# **Project Proposal: TimeSeriesAI: Bitcoin Price Prediction**

# Submitted By:

# Noman Asif Khan(22k-5007)

Maaz Siddiqui(22k-5116)

# Course: AI

# Instructor: Ms. Almas Ayesha Ansari

# Submission Date: 16-4-2025

**1. Project Overview**

**Project Topic:**  
This project aims to analyze temporal trends in real-world data and build predictive models using ARIMA and SARIMA. These models will help forecast future values based on historical time series patterns.

**Objective:**  
To develop accurate time series forecasting models using statistical techniques. ARIMA will be used for non-seasonal data trends, while SARIMA will handle seasonality. The goal is to compare their effectiveness and understand temporal dynamics in the dataset.

**2. Data Description**

**Dataset Used:**

* Bitcoin Price Data: This dataset contains historical Bitcoin price data retrieved from Yahoo Finance (or another reliable source), spanning from 2023 to 2025. It includes daily closing prices for Bitcoin (BTC) in USD.

**Features:**

* Date: The timestamp indicating the date of each data point (used as the index for time series analysis).
* Close: The daily closing price of Bitcoin (USD), which is the target variable for forecasting. This is the feature that will be predicted using ARIMA and SARIMA models.

**Preprocessing Steps:**

Handling Missing Values: Missing values will be handled using forward-fill (fillna(method='ffill')), propagating the last valid observation to fill gaps.

Checking Stationarity (ADF Test): The Augmented Dickey-Fuller (ADF) test will be used to check for stationarity. If the p-value is above 0.05, the data is considered non-stationary.

Differencing or Transformations: If the data is non-stationary, differencing will be applied to remove trends and seasonality.

**3. Methodology**

**Techniques to be Used:**

* ARIMA (AutoRegressive Integrated Moving Average)
* SARIMA (Seasonal ARIMA)
* ACF & PACF plots for parameter selection
* Model diagnostics and residual checks

**Evaluation Metrics:**

* Mean Absolute Error (MAE)
* Root Mean Squared Error (RMSE)
* Mean Absolute Percentage Error (MAPE)

**Complexity Analysis:**

* Grid search for parameter tuning adds computation time
* Seasonal models (SARIMA) increase model complexity and training time
* Evaluation on train-test split or cross-validation

**4. Model Design & Forecasting Plan**

**Step-by-step Plan:**

* Import and explore the dataset
* Clean and preprocess data
* Check for seasonality and trends
* Build ARIMA model and evaluate
* Build SARIMA model and evaluate
* Compare performance and visualize results

**Forecasting Objective:**  
To predict the next N time steps based on historical data trends using the best performing model.

**5. Implementation Plan**

**Programming Language:**  
Python

**Libraries and Tools:**

* pandas, numpy
* matplotlib, seaborn
* statsmodels
* scikit-learn

**Timeline:**

* **Week 1-:** Get data, preprocessing
* **Week 2:** ARIMA modeling/**:** SARIMA modeling
* **Week 3:** Model evaluation and comparison/ Final report preparation

**6. References**

* Box, G.E.P. et al. *Time Series Analysis: Forecasting and Control*
* https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/