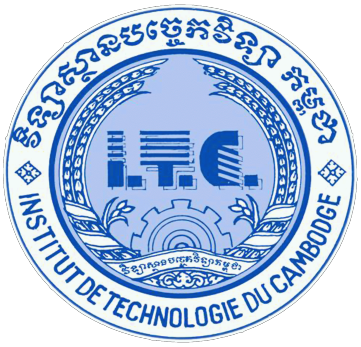

AIIS
Department of Applied Mathematics and Statistics

Institute of Technology of Cambodia

Department of Applied Mathematics and Statistics

Bitcoin Price Prediction Using LSTM and XGBoost

Academic Year: 2024-2025

4th Year Engineering Degree in Data Science

Group Members:

Chhon Menghout	e20211474
Dok Dominique	e20210337

Lecturers:

Mr. Sok Kimheng	(Course)
Dr. Neang Pheak	(TP)
Dr. Phauk Sokkhey	(TP)

Bitcoin Price Prediction Using LSTM and XGBoost

Chhon Menghout[1*], Dok Dominique[2*], Neang Pheak[3], Phauk Sokkhey[3], Sok Kimheng[3]

[1,2,3] Department of Applied Mathematics and Statistics,
Institute of Technology of Cambodia, Cambodia

Academic Year: 2024-2025

Abstract

This study investigates the prediction of Bitcoin prices using machine learning techniques, specifically comparing the Long Short-Term Memory (LSTM) neural network and the XGBoost model. Bitcoin's inherent volatility and the complexity of its market dynamics present substantial challenges for accurate forecasting, often exceeding the limitations of traditional time-series models. In response to these challenges, the study explores the application of advanced deep learning methods like LSTM and robust gradient-boosting algorithms like XGBoost. By leveraging historical Bitcoin price data, the data was pre-processed, transformed into temporal windows, and subsequently used to train and test both models. A comprehensive evaluation was performed using various performance metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Scaled Error (MASE), providing a detailed analysis of the prediction accuracy. The results highlight LSTM's superior ability to capture long-term dependencies and model non-linear relationships, while XGBoost, with its powerful feature-based learning, delivers competitive and reliable predictions. This study emphasizes the effectiveness of combining deep learning and gradient boosting techniques in addressing the challenges of predicting Bitcoin prices and offers insights into their applicability for other prediction tasks.

Keywords: Bitcoin price prediction, machine learning, LSTM, XGBoost, time-series forecasting, deep learning, gradient boosting, predictive modeling, market analysis, volatility.

Contents

List of Figures

List of Tables

Acknowledgement

I would like to extend my heartfelt gratitude to the Department of Applied Mathematics and Statistics at the Institute of Technology of Cambodia for their invaluable support, guidance, and resources throughout this project. Their expertise and encouragement have been instrumental in shaping this research and bringing it to fruition. I am particularly grateful to my professors and mentors, whose insights have guided me at every stage. Furthermore, I wish to express my deepest appreciation to my family for their generous financial support, without which this project would not have been possible. Their commitment to my education and unwavering encouragement have provided me with the stability and motivation necessary to pursue this research with dedication and purpose. This project is a testament to the collaborative spirit and support of both my academic institution and my family, and I am sincerely thankful for their contributions. This research would never have been done completely without these important supporters.

1 Introduction

1.1 Background of Project

Bitcoin is an electronic currency that has become increasingly popular since its introduction in 2008. Transactions in the Bitcoin system are stored in a public transaction ledger (the blockchain), which is stored in a decentralized peer-to-peer network. Bitcoin provides decentralized currency issuance and transaction clearance. The security of the blockchain depends on a compute-intensive algorithm for Bitcoin mining, which prevents double spending of bitcoins and tampering with confirmed transactions.

1.2 Statement of Problem

Existing Bitcoin price prediction models, particularly traditional approaches, often fail to capture the complexity and nonlinearity inherent in Bitcoin's price behavior, leading to inaccurate forecasts. This study aims to address these challenges by comparing Long Short-Term Memory (LSTM) and XGBoost models, providing a comprehensive evaluation of their forecasting capabilities.

1.3 Purpose of Study

The study begins with data manipulation, which involves collecting, preprocessing and analyzing historical Bitcoin price data to ensure that it is properly formatted and organized for training and evaluation. Next, prediction models are developed and trained using LSTM and XGBoost, leveraging their strengths in processing sequential time-series data to effectively capture Bitcoin's complex price trends and high volatility patterns. Finally, model performance is evaluated through a thorough comparative analysis, benchmarking the models' forecasting accuracy by assessing their performance using key metrics and visualizing the results to validate each model's effectiveness.

1.4 Scope and Limitation

1.4.1 Scope

This study focuses on prediction of Bitcoin price using machine learning techniques. It involves the development and evaluation of the LSTM and XGBoost models based on historical price data. We have collected the historical price of Bitcoin from 2012 to 2024. This data contains 4707 rows and columns such as Start and End (dates), Close, Open, High, and Low (Bitcoin prices), Volume, and Market Cap.

1.4.2 Limitation

The accuracy of the model is highly dependent on the quality and availability of historical Bitcoin data. Incomplete datasets, missing values, or external anomalies can negatively impact the performance of the model and the accuracy of the forecast. The unpredictable nature of cryptocurrency markets poses a challenge for the model. Extreme price fluctuations or sudden market shocks, such as regulatory changes or macroeconomic events, can lead to unpredictable shifts in price trends, reducing the model's ability to forecast accurately during such volatile periods.

2 Literature Review

Several studies have explored the prediction of Bitcoin prices using machine learning models. Vinay Karnati, Lakshmi Dathatreya Kanna, and Trilok Nath Pandey studied the performance of Facebook Prophet, ARIMA, SVM, and LSTM algorithms for time-series forecasting and training. Their methods included pre-processing the data, running the algorithms, and evaluating their efficacy with appropriate metrics. Data were collected from the Yahoo Finance website, dated from 2013 to 2021. According to their report, LSTM had the best performance, followed by Prophet, SVM, and ARIMA, respectively.

Junwei Chen developed an algorithmic model with high predictive accuracy for the next-day price of Bitcoin using random forest regression and LSTM. The study also aimed to identify the variables that influence Bitcoin prices. The Diebold-Mariano test did not conclusively show that random forest regression outperforms LSTM in prediction accuracy, but the RMSE and MAPE errors for random forest regression were lower than those for LSTM. The study used eight categories (47 variables) as explanatory variables: Bitcoin price variables, specific technical features of Bitcoin, other cryptocurrencies, commodities, market indices, foreign exchange, public attention, and dummy variables of the week.

Yaowen Hu performed Bitcoin price prediction using various machine learning models, including Support Vector Machine (SVM), Random Forest, Neural Network, XGBoost, and LightGBM. The Bitcoin price dataset was divided into training and test sets in a ratio of 7:3. By comparing the MSE, RMSE, MAE, MAPE, and R^2 of the different models, it was found that XGBoost had the best performance and prediction accuracy. The performance of the other four models ranged from good to poor, with LightGBM, Random Forest, SVM, and Neural Network following in descending order of accuracy.

Yangyu Chen's case study provides an in-depth examination of Bitcoin, exploring its origins, technological foundations, market behavior, trading strategies, and regulatory considerations. The study highlights Bitcoin's transformative journey from its inception to its current role as a dynamic digital asset, emphasizing the revolutionary impact of blockchain technology.

Saber Talazade and Dragan Peraković introduced a novel approach to stock market prediction by integrating sentiment analysis with the traditional Random Forest model. This new methodology, termed "Sentiment-Augmented Random Forest" (SARF), leverages the nuanced understanding of financial sentiments provided by the FinGPT generative AI model to optimize the accuracy of stock price forecasts. The proposed SARF technique incorporates sentiment features into the Random Forest framework. Experiments demonstrated that SARF outperforms conventional Random Forest and LSTM models, with an average accuracy improvement of 9.23% and lower prediction errors in predicting stock market movements.

Yao Yao's study, "Predicting Stock Prices Using The RF-LSTM Combination Model," addresses the challenge of predicting stock prices given the nonlinearity and complexity of stock data. The study proposes a combined Random Forest (RF) and Long-Short Term Memory (LSTM) model to forecast stock closing prices. The results showed that the RF-LSTM combined model outperformed the single LSTM neural network model, with reductions in root mean squared error (RMSE) and mean squared error (MSE) of 4.14% and 1.00%, respectively.

3 Data Preprocessing

3.1 Data Collection

The data is collected from the CoinCodex website, dated from 2012 to 2024. This data contains attributes such as date, highest price, lowest price, opening price, closing price, volume, and market cap on those particular days.

3.2 Data Cleaning

Data cleaning refers to the process of handling missing values and removing erroneous, incomplete, or inaccurate data from time series datasets. This involves ensuring the data is reliable and ready for analysis. When dealing with missing values, one can either remove them entirely or apply imputation techniques, such as replacing them with the mean, median, or mode of the dataset.

3.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial initial step in data science projects. It involves analyzing and visualizing data to understand its key characteristics, uncover patterns, and identify relationships between variables. EDA is typically carried out as a preliminary step before undertaking more formal statistical analyses or modeling.

Table 1: Descriptive Statistics

	Open	High	Low	Close	Volume	Market Cap
count	4707.000000	4707.000000	4707.000000	4707.000000	4.707000e+03	4.707000e+03
mean	14791.122668	15125.236917	14448.993594	14812.040377	2.608945e+10	2.806215e+11
std	19774.971069	20211.038246	19339.952351	19808.898442	3.684074e+10	3.841040e+11
min	4.222000	4.222000	4.222000	4.222000	0.000000e+00	3.525001e+07
25%	416.477000	421.687500	410.237500	416.583500	2.616035e+07	5.788002e+09
50%	6328.687636	6434.372794	6232.734067	6330.669843	3.950140e+09	1.099704e+11
75%	23595.945829	24158.389102	23120.778738	23650.821667	4.314177e+10	4.502643e+11
max	91135.390000	93836.940000	90406.060000	92119.470000	2.121958e+11	1.818537e+12

Insights and observations from the data analysis reveal several key trends. First, the large standard deviations in the Open, High, Low, and Close prices highlight Bitcoin's significant price volatility over time, reflecting the unpredictable nature of the cryptocurrency market. Additionally, anomalies such as the minimum values of Volume (0) and Market Cap ($\tilde{}$35.25 million) may point to data entry errors, missing data for specific time periods, or legitimate instances of low activity, especially in Bitcoin's early history. Furthermore, the 25th and 75th percentiles indicate substantial growth over time, with prices shifting from hundreds of dollars ($416) in the early stages to tens of thousands of dollars ($23,650) in later periods, illustrating the cryptocurrency's dramatic rise in value.

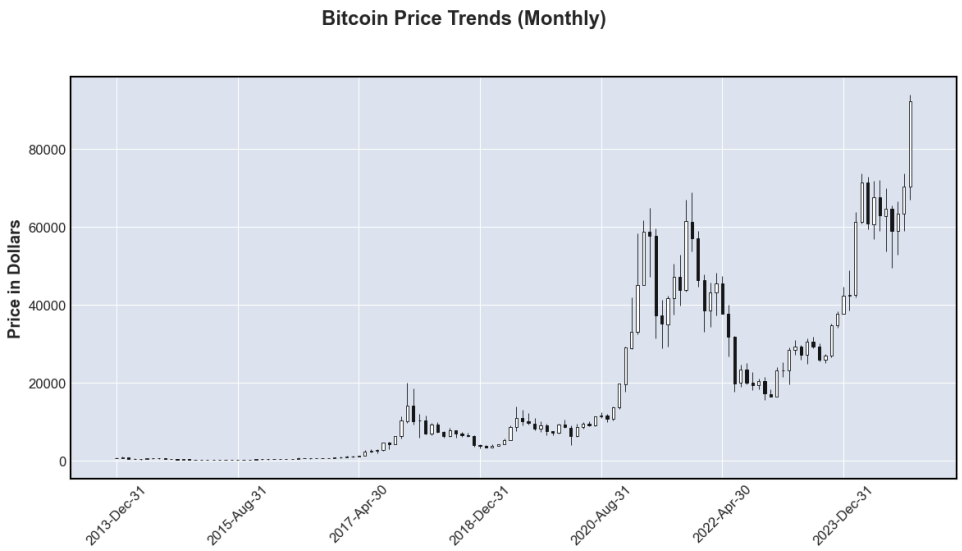

Figure 1: Time-series chart of Bitcoin Price

The chart illustrates several key observations regarding Bitcoin's price trends. First, a long-term growth trend is evident, particularly after 2017, suggesting an increase in market adoption, speculative interest, and potential use as an inflation hedge. The chart also highlights periods of significant volatility, characterized by sharp price increases followed by steep declines, such as the rapid surge and subsequent correction between late 2017 and early 2018, and another major price jump from mid-2020 to 2021, followed by a subsequent correction. In addition, recent peaks in Bitcoin's price have seen all-time highs surpassing $90,000, likely influenced by broader market events, institutional investor adoption, and macroeconomic factors. Furthermore, the chart reflects the market's growing maturity over time, as earlier periods feature relatively stable prices, while later periods are marked by extreme fluctuations, indicating increasing investor participation and market evolution. Overall, this visualization underscores the importance of considering market volatility and external influences when developing predictive models for Bitcoin prices.

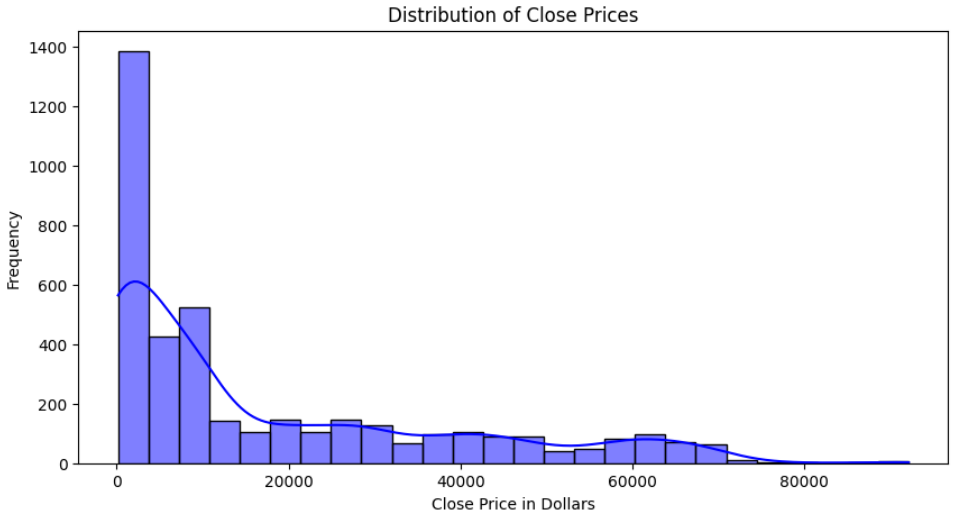

Figure 2: Close Price Distribution

The histogram provides insights into the distribution of Bitcoin's closing prices during the analyzed period. The data shows a highly skewed distribution with a strong positive skew, where the majority of closing prices are concentrated in the lower price range, particularly below $20,000. This suggests that Bitcoin's price has predominantly remained at lower levels throughout the period. As the closing prices rise, the frequency of occurrences decreases significantly, indicating that periods of higher Bitcoin prices, such as above $40,000, were less frequent. Additionally, the long tail of the distribution reflects occasional price spikes, with rare instances reaching $60,000 or higher, likely driven by major market events or speculative trends. These occasional high prices might represent potential outliers, highlighting periods of exceptional market activity that warrant further exploration.

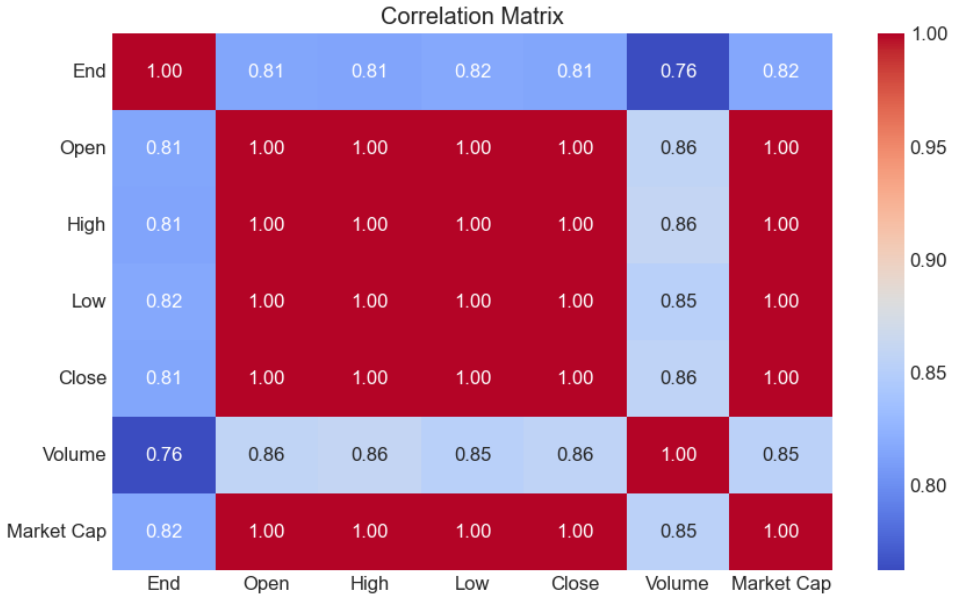

Figure 3: Correlation Matrix

The heatmap reveals important insights into the relationships between Bitcoin's price and market metrics. It shows strong correlations among the Open, High, Low, and Close prices, which are perfectly correlated (correlation coefficient = 1.00), as these values typically move together within a given time frame. Market Capitalization also demonstrates a perfect correlation (1.00) with the price variables, reflecting that it is inherently derived from these metrics. On the other hand, Volume has a moderate correlation with the End price (0.76), suggesting that trading volume does not always align perfectly with price movements. The color gradient in the heatmap helps interpret these relationships, with darker red areas indicating higher correlations and blue areas, such as those related to Volume, representing weaker associations. This visualization emphasizes the need for careful feature selection in predictive modeling to avoid multicollinearity, especially with highly correlated variables like Open, High, Low, and Close.

4 Model Implementation: Time Series Forecasting

4.1 Data Preparation

4.1.1 Windowing Dataset

Windowing transforms a time series dataset into a supervised learning problem by creating labeled windows of historical data to predict future values. The horizon is the number of future steps to predict (set to 1), and the window size is the number of past timesteps used for prediction (set to 7).

4.1.2 Creating Windows

The make_windows function was implemented to convert a 1D array of time series data into sequential windows of size WINDOW_SIZE. The output included both windows (inputs) and their corresponding labels (targets).

4.1.3 Train-Test Split

The dataset was split into training and test sets using an 80-20 split: Training Windows and Test Windows are 3,760 and 940, respectively.

4.2 Model Development

4.2.1 LSTM

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) architecture that is effective in handling the issue of vanishing gradients in traditional RNNs. The LSTM model consists of three main layers: the input layer, the LSTM layer, and the output layer.

In this implementation, the LSTM layer uses a **ReLU activation function** instead of the traditional tanh for the candidate cell state computation. This modification allows the model to better capture non-linear patterns in the data.

The LSTM model is trained using historical Bitcoin price data as input sequences and the corresponding target values. The objective is to capture long-term dependencies and temporal patterns in the data. The optimization algorithm used is Adam, and the loss function is Mean Absolute Error (MAE):

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

where y_i is the actual value and $\hat{y}_i$ is the predicted value. The architecture of the LSTM model is as follows:

- **Input Layer**: Takes input sequences of shape $(WINDOW\_SIZE, 1)$, where $WINDOW\_SIZE$ is the number of time steps in the input sequence.
- **LSTM Layer**: Consists of 128 units with ReLU activation for the candidate cell state. The gates (input, forget, and output) use sigmoid activation.
- **Output Layer**: A Dense layer with $HORIZON$ units, where $HORIZON$ is the number of time steps to predict.

The model is compiled with the Adam optimizer and MAE loss function. During training, a custom callback is used to print the performance at the end of each epoch, and a model checkpoint callback is used to save the best model based on validation loss.

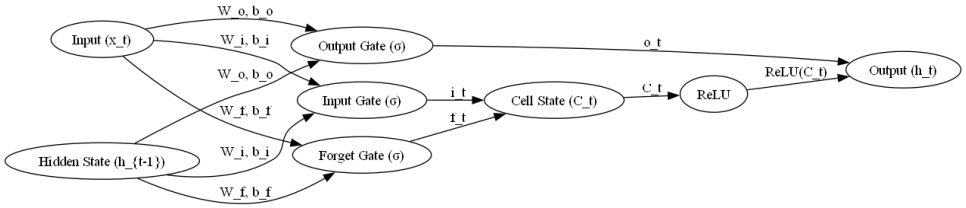

Figure 4: Architecture of the LSTM model with ReLU activation. The diagram shows the input layer, LSTM layer (with input, forget, and output gates), and output layer.

4.2.2 Mathematical Details of LSTM Gates

The LSTM cell operates using the following equations:

1. Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

where:

- i_t is the input gate activation at time t,
- σ is the sigmoid activation function,
- W_i is the weight matrix for the input gate,
- h_{t-1} is the hidden state at time $t-1$,
- x_t is the input at time t,
- b_i is the bias for the input gate.

2. Forget Gate:

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

where:

- f_t is the forget gate activation at time t,
- W_f is the weight matrix for the forget gate,
- b_f is the bias for the forget gate.

3. Output Gate:

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

where:

- o_t is the output gate activation at time t,
- W_o is the weight matrix for the output gate,
- b_o is the bias for the output gate.

4. Cell State Update:

$$\tilde{C}_t = \text{ReLU}(W_C \cdot [h_{t-1}, x_t] + b_C)$$

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

where:

- $\tilde{C}_t$ is the candidate cell state at time t,
- C_t is the updated cell state at time t,
- W_C is the weight matrix for the cell state,
- b_C is the bias for the cell state,
- ReLU is the Rectified Linear Unit activation function, defined as $\text{ReLU}(x) = \max(0, x)$.

5. Hidden State Update:

$$h_t = o_t \cdot \text{ReLU}(C_t)$$

where:

- h_t is the hidden state at time t,
- ReLU is applied to the cell state C_t before being gated by the output gate o_t.

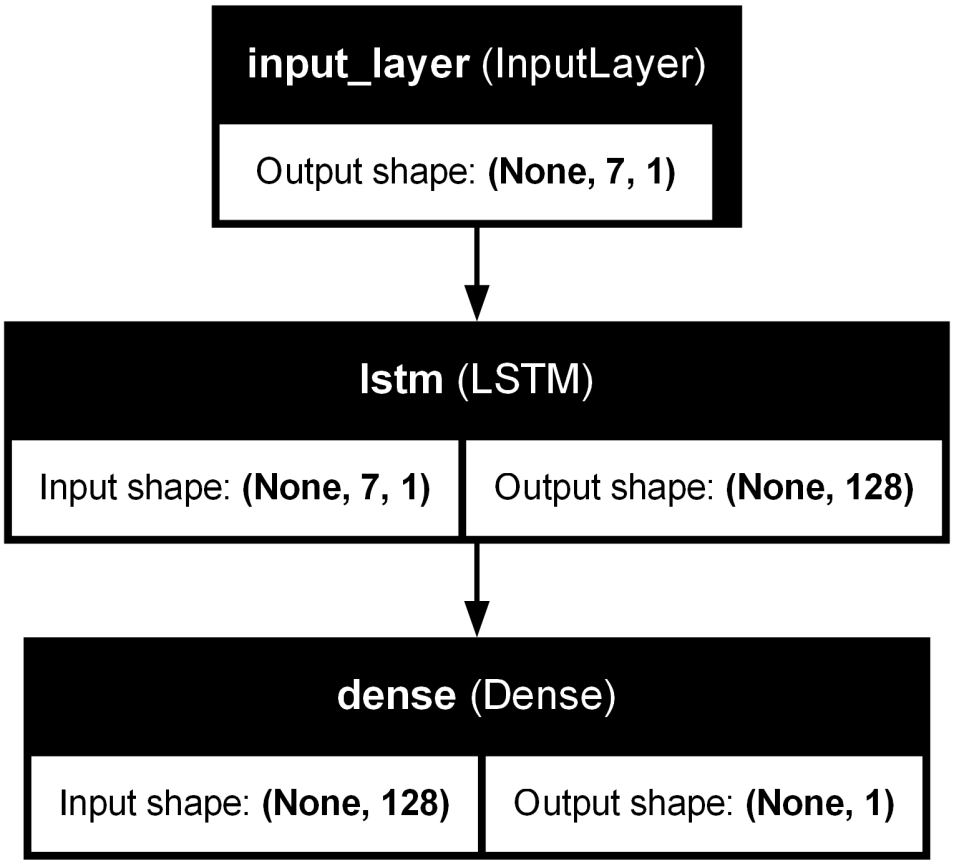

Figure 5: LSTM Model Architecture

4.2.3 XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework. The XGBoost model is trained using historical Bitcoin

price data, and the objective function used is 'reg:squarederror', which is appropriate for regression tasks.

The loss function for XGBoost is the Mean Squared Error (MSE):

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

where y_i is the actual value and $\hat{y}_i$ is the predicted value.

4.2.4 Model Architecture of XGBoost

XGBoost builds an ensemble of decision trees in a sequential manner, where each tree corrects the errors of the previous one. The model is trained using gradient boosting, which minimizes the loss function by iteratively adding trees that predict the residuals of the previous trees.

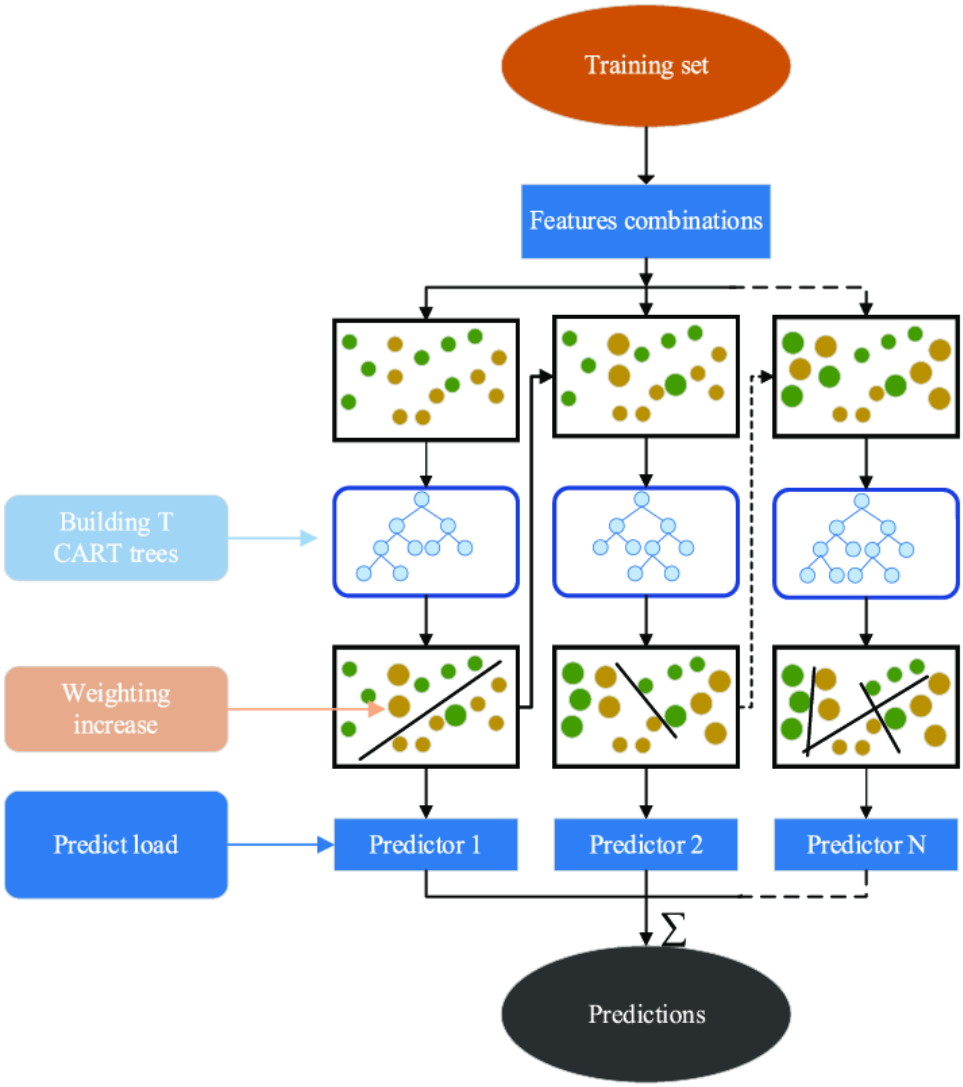

Figure 6: Architecture of the XGBoost model.

4.2.5 Mathematical Details of XGBoost

XGBoost uses gradient boosting to optimize the model. The key steps are:

1. Model Prediction:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

where:

- $\hat{y}_i$ is the predicted value for the i-th instance,
- f_k is the k-th tree in the ensemble,
- $\mathcal{F}$ is the space of regression trees.

2. Objective Function:

$$\text{Obj}(\theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where:

- $L(y_i, \hat{y}_i)$ is the loss function (e.g., MSE),
- $\Omega(f_k)$ is the regularization term to control model complexity.

3. Gradient Boosting: At each iteration t, the model adds a new tree f_t to minimize the objective:

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

where:

- $\hat{y}_i^{(t-1)}$ is the prediction from the previous iteration.

4. Tree Splitting: The tree is grown by splitting nodes to maximize the gain:

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma$$

where:

- g_i and h_i are the first and second derivatives of the loss function,
- I_L and I_R are the instance sets for the left and right child nodes,
- λ and γ are regularization parameters.

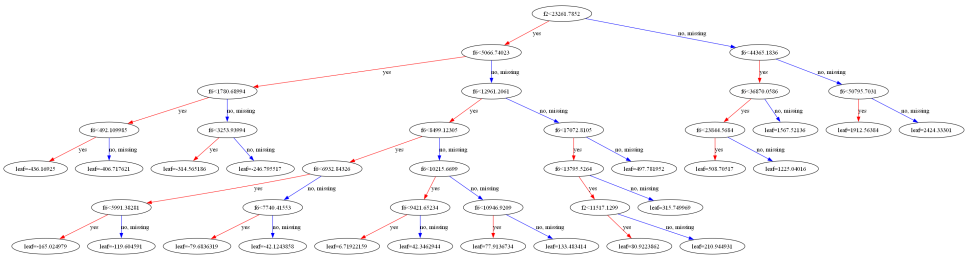

Figure 7: XGBoost Model Architecture

5 Result and Analysis

5.1 Evaluation Metrics

The following evaluation metrics were used to assess the performance of the models:

- **Mean Absolute Error (MAE)**:

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

- **Mean Squared Error (MSE)**:

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

- **Root Mean Squared Error (RMSE)**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- **Mean Absolute Percentage Error (MAPE)**:

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- **Mean Absolute Scaled Error (MASE)**:

$$\text{MASE} = \frac{\text{MAE}}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|}$$

5.2 LSTM Results

The LSTM model demonstrated high accuracy in capturing temporal dependencies and patterns within the data. The model's predictions closely followed the actual training data, indicating strong performance during training. On the test data, the model successfully generalized to unseen data, showing robust performance.

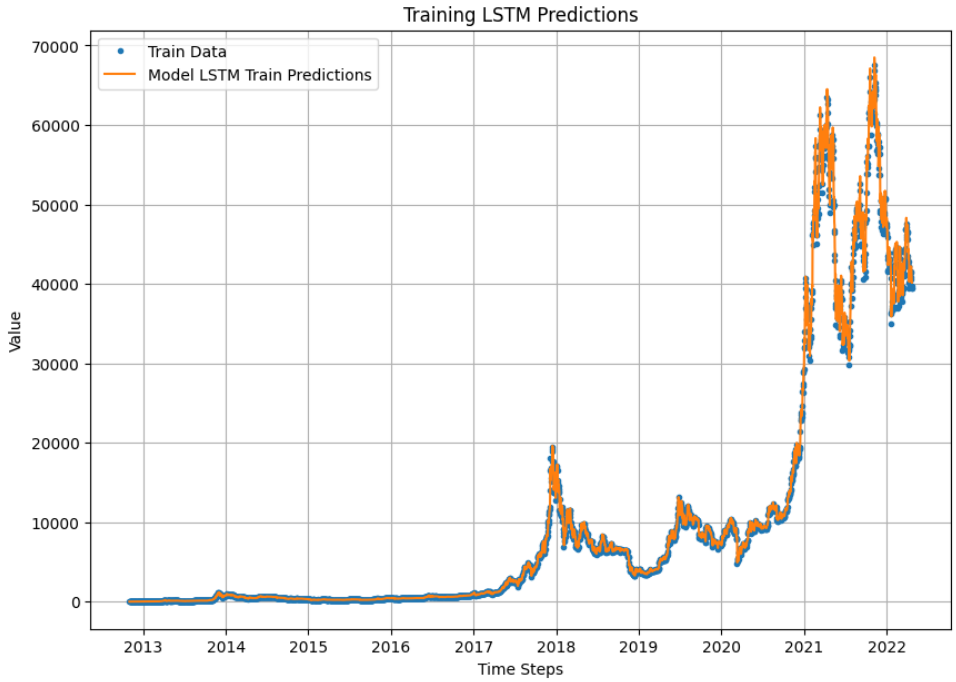

Figure 8: LSTM Training Performance

The orange line (model predictions) closely follows the blue line (actual training data) across the 11 years. This indicates the model's strong ability to capture trends and minimize errors during training.

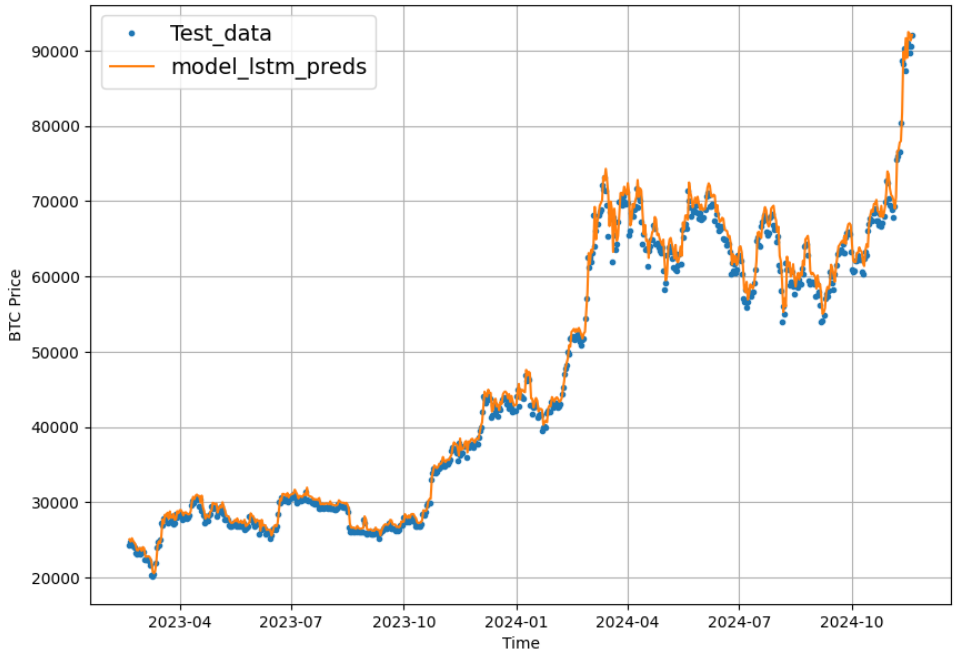

Figure 9: LSTM Testing Performance

The predicted values (orange line) align well with the test data (blue points) from 2023 to 2024. The model successfully generalizes to unseen data, indicating robust performance. Strengths of LSTM: Recurrent architecture allows it to model temporal de-

pendencies. High accuracy in capturing trends and sudden changes in volatile data, as evident in the testing phase.

5.3 XGBoost Results

The XGBoost model performed well on the training data, with predictions closely matching actual values. However, on the test data, the predictions showed some deviations, suggesting room for improvement in generalization.

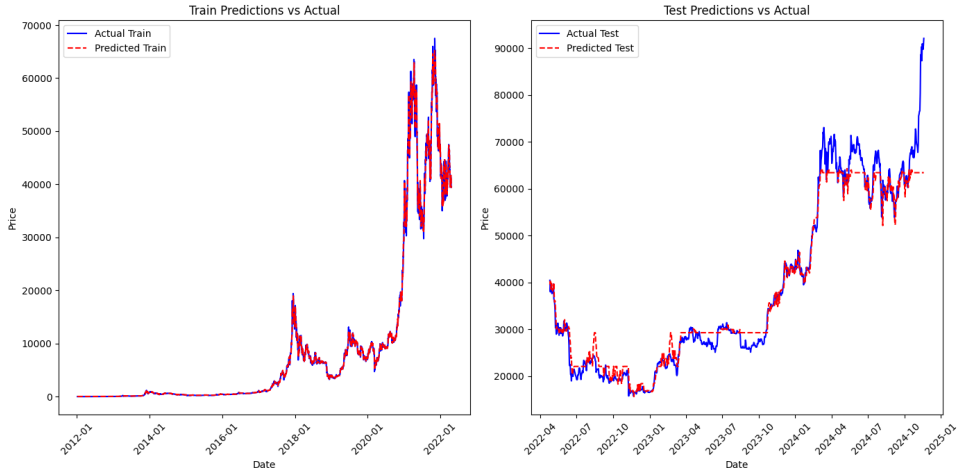

Figure 10: XGBoost Training vs. Testing Predictions

Training vs. Testing Predictions:

Left Panel (Training), the red dashed line (predicted prices) aligns closely with the blue line (actual prices), indicating accurate fitting, while the Right Panel (Testing), the red dashed line (predictions) follows the overall trend of the blue line but struggles slightly during periods of high volatility, showing minor deviations.

5.4 Model Performance Comparison: LSTM vs. XGBoost

The LSTM model significantly outperformed the XGBoost model across all evaluation metrics, including MAE, MSE, RMSE, MAPE, and MASE. The LSTM's ability to model temporal dependencies made it especially effective in capturing trends and sudden changes in volatile data.

Table 2: Metrics Comparison

Metric	LSTM	XGBoost	Comparison
MAE	744.74	2107.53	LSTM demonstrates significantly lower MAE
MSE	1,442,230.8	13,777,630.0	LSTM achieves much lower MSE
RMSE	1,200.93	3,711.82	LSTM has substantially lower RMSE
MAPE	1.89%	5.48%	LSTM outperforms XGBoost in MAPE
MASE	1.0025	2.837	LSTM has lower MASE

6 Conclusion and Future Work

6.1 Conclusion

In this study, we compared two machine learning models, Long Short-Term Memory (LSTM) and XGBoost, for univariate time series forecasting using historical Bitcoin price data. The results highlighted several key insights regarding model performance. The LSTM model significantly outperformed XGBoost across all evaluation metrics, including MAE, MSE, RMSE, MAPE, and MASE. LSTM's ability to model temporal dependencies made it especially effective in capturing trends and sudden changes in volatile data, which is critical for financial time series forecasting.

6.2 Future Work

To further enhance and expand upon this research, several directions are proposed. First, feature engineering could be explored by incorporating additional variables such as trading volume, sentiment analysis from social media, macroeconomic indicators, and market sentiment data to enrich model inputs and potentially improve forecasting accuracy. Additionally, hybrid models combining the strengths of LSTM and XGBoost could be investigated to leverage both temporal modeling capabilities and feature interaction strengths.

References

1. Sustainability of bitcoin and blockchains - ScienceDirect.
https://www.sciencedirect.com/science/article/abs/pii/S1877343517300015

2. tf.keras.layers.LSTM — TensorFlow v2.16.1.
https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM

3. XGBoost Python Package — xgboost 2.1.3 documentation.
https://xgboost.readthedocs.io/en/latest/python/index.html

4. Prediction and Analysis of Bitcoin Price using Machine learning and Deep learning models — EAI Endorsed Transactions on Internet of Things.
https://publications.eai.eu/index.php/IoT/article/view/5379

5. Analysis of Bitcoin Price Prediction Using Machine Learning.
https://www.mdpi.com/1911-8074/16/1/51

6. Bitcoin Price Prediction Based on Multiple Machine Learning Algorithms.
https://www.researchgate.net/publication/380180701_Bitcoin_Price_Prediction_Based_on_Multiple_Machine_Learning_Algorithms

7. Understanding Bitcoin: A Case Study Method to Understand Market Dynamics, Strategies, and Risks of Cryptocurrency.
https://www.researchgate.net/publication/376887854_Understanding_Bitcoin_A_Case_Study_Method_to_Understand_Market_Dynamics_Strategies_and_Risks_of_Cryptocurrency

8. SARF: Enhancing Stock Market Prediction with Sentiment-Augmented Random Forest.
https://www.researchgate.net/publication/384811602_SARF_Enhancing_Stock_Market_Prediction_with_Sentiment-Augmented_Random_Forest

9. Predicting Stock Prices Using The RF-LSTM Combination Model.
https://www.researchgate.net/publication/383230619_Predicting_Stock_Prices_Using_The_RF-LSTM_Combination_Model

10. Bitcoin Historical Data - CoinCodex.
https://coincodex.com/crypto/bitcoin/historical-data/

11. What is Exploratory Data Analysis? - GeeksforGeeks.
https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/

12. Bitcoin price prediction using ARIMA and LSTM — E3S Web of Conferences.
https://www.e3s-conferences.org/articles/e3sconf/abs/2020/78/e3sconf_iseese2020_01050/e3sconf_iseese2020_01050.html

13. Pandey, T.N., Priya, T., Jena, K.: Prediction of Exchange rate in a cloud computing environment using machine learning tools. Intell. Cloud Comput. 137–146 (2021).
https://www.bing.com/search?pglt=299&q=Pandey%2C+T.N.%2C+Priya%2C+T.%2C+Jena%2C+K.%3A+Prediction+of+Exchange+rate+in+a+cloud+computing+environment+

using+machine+learning+tools.+Intell.+Cloud+Comput.+137%E2%80%93146+(2021)
.&cvid=135945fe8d1c48ffaff1c824a648dbf6&gs_lcrp=EgRlZGdlKgYIABBFGDkyBggAEEUYOdIBB
FORM=ANNTA1&PC=U531

14. Samiksha Marne, Shweta Churi, Delisa Correia, Joanne Gomes, 2021, Predicting Price of Cryptocurrency – A Deep Learning Approach, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NTASU – 2020 (Volume 09 – Issue 03).
https://www.bing.com/search?q=Samiksha+Marne%2C+Shweta+Churi%2C+Delisa+
Correia%2C+Joanne+Gomes%2C+2021%2C+Predicting+Price+of+Cryptocurrency+
%E2%80%93+A+Deep+Learning+Approach%2C+INTERNATIONAL+JOURNAL+OF+ENGINEERING+
RESEARCH+%26+TECHNOLOGY+(IJERT)+NTASU+%E2%80%93+2020+(Volume+09+%E2%80%
93+Issue+03)%2C%27&cvid=b255a44f58034dca9cfde4bb2682b572&gs_lcrp=EgRlZGdlKgYIABBF
FORM=ANAB01&PC=U531

15. Prediction of Bitcoin Price Based on LSTM — Semantic Scholar.
https://www.semanticscholar.org/paper/Prediction-of-Bitcoin-Price-Based-on-LSTM-
76c017e5d09065f481d926bbb9bf56b3ce92717f

16. XGBoost Documentation — xgboost 2.1.3 documentation.
https://xgboost.readthedocs.io/en/latest/index.html

Appendix

EDA

The exploratory data analysis (EDA) can be found at the following link: `https://drive.google.com/file/d/1fvH3WkpuEr7C5cQdwJLnpRUag2OkJ-Pb/view?usp=sharing`

Models

The trained models can be accessed at the following link: `https://drive.google.com/file/d/1J6b0n_EmEiMMza5korShGwq_zi2BdxMW/view?usp=sharing`