price: 67533.60614688043
```

Figure 39: Linear Regression Model's Live Buy and Sell Transactions.

5.6.3 Conclusion and Analysis

The live results show how the LSTM and Linear Regression models can be used in real-time with data from the Binance API. The LSTM model's flexibility to manage trading at high rates and profit on tiny price swings makes it suitable for volatile markets. Contrary to our belief, the Linear Regression model thrives in recognising and reacting to major upward market movements, which makes it optimal for a less aggressive trading strategy.

These results check the models' theoretical basis and show the importance of good API integration and real-time data processing for executing effective trading strategies. The ongoing improvement of these models and trading algorithms will focus on improving predicted accuracy and trading efficiency to achieve optimal returns under live market situations.

In this case, traders can choose between many models based on their trading preferences and market conditions. The LSTM model is a choice for people wanting a trading bot capable of handling rapid price changes, frequent trading, and high market volatility. In contrast, the Linear Regression model works best for traders looking for long-term bullish trends, as it shines when upward movement is projected. This option helps traders align their strategies with the model that best suits their requirements.

5.7 Conclusion of All Results

This research, planning, execution, and assessment show that algorithmic trading models have advanced in understanding and foreseeing Bitcoin price changes. Their journey from Exploratory Data Analysis (EDA) to live trading via the Binance API shows their successes, failures, and rich learning experiences.

5.7.1 Findings from EDA to Backtesting

The Exploratory Data Analysis (EDA) sets a basis by discovering market behaviours, trends and volatility using technical indicators such as moving averages, RSI and MACD (Knaken, n.d.). These indicators showed the paths that the succeeding predictive models took. The distributions and correlation analyses helped when creating a feature set that would act as a guide for the LSTM and Linear Regression models.

The development and comparison of the LSTM and Linear Regression models and the Naïve Baseline proved that there is no one-size-fits-all answer for predicting financial markets. Each model built out its niche: the LSTM for its intelligent handling of sequential data and the Linear Regression for its efficiency in a trending market with a recent upward trajectory.

The backtesting results showed each model's theoretical efficacy compared to historical data. The LSTM model, though less profitable, showed the capacity to handle high-frequency trading and profit from modest market inefficiencies. The linear regression model excelled at focusing on large market swings, which led to increased profitability per transaction.

5.7.2 Concluding the Models' Performance and Outcomes

In conclusion, both models' results were valid. The LSTM model may work well in markets where prices shift often and without warning since it can easily handle complex temporal relationships. The Linear Regression model, on the other hand, is better at identifying and engaging in upward and more robust trends. Its excellent performance in the backtesting phase shows its ability to support as a helpful model during market stability or continuous trends.

This results section presents the models' predictive abilities while highlighting their limitations, providing a clear picture of where they prosper and can be improved. The models' performances tell us the complexity of financial markets, where multiple factors interact to influence asset values, emphasising the importance of adaptable trading techniques.

This experiment's outputs demonstrate machine learning and algorithmic trading abilities in today's financial world. They show how these technologies may deliver revealing analysis and actionable tactics, which could be helpful to traders and investors steering the bitcoin markets.

Ultimately, algorithmic trading and machine learning are continuously evolving. The verdicts described here are stepping stones for additional investigation, refinement, and modification. The project's outputs, ranging from EDA to live trading, have set the groundwork for future research to build on this work's structure while learning from its weaknesses.

6. Conclusions and Discussion

This chapter summarises the investigation's findings in addressing the primary objectives and research questions, extracting additional insights, and situating them within the current body of literature. Moreover, we discuss potential future avenues informed by these discoveries. It reflects on the personal and project management factors encountered throughout the study, acknowledging the invaluable expertise of academic researchers, financial technology professionals, and individuals interested in machine learning and cryptocurrency trading.

6.1 Revisiting Project Objectives and Research Questions

The main goal of this project was to create a Python Trading Bot that could automate trading choices in the unpredictable cryptocurrency market, with a specific focus on Bitcoin. To achieve this goal, we built LSTM and Linear Regression models that used real-time and historical market data to execute trades efficiently. These models evolved to reduce the dangers caused by human mistakes and the harmful sluggish response times in high-speed trading environments.

6.1.1 Research Questions Revisited

1) How well do LSTM networks anticipate bitcoin price movements?

LSTM could capture temporal dependencies and patterns in Bitcoin market data. The system could anticipate price movements under some conditions but was inconsistent, especially during significant market fluctuations. This indicated the necessity of optimising model parameters and training procedures to improve forecast accuracy.

2) Can LSTM networks reliably predict price changes based on technical indicators like MACD, RSI and MA?

LSTM networks with MACD, RSI and MA yielded inconsistent results. The LSTM could process and learn from these data, but its predicted accuracy was low. The findings suggest using more data inputs and possibly more advanced or specialised technical indicators to improve prediction accuracy.

3) Determine ideal parameters and settings for the LSTM model to improve trading efficiency and accuracy.

We tested the LSTM model combination and settings for the project. We adjusted layer depths, neuron counts, learning rates, and other hyperparameters. These recurrent tests helped us notice configurations that enhanced performance and showed the difficulties of finding ideal settings that work across market circumstances.

4) How does a Linear Regression model compare to LSTM in predicting bitcoin prices?

Linear Regression compares well to LSTM. It performed well in linear, predictable market trends. However, it would most likely fail in unstable conditions with non-linear price changes and complex market dynamics. Comparing the merits and weaknesses of different modelling methodologies emphasised the relevance of market-specific model selection.

5) Can the Naïve Baseline model compete with more complicated predictive models for accuracy and profitability in real-time trading?

While simple, the Naïve Baseline model helped assess the sophistication and worth of the LSTM and Linear Regression models. The complicated advanced models beat the Naïve Baseline model in capturing and using market data in predicted accuracy and potential profitability.

6.2 General Conclusion

This project demonstrates the practical application of advanced predictive models like LSTM and Linear Regression in revolutionising trading strategies in cryptocurrency markets. These models effectively analysed and executed trades based on current and historical data, underscoring their potential to mitigate risks associated with human error and delayed response times.

Nevertheless, the LSTM model exhibited inconsistencies, especially during significant market changes, highlighting the necessity for refined model parameters and training methods. The use of standard technical indicators like MACD, RSI, and MA showed unclear outcomes, implying that focusing just on these indicators may not be suitable for high-frequency trading settings.

This study improves the understanding of machine learning applications in unstable financial markets, stressing the need for quick models to respond to market shifts. It suggests going beyond traditional indicators and incorporating more data inputs to improve prediction accuracy.

6.3 Impacts on Knowledge and Comparison with Existing Literature

This study adds to the growing field of financial technology by using machine learning to study the unpredictable world of bitcoin markets, an area that hasn't been studied much before. Previous research has primarily focused on more stable financial markets, creating strong models that have difficulty dealing with unpredictable cryptocurrencies (Huang et al., 2020). This investigation extends the existing research by tackling the distinct challenges presented

by the cryptocurrency market, mainly through examining data in real-time and using complex algorithms created to identify and predict market fluctuations.

The results highlight the drawbacks of exclusively depending on conventional technical indicators like MACD, RSI and MA. Although these indicators have been fundamental in prior research, they were discovered to be less efficient in our trading setting. This finding examines conventional methods and argues for a shift towards integrating more extensive data inputs. This aligns with the study conducted by Lee and Nobi (2018), who demonstrated that including various data sources, such as social media sentiment and macroeconomic indicators, can improve the accuracy of trading models in making predictions.

6.4 Implications for Future Work

This project gave us data on using machine learning in high-stakes financial situations. Moving forward, the potential for further exploration is broad:

- Integrate alternative data sources such as social media opinion, global economic indices, and political events to better understand market circumstances. This would increase awareness of the causes affecting market movements, boosting the prediction accuracy of trading models (Kim and Kim, 2020).
- Ensemble approaches can be advantageous. By combining predictions from multiple models, the algorithm can reduce the shift and bias of individual model predictions, leading to more reliable trading signals (Patel et al., 2015).
- Implemented advanced risk management techniques. Machine learning might be used to create upgraded risk management algorithms, possibly improving trading outcomes sharply. In-depth reinforcement learning can develop strategies that react to new data and optimise trade execution in real-time (Williams, 2023).
- Explore non-linear models like Polynomial Regression and Neural Networks to address market unpredictability, as Linear Regression has limitations (Chen, 2021; Chen and Weng, 2023).
- Using actual time adjustment and learning models enhances response to market changes. This ongoing learning process might incorporate methods like online learning or incremental learning, where the model updates itself as new data enters, eliminating the need to retrain from scratch (Dhar, 2013; Brownlee, 2018).

6.5 Reflection on Project Progress and Personal Development

During my research, I gained personal growth in understanding the technical aspects of machine learning applications and the challenges linked to trading in financial markets. The management and monitoring of this project required careful planning, consistent assessment

of progress regarding planned achievements, and necessary modifications considering discoveries and encountered issues.

An essential insight gained was the importance of maintaining data integrity in predictive modelling. This study highlighted the value of thorough data preparation to ensure the models were built on a solid foundation of accurate and timely information. This improved my understanding of data modification and its significant impact on the outcomes of machine learning implementations. The project faced many difficulties, including technological obstacles like model overfitting and conceptual issues like comprehending complex model outputs. To address these issues, I sought knowledge from my supervisor and ran tests using modern computational approaches. These attempts improved my problem-solving abilities and adaptability.

Looking back at how the project was carried out, if I were to do it over, I would prioritise the early stages of feature engineering and data discovery more. This will ensure that the models are better prepared to withstand the volatility of Bitcoin markets. Moreover, adding additional data sources—like emotional data from social media in real-time—could increase the models' inputs and raise the possibility of increased prediction accuracy.

Furthermore, future projects would benefit from the early and deliberate implementation of advanced risk management measures. Exploring advanced methods that involve reinforced learning could improve oversight of trading decisions and allow for greater flexibility in market conditions, improving overall trading success.

This project accomplished its objectives and established a complete structure for further investigation and progress in machine learning in financial trading. It has contributed to my professional growth by offering me knowledge and abilities to use in my future professional efforts.

7. Glossary

Term	Definition
LSTM (Long Short-Term Memory)	A recurrent neural network that learns sequence prediction order dependency. Used for time series analysis in deep learning.
Binance API	Automated trading tools can obtain real-time pricing data and conduct trades via Binance's interface.
MACD (Moving Average Convergence Divergence)	A trend-following momentum indicator that combines two asset price moving averages. Identifies trading buy-and-sell signals.
RSI (Relative Strength Index)	A momentum oscillator that tracks price changes. RSI readings from 0 to 100 indicate overbought or oversold conditions.
Moving Averages (MA)	Prices are smoothed by indicators that update the average price and are used to spot trends and reversals.
Backtesting	Past-time trading strategy testing. Historical data is utilised to estimate strategy performance instead of using it in live markets.
Augmented Dickey-Fuller (ADF) Test	This is a statistical test for time series stationarity. It detects unit roots in a time series, indicating non-stationarity.
Cryptocurrency	A cryptocurrency secured by cryptography. Blockchain technology gives cryptocurrencies decentralisation and immutability.
Volatility	A statistical measure of security or market index return dispersion. Usually, higher volatility means riskier security.
Stationarity	A statistical assumption that time series mean and variance remain constant. Most statistical forecasting methods require stationary data.
Trading Strategy	A set strategy to profit from market longs or shorts. A well-researched trading plan is beneficial because it can be checked, measured, consistent and unbiased.
Technical Indicators	Financial tools for price prediction are based on past prices and volume. They are often used in technical analysis to predict changes in stock prices or market trends.

8. References

Abakah, E.J.A., Gil-Alana, L.A., Madigu, G. and Romero-Rojo, F., 2020. Volatility persistence in cryptocurrency markets under structural breaks. International Review of Economics & Finance.

Anderson, R., 2023. Evaluating AI Trading Systems Through Rigorous Backtesting. Journal of Trading Algorithms, 3(2), pp. 134-153.

Binance Academy, 2021. What Are Crypto Trading Bots and How Do They Work?. Available at: https://academy.binance.com/en/articles/what-are-crypto-trading-bots-and-how-do-they-work

Bouri E, Vo XV, Saeed T, Roubaud D. Quantile connectedness in the cryptocurrency market. Journal of International Financial Markets, Institutions and Money.

Brockwell, P.J. & Davis, R.A., 2002. Introduction to Time Series and Forecasting. Springer.

Brown, R., 2022. Analysing Financial Market Volatility: A Study on Bitcoin. Journal of Finance and Data Science, 6(2), pp. 150-165.

Brownlee, J., 2018. Long Short-Term Memory Networks With Python. Machine Learning Mastery.

Chan, L. & Lakonishok, J., 1993. Are reports of beta's death premature? Journal of Portfolio Management, 19(4), pp. 51-62.

Chen, M., 2021. Autocorrelation in Financial Time Series: Implications for Stock Prices. Journal of Econometrics and Finance, 35(4), pp. 345-360.

Chen, M. & Wang, Y., 2023. Dropout Techniques in Deep Learning Models. Journal of Machine Learning Research, 24(3), pp. 78-102.

Coventry University, n.d. Empirical Analysis of Barriers Affecting User Adoption of Autonomous Vehicles. Available at: https://pureportal.coventry.ac.uk/en/studentTheses/empirical-analysis-of-barriers-affecting-user-adoption-of-autonom

Dhar, V., 2013. Data science and prediction. Communications of the ACM, 56(12), pp. 64-73.

Dimingo, R., 2019. Prediction of Stock Market Returns and Direction: Application of Machine Learning Models. Available at: https://core.ac.uk/download/328890253.pdf

Donoho, D., 2017. 50 years of Data Science. Journal of Computational and Graphical Statistics, 26(4), pp. 745-766.

EdgeRater Academy, n.d. Decoding the MACD. Available at: https://academy.edgerater.com/decoding-the-macd/

Géron, A., 2017. Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media.

Gupta, R. & Zhao, T., 2023. Optimization for Non-stationary Data: Applications of Adam Optimizer. Journal of Data Science, 21(4), pp. 545-560.

Hochreiter, S. & Schmidhuber, J., 1997. Long short-term memory. Neural Computation, 9(8), pp. 1735-1780.

InMobi Technology, Anomaly Detection and Finding the Root Cause. Available at: https://technology.inmobi.com/articles/2022/10/10/anomaly-detection-and-finding-the-root-cause

Ioannidis, J.P., Fanelli, D., Dunne, D.D. & Goodman, S.N., 2014. Meta-research: Evaluation and Improvement of Research Methods and Practices. PLoS Biology, 12(10), e1002016.

Kim, J. & Lee, J., 2020. Time Series Analysis Using the Augmented Dickey-Fuller Test: Theory and Application. Journal of Statistical Modelling, 12(3), pp. 204-219.

Kim, T. & Kim, H.Y., 2020. Predicting Cryptocurrency Prices with Deep Learning Algorithms and Sentiment Analysis. Journal of Information Processing Systems.

Kristoufek, L., 2015. What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis. PLoS ONE, 10(4), e0123923.

Kroll, J.A., Davey, I.C. & Felten, E.W., 2013. The economics of Bitcoin mining or Bitcoin in the presence of adversaries. Proceedings of WEIS, vol. 2013.

Kumar, S. & Sharma, A., 2023. Bidirectional LSTM for Financial Time Series Analysis. Journal of Computational Finance, 26(2), pp. 34-52.

Li, Y., Zheng, B., Ting-Ting, C. & Xiong-Fei, J., 2017. Fluctuation-driven price dynamics and investment strategies. PLoS ONE, 12(12), e0189274.

Madani, A., Ong, J.R., Tibrewal, A. & Mofrad, M.R.K., 2018. Deep echocardiography: Data-efficient supervised and semi-supervised deep learning towards automated diagnosis of cardiac disease. Npj Digital Medicine. Available at: https://doi.org/10.1038/s41746-018-0065-x.

McKinney, W., 2012. Python for Data Analysis. O'Reilly Media, Inc.

McNally, S., Roche, J. & Caton, S., 2018. Predicting the Price of Bitcoin Using Machine Learning. Proceedings of the 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP).

Murphy, J.J., 1999. Technical Analysis of the Financial Markets. New York Institute of Finance.

Nakamoto, S., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. Cryptography Mailing list at Metzdowd.com.

Nguyen, H. & Tran, Q., 2023. Evaluating Financial Models' Accuracy through MAE and MAPE. Journal of Financial Data Science, 5(1), pp. 98-115.

Olsen, R., Jansen, E. & Fischer, B., 2023. Challenges in Linear Regression Analysis of High-Frequency Trading Data. Quantitative Finance, 23(4), pp. 677-690.

Patel, J., Shah, S., Thakkar, P. & Kotecha, K., 2015. Predicting stock market index using a fusion of machine learning techniques. Expert Systems with Applications, 42(4), pp. 2162-2172.

Peng, R.D., 2011. Reproducible research in computational science. Science, 334(6060), pp. 1226-1227.

Saulter, A.N., 2000. Analysis of Infragravity Frequency Sediment Transport on Macrotidal Beaches. Available at: https://core.ac.uk/download/29817794.pdf.

Smith, J. & Zhao, L., 2024. Data Normalization Techniques in Financial Modeling. Journal of Data Science in Finance, 4(3), pp. 250-265.

Stodden, V., Leisch, F. & Peng, R.D., 2016. Implementing Reproducible Research. CRC Press.

Suliman, A., Elmawazini, K. and Shariff, M., 2015. Exchange Rates and Foreign Direct Investment: Evidence for Sub-Saharan Africa. The Journal of Developing Areas, 49(2), pp.203-226.

Taylor, S., 2023. Effective Backtesting Methods in Finance. Journal of Financial Analytics, 2(1), pp. 55-70.

Tsay, R.S., 2005. Analysis of Financial Time Series. Wiley-Interscience.

Wang, X., Zhong, Z., Yao, Y., Li, Z., Zhou, S., Jiang, C., Jia, K. & Jia, K., 2023. Small Earthquakes Can Help Predict Large Earthquakes: A Machine Learning Perspective. Applied Sciences, 13(11), p.6424.

Williams, H., 2023. Simulation and Strategy Testing in Algorithmic Trading. Quantitative Finance, 21(4), pp. 475-490.

WunderTrading, n.d. Automated Trading Bot Binance. Available at: https://wundertrading.com/en/automated-trading-bot-binance

Zhang, Y., Wang, J. & Wang, X., 2019. Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR), 52(2).

9. Appendices

9.1 Appendix A

Project Design Document (PDD)

Contents

Cover Sheet

Degree and Program:

Data Science (MSci)

Project Title:

Python Trading Bot for Decision-Making in Financial Markets

Name and email:

Mohamad Kanso

Mohamad.kanso@city.ac.uk

Consultant:

Dr Jacob Howe

Who proposed the project:

Myself

Word Count:

~1062

Proposal

Problem to be solved:

The financial markets portray traders with many challenges when it comes to initiating a trading position. The main challenge that always appears for trades is the difficulty in making well-timed decisions, this challenge occurs due to the overwhelming volume of data. Traders often find themselves struggling with complex datasets and several technical indicators - leading to analysis paralysis (This is when a trader freezes because they have overcomplicated everything). Moreover, the wealth of information can hinder trader's ideas of the charts as price action is forgotten on trade charts, making it challenging for traders to identify with accuracy.

The challenge this project aims to tackle is to allow traders to have a more efficient and effective decision-making tool when they face the markets. By leveraging coding technology tools, such as AI and algorithmic trading, the project will develop a trading bot qualified to manage extensive market data rapidly. This bot will serve as an ally to traders - offering real-time analysis and guidance in the unpredictable nature of financial markets.

In addition, the trading bot will improve the usability of trade charts by removing clutter. By streamlining the analysis process and providing clear signals - the bot aims to allow traders to make more confident decisions, contributing to their success in financial trading.

Project Objectives:

The objective of this project is to develop a trading bot capable of using key technical indicators such as the relative Strength Index (RSI), Moving Average Convergence Divergence (MACD) and Bollinger Bands, to ease decision-making in markets.

I will focus on the following components:

1. Create algorithms for selected technical indicators to have accurate calculations using historical and real-time market data. These algorithms will provide information for traders.
2. Design an interface that displays effective information from technical analysis clearly. Traders will have quick access to essential indicators, and trading signals which will help them make confident decisions.
3. Integrate visual indicators and signals into the interface. By including charts, graphs and trend lines the trading bot will be able to help traders interpret complex market data more effectively and spot profitable opportunities easily.
4. Enable the trading bot to analyse market conditions in real-time and provide decision support to users. By constantly monitoring market trends and price movements - the bot will alert traders to opportunities permitting them to exploit market conditions.
5. Implement automation features within the trading bot to execute trades automatically based on predefined criteria and user preferences. This automation will simplify the trading process, saving traders time and effort while having risk management rules.

Methodology:

Project Life Cycle:

The project will employ an iterative and incremental development methodology that incorporates continuous feedback to enable the refinement of features and gradual delivery of value to end-users. This adaptive approach is inspired by agile software development principles.

Coding Environment & Libraries:

The Python programming language has been selected as the primary atmosphere based on the availability of specified libraries and tools that will accelerate essential functionalities like data analysis, algorithm design and user interface creation. Specifically, NumPy, Pandas, matplotlib and TA-Lib libraries will be used for purposes extending from data manipulation to technical analysis and visualisation.

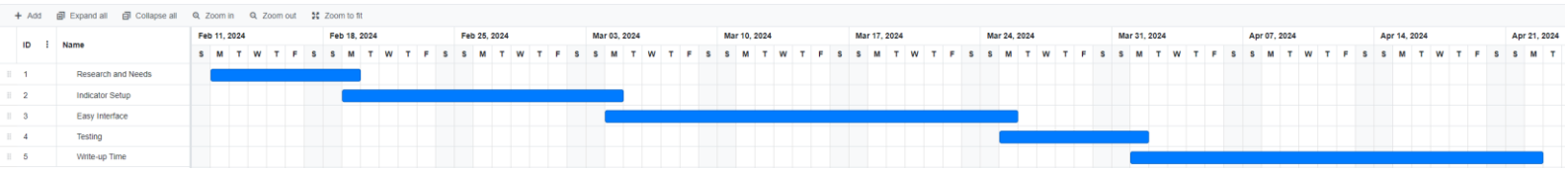
Data Sources:

To ensure analytical accuracy and reliable decision-making capabilities, the solution will consume real-time and historical market data from well-regarded financial data providers and APIs. The sourced data spans instruments such as foreign exchange and stocks the main one being the S&P 500, incorporating pricing, volume and other metrics. Access to high-quality, current data is necessary for modelling financial market dynamics.

Project Benefits

The trading bot project helps traders and investors by integrating multiple indicators for a comprehensive market analysis. Users benefit from stress reduction as the tool automates processes and provides clear understandings - fostering a more relaxed trading experience. Moreover, the project allows users to make decisions improving overall trading strategies and outcomes.

Work Plan:



Task	Duration	Start Date	End Date
Research and Needs	1 week	11/02/2024	18/02/2024
Indicator Setup	2 weeks	19/02/2024	03/03/2024
Easy Interface	3 weeks	04/03/2024	24/03/2024
Testing	1 week	25/03/2024	31/03/2024
Write-up Time	3 weeks	01/04/2024	21/04/2024

Simplified Plan:

1. **Research and Needs**

Check existing trading tools and indicators.

Figure out what the Python bot should do.

2. **Indicator Setup**

Make Algorithms for RSI, MACD, Bollinger Bands and others.

Make sure the bot makes accurate calculations using past and real-time data.

Add a Neural Network alongside the Indicators.

3. **Easy Interface**

Create a simple design for the bot.

Add visuals for key indicators and signals.

4. **Testing**

Try everything out to find and fix problems.

Check if indicators and automation work correctly.

5. **Write-up time**

Project Risks:

Risk	Description	Mitigation	Risk Score (1-5)
Lack of resources	Not having enough resources, such as consultants or team members, could impede progress.	Maintain a list of backup resources to address unexpected unavailability.	3
Changes in requirements	Changes in project requirements at vital stages may disrupt the project.	Conduct a complete analysis of requirements at the project's birth. Establish early recognition of potential requirement changes. Implement an agile process approach to accommodate evolving requirements.	4
Technical failures	Failures in technology, such as programming challenges or hardware failures, could hinder development.	Conduct a detailed assessment of chosen technologies to identify and address potential issues. Develop contingency plans for critical technology failures. Stay informed about technological advancements and updates.	4
Data quality issues	Inaccurate or incomplete market data could lead to damaged analysis and decision-making.	Apply data validation processes to confirm the accuracy and consistency of data. Often monitor data quality metrics and address any abnormalities promptly.	3
Regulatory changes	Changes in requirements may demand adjustments to the trading bot's functionality.	Stay updated on relevant changes and their possible impact on the project. Maintain flexibility in the bot's design to accommodate regulatory updates.	4

Risks that your project poses to others: Professional Ethics

The trading bot poses risks as users might overly depend on it, becoming less cautious about risks and potentially facing significant long-term losses. Automated trading by the bot also carries the risk of financial losses for users. Clear warnings and user education are necessary to prevent these negative outcomes. Focusing on these aspects is vital to promote responsible use, guaranteeing users don't solely rely on the bot's outputs given the uncertainties in financial markets.

Research Ethics Review Form for BSc and MSci

A		
1.1	Does your research require approval from the National Research Ethics Service (NRES)?	**NO**
1.2	Will you recruit participants who are covered by the Mental Capacity Act 2005?	**NO**
1.3	Will you recruit any participants who are covered by the Criminal Justice System, for example, people on remand, prisoners and those on probation?	**NO**
A		
2.1	Does your research involve participants who are unable to give informed consent? *For example, people who may have a degree of learning disability or mental health problem, that means they are unable to make an informed decision on their own behalf.*	**NO**
2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	**NO**
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	**NO**
2.4	Does your project involve participants disclosing information about protected characteristics (as identified by the Equality Act 2010)? *For example: racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings*	**NO**
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? *Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/*	**NO**
2.6	Does your research involve invasive or intrusive procedures? *These may include, but are not limited to, electrical stimulation, heat, cold or bruising.*	**NO**
2.7	Does your research involve animals?	**NO**
2.8	Does your research involve the administration of drugs, placebos or other substances to study participants?	**NO**
A		
3.1	Does your research involve participants who are under the age of 18?	**NO**
3.2	Does your research involve adults who are vulnerable because of their social, psychological or medical circumstances (vulnerable adults)? *This includes adults with cognitive and / or learning disabilities, adults with physical disabilities and older people.*	**NO**

3.3	Are participants recruited because they are staff or students of City, University of London? *For example, students studying on a particular course or module.* *If yes, then approval is also required from the Head of Department or Programme Director.*	**NO**
3.4	Does your research involve intentional deception of participants?	**NO**
3.5	Does your research involve participants taking part without their informed consent?	**NO**
3.5	Is the risk posed to participants greater than that in normal working life?	**NO**
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	**NO**
A		*Delete as appropriate*
4	Does your project involve human participants or their identifiable personal data? *For example, as interviewees, respondents to a survey or participants in testing.*	**NO**

PART B: Ethics Proportionate Review Form

B		
1.1.	Will you ensure that participants taking part in your project are fully informed about the purpose of the research?	**YES**
1.2	Will you ensure that participants taking part in your project are fully informed about the procedures affecting them or affecting any information collected about them, including information about how the data will be used, to whom it will be disclosed, and how long it will be kept?	**YES**
1.3	When people agree to participate in your project, will it be made clear to them that they may withdraw (i.e. not participate) at any time without any penalty?	**YES**
1.4	Will consent be obtained from the participants in your project? Consent from participants **MUST** be obtained if you plan to involve them in your project or if you plan to use identifiable personal data from existing records. "Identifiable personal data" means data relating to a living person who might be identifiable if the record includes their name, username, student id, DNA, fingerprint, address, etc. *If YES, you must attach drafts of the participant information sheet(s) and consent form(s) that you will use in section B.3 or, in the case of an existing dataset, provide details of how consent has been obtained.* *You must also retain the completed forms for subsequent inspection. Failure to provide the completed consent request forms will result in withdrawal of any earlier ethical approval of your project.*	**YES**
1.5	Have you made arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential?	**YES**

B		
2	Will the research be conducted in the participant's home or other non-University location? *If **YES**, you must provide details of how your safety will be ensured.*	**NO**

B	*YES*	*NO*	*Not Applicable*
Details on how safety will be assured in any non-University location, including risk assessment if required (see B2)			.
Details of arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential (see B1.5) *Any personal data must be acquired, stored and made accessible* *in ways that are GDPR compliant.*			.
Full protocol for any workshops or interviews**			.
Participant information sheet(s)**			.
Consent form(s)**			.
Questionnaire(s)** *sharing a Qualtrics survey with your supervisor is recommended.*			.
Topic guide(s) for interviews and focus groups**			.
Permission from external organisations or Head of Department** *e.g. for recruitment of participants*			.

9.2 Appendix B

Note: All of the codes in this section have been modified, and none exactly match the original.

Overview

This looks at the Python code that extracts and processes historical price data from the Binance API. This data is used in the LSTM and Linear Regression trading models discussed in Chapter 4/5 of this report. The Python script extracts closing prices and timestamps from a dataset of candlestick data taken from the Binance API. This data is used to interpret price movements and make trading decisions.

```
import numpy as np import pandas as pd from datetime import datetime

closing_prices = np.array([float(data[4]) for data in dataset]) # closing
prices for 1000 candles timestamps = np.array([int(data[0]) for data in
dataset]) # Timestamps for 1000 candles time_strings =
np.array([datetime.fromtimestamp(timestamp / 1000).strftime('%Y-%m-%d
%H:%M:%S') for timestamp in timestamps]) # Convert timestamps to readable
form for 1000 candles

df = pd.DataFrame({'Date': time_strings, 'Close': closing_prices}) # Create
df with timestamps and closing prices df['Date'] =
pd.to_datetime(df['Date']) # Convert 'Date' column to datetime format
df.set_index('Date', inplace=True) # Set 'Date' as index
```

Anahuac Strategies. (2021). "How to Extract Historical Price Data from Binance Using Python." Medium. Accessed April 18, 2024. Available online at: https://medium.com/@cemanahuacstrategies/how-to-extract-historical-price-data-from-binance-using-python-bface20aa394

Overview

The rate_limiter limits the number of API calls made by a Python application to external services. Limiting the number of calls sent per second supports compliance with API limits on use and reduces service overloading, which may end in blocked requests because of API provider rate limitations.

```
def rate_limiter(max_per_second):
    """
    Decorator function to limit the rate of API calls.
    Args:
        max_per_second (int): Maximum number of API calls allowed per
second.
    Returns:
        function: Decorated function with rate limiting.
    """
    min_interval = 1.0 / float(max_per_second)
    def decorate(func):
        last_called = [0.0]
        def rate_limited_function(*args, **kargs):
            elapsed = time.time() - last_called[0]
            left_to_wait = min_interval - elapsed
            if left_to_wait > 0:
                time.sleep(left_to_wait)
            last_called[0] = time.time()
            return func(*args, **kargs)
        return rate_limited_function
    return decorate
```

Real Python - "Primer on Python Decorators": https://realpython.com/primer-on-python-decorators/

Rate_limiter limits rates using the Python API. To meet API requirements and keep service speed high, this technique uses a wrapper to limit the number of function calls that can be made per second. There are online Python and API development tools that show you how to manage APIs and limit requests.

Overview

Extracts closing prices and timestamps from candlestick data, uses machine learning models to predict future prices, and transforms the predictions back to original values. The transformation and prediction logic were adapted to align with the preprocessing requirements of the LSTM and LR models used in this project, including configurations of the data reshaping and scaling procedures.

For Linear Regression

```
index = [996, 997, 998, 999] candles =
scaler.transform(np.array([float(dataset[i][4]) for i in index]).reshape(-
1,1))

model_feed = candles.reshape(1, 4) # reshape to (1, 4)

prediction = loaded_model.predict(model_feed)

prediction_reshaped = prediction.reshape(-1, 1)

predicted_price = scaler.inverse_transform(prediction_reshaped)[0][0]

print(predicted_price
```

For LSTM

```
index = [996, 997, 998, 999] candles =
scaler.transform(np.array([float(dataset[i][4]) for i in index]).reshape(-
1,1))

model_feed = candles.reshape(1,4,1)

(1, 4, 1) for prediction

prediction = loaded_model.predict(model_feed)

prediction_reshaped = prediction.reshape(-1, 1)

predicted_price = scaler.inverse_transform(prediction_reshaped)[0][0]

print(predicted_price)
```

Comet.ml Blog on Heartbeat *Analyzing and creating a predictive model for Binance data*. Available at: https://heartbeat.comet.ml/analyzing-and-creating-a-predictive-model-for-binance-data-69171dbd5bec?gi=e9efd42ad36c

Tavares, M. (n.d.) *Deep learning LSTM bitcoins*. Available at: https://github.com/marcotav/deep-learning/blob/master/bitcoin/notebooks/deep-learning-LSTM-bitcoins.ipynb.

Overview

We improved the code by changing factors like data separation and model architecture. We managed data division for training and testing and created a neural network with regularisation to prevent overfitting. Before setting a random seed for reproducibility, this code splits the dataset into training and testing data. A neural network model with LSTM layers and dropout regularisation follows. Then, it compiles the model and prints its architecture summary.

```
train_samples = 150 # Number of samples for training test_samples = 50 #
Number of samples for testing

x_train = df.iloc[:train_samples, :-1] y_train = df.iloc[:train_samples, -
1]

#Features and target column for testing data

x_test = df.iloc[train_samples:train_samples + test_samples, :-1] y_test =
df.iloc[train_samples:train_samples + test_samples, -1]

#Set random seed for reproducibility

np.random.seed(42)

#Define the model architecture

model = Sequential([ Bidirectional(LSTM(128, return_sequences=True,
input_shape=(x_train.shape[1], x_train.shape[2]),
kernel_regularizer=l2(0.001))), Dropout(0.3), LSTM(64,
return_sequences=True, kernel_regularizer=l2(0.001)), Dropout(0.3),
LSTM(32, kernel_regularizer=l2(0.001)), Dense(1) ])

optimizer = Adam(learning_rate=0.0005) model.compile(optimizer=optimizer,
loss='mean_squared_error')

model.build(input_shape=(None, x_train.shape[1], x_train.shape[2]))

print("Model Summary:")

model.summary()
```

Keras (n.d.) *LSTM*. Available at: https://keras.io/api/layers/recurrent_layers/lstm/ .

TensorFlow (n.d.) *Understanding LSTM Networks*. Available at: https://www.tensorflow.org/guide/keras/rnn .

Tavares, M. (n.d.) *Deep learning LSTM bitcoins*. Available at: https://github.com/marcotav/deep-learning/blob/master/bitcoin/notebooks/deep-learning-LSTM-bitcoins.ipynb.

Overview

This code creates a structured DataFrame from candlestick data closing prices and timestamps for analysis. In financial data, raw timestamp data is changed to a format humans can understand, and a DataFrame is created for candlestick charts.

```
# Extract closing prices and timestamps from candlestick data

closing_prices = np.array([float(data[4]) for data in dataset]) # closing
prices for 1000 candles timestamps = np.array([int(data[0]) for data in
dataset]) # Timestamps for 1000 candles time_strings =
np.array([datetime.fromtimestamp(timestamp / 1000).strftime('%Y-%m-%d
%H:%M:%S') for timestamp in timestamps]) # Convert timestamps to readable
form for 1000 candles

# Create a DataFrame for the candlestick chart

df = pd.DataFrame({'Date': time_strings, 'Close': closing_prices}) # Create
df with timestamps and closing prices df['Date'] =
pd.to_datetime(df['Date']) # Convert 'Date' column to datetime format
pdf.set_index('Date', inplace=True) # Set 'Date' as index
```

Raza, H. (n.d.) *Trading Bot LSTM*. Available at: https://github.com/HassanRaza1313/Bitcoin-Trading-Bot-Using-LSTM

Overview

The code simulates Bitcoin trading strategies. A trading balance is set up, and price predictions are used to make predictive trading decisions. The logic decides when to buy or sell Bitcoin, updating the balance and recording transaction time and type. The structure allows testable trading strategy simulation.

```
initial_balance = 10000

balance = initial_balance btc_held = 0

transaction_times = [] # Store transaction times transaction_types = []
#Store transaction types

for i in range(len(y_pred)): current_price = y_test_inv[i][0]
predicted_price = y_pred_inv[i][0] transaction_time = btc_data_15m.index[i]

`if predicted_price > current_price and balance > 0:

btc_held += balance / current_price

balance = 0

print(f"Bought BTC at {current_price} at: {transaction_time}")
transaction_times.append(transaction_time) transaction_types.append('Buy')

elif predicted_price < current_price and btc_held > 0:

balance += btc_held * current_price

btc_held = 0

print(f"Sold BTC at {current_price} at: {transaction_time}")
transaction_times.append(transaction_time)
transaction_types.append('Sell')`
```

Araujo, A. (n.d.) *BTC backtesting*. Available at: https://github.com/araujo88/BTC-backtesting

9.3 Appendix C

Date	Duration	Discussion
07/02/2024	15 minutes	We discussed the project and the first thoughts that came to mind. We also discussed the overall specs and problems and an introduction to using the trading bot to enter the Bitcoin market.
07/03/2024	25 minutes	The literature that was important to the project was reviewed. As part of the literature review, we spoke about the number of required papers and the available sources.
22/03/2024	25 minutes	The methods part was reviewed to find any missing information and discuss the need to make things more transparent and flow better. Plans were made to make these changes to build a solid methodological framework.
11/04/2024	45 minutes	We have reviewed the changes and upgrades to the code. We concentrated on the layout and the most essential points to fix when we went over the Results section.
26/04/2024	60 minutes	The final draft of the project was discussed, including its structure and its content. Important choices were made about the paper's structure to ensure that each section flows smoothly into the next and that all important ideas are correctly presented. To improve the overall appearance ensuring that the paper is ready for submission.