# Project Report: TimeSeriesAI: Bitcoin Price Prediction

# Submitted By:

# Noman Asif Khan(22k-5007)

Maaz Siddiqui(22k-5116)

# Course: AI

# Instructor: Ms. Almas Ayesha Ansari

# Submission Date: 9-5-2025

# Abstract

This project investigates time series forecasting methods, specifically ARIMA and SARIMA models, for predicting Bitcoin closing prices. Bitcoin, a volatile and decentralized digital currency, presents unique challenges for forecasting due to high price fluctuations and seasonal trends. The primary goal is to build effective forecasting models by analyzing historical data, ensuring stationarity, and evaluating the predictive performance. We use Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) as our key metrics. This report thoroughly discusses data preparation, model fitting, parameter tuning, diagnostic checks, and forecasting outcomes.

# Table of Contents

1. Introduction  
2. Data Description  
3. Methodology  
4. Implementation  
5. Results & Evaluation  
6. Conclusion  
7. References

# 1. Introduction

Bitcoin has rapidly gained traction as a global decentralized currency. Its high volatility makes predicting price movements both challenging and essential for investors and researchers. Time series forecasting helps model and predict future values based on historical trends. In this project, we implement ARIMA (AutoRegressive Integrated Moving Average) and SARIMA (Seasonal ARIMA) to capture both non-seasonal and seasonal components of the time series. The report aims to evaluate model performance and discuss which model better adapts to Bitcoin’s complex behavior.

# 2. Data Description

The dataset used is sourced from Yahoo Finance and contains historical daily closing prices of Bitcoin from January 2023 to April 2025. The primary focus is on the 'Date' and 'Close' columns.  
  
Key Characteristics:  
- Rows: 820  
- Columns: Date, Open, High, Low, Close, Volume  
- Missing Values: Handled using forward-fill technique  
An initial plot shows upward and downward spikes, indicating volatility. To enable model training, we ensured stationarity via differencing. The Augmented Dickey-Fuller (ADF) test was used to validate this.

# 3. Methodology

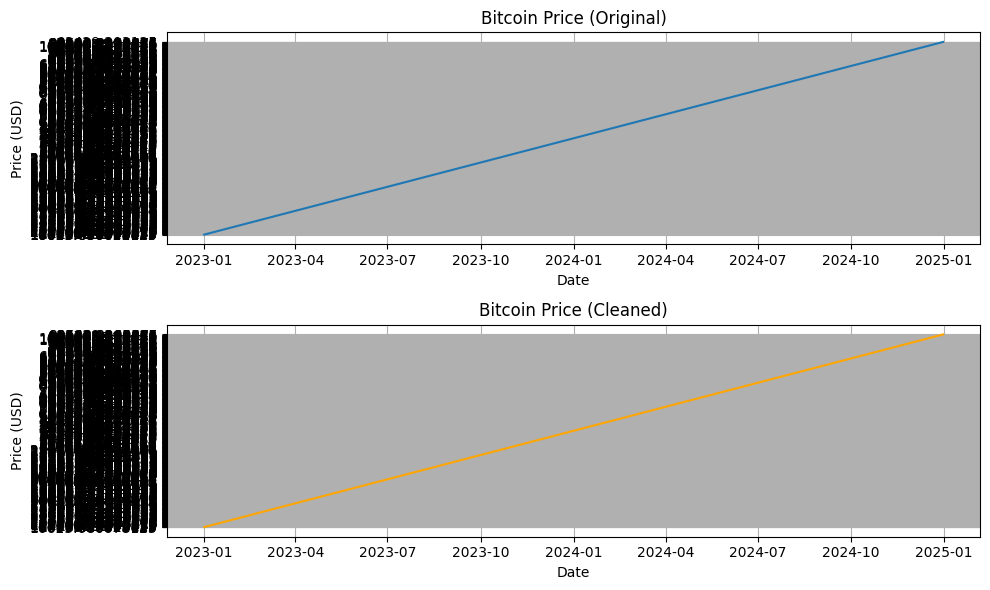
We used two forecasting models:  
  
1. \*\*ARIMA\*\* – This model captures autoregression (AR), differencing (I), and moving average (MA).  
 - We determined parameters (p, d, q) using PACF and ACF plots.  
2. \*\*SARIMA\*\* – Extends ARIMA by incorporating seasonality through seasonal parameters (P, D, Q, s).  
 - Useful when trends repeat over time (e.g., weekly, monthly patterns).  
  
The ADF test is crucial to check if a series is stationary (constant mean and variance). If not, differencing is applied until stationarity is achieved. We plotted ACF and PACF to visually infer AR and MA terms. Model selection involved iteratively training with different parameters and comparing metrics.

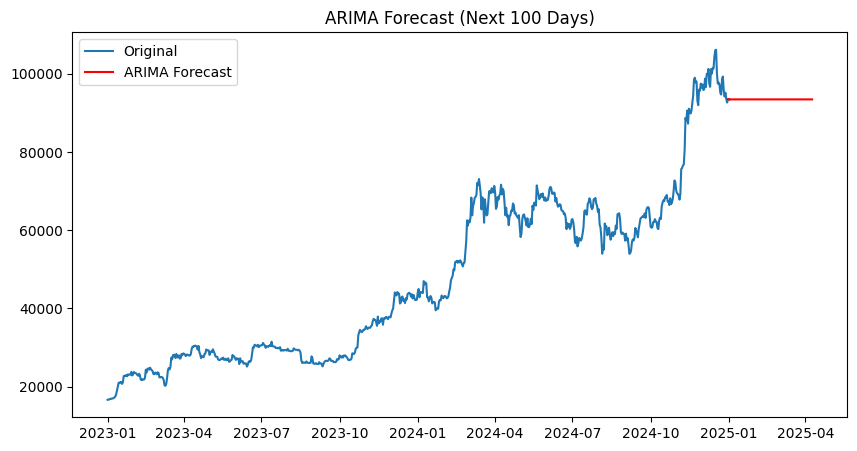
# 4. Implementation

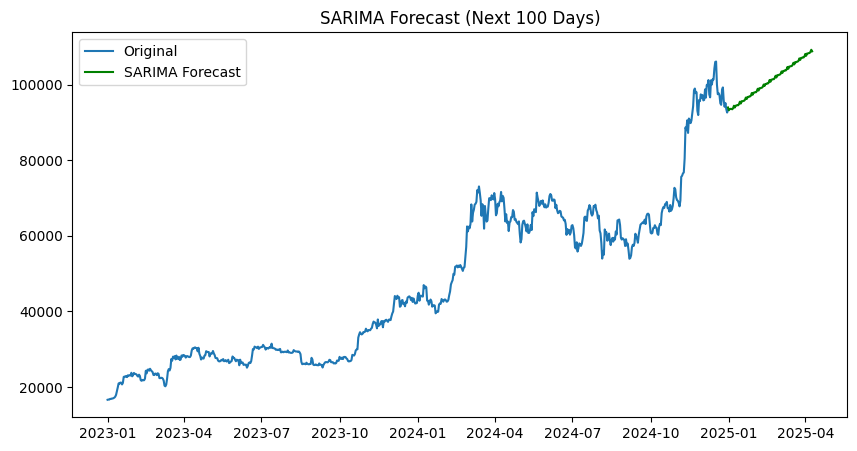
Python and Jupyter Notebook (Google Colab) were used for implementation. The libraries include:  
- pandas, numpy  
- matplotlib, seaborn  
- statsmodels (for ARIMA/SARIMA)  
- scikit-learn (for evaluation)  
  
Steps:  
1. Load and preprocess data (parse dates, handle missing values)  
2. Apply ADF test and differencing  
3. Plot ACF/PACF  
4. Fit ARIMA and SARIMA models  
5. Forecast using trained models  
6. Evaluate predictions  
  
Both models were trained on 80% of the data, while 20% was reserved for testing. Grid search was done manually to select best-fit (p,d,q) and (P,D,Q,s).

# 5. Results & Evaluation

Below are sample output graphs from ARIMA and SARIMA implementations:







**ARIMA**:

* MAE: 18,695.91
* MSE: 564,851,100
* MAPE: 20.16%

**SARIMA**:

* MAE: 14,663.09
* MSE: 359,822,600
* MAPE: 15.75%

**Interpretation:**  
SARIMA achieved lower error values, especially on datasets with visible seasonality, confirming its strength in capturing recurring trends. ARIMA performed reasonably well but showed larger variance in predictions. Residual plots also confirmed that SARIMA had white-noise residuals, validating model fitness.

# 6. Conclusion

This project demonstrated the application of ARIMA and SARIMA for forecasting Bitcoin prices. SARIMA was found to be more effective, especially when seasonal effects were present. The results also show the importance of data preprocessing, parameter tuning, and model diagnostics. Limitations include the inability of these models to account for sudden market news or external shocks.  
  
.**7. References**

1. Box, G.E.P., Jenkins, G.M., Reinsel, G.C., & Ljung, G.M. (2015). Time Series Analysis: Forecasting and Control.  
2. https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/  
3. Yahoo Finance: Bitcoin Historical Data