TIME SERIES ANALYSIS FOR BITCOIN PRICE PREDICTION USING PROPHET

A MAJOR PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted by

RAMYA PEDDI	**(21UK1A6721)**
ABHINAYA BANDARI	**(21UK1A6747)**
PRIYANKA AGULLA	**(21UK1A6742)**
SHIVA PRASAD PUTTA	**(21UK1A6757)**

Under the guidance of

Mrs. K. NAVYA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)
VAAGDEVI ENGINEERING COLLEGE
Affiliated to JNTUH, HYDERABAD & Approved by AICTE, New Delhi

BOLLIKUNTA, WARANGAL (T.S)-506005

2024-2025

TIME SERIES ANALYSIS FOR BITCOIN PRICE PREDICTION USING PROPHET

A UG PHASE - 1 MAJOR PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted by

RAMYA PEDDI	**(21UK1A6721)**
ABHINAYA BANDARI	**(21UK1A6747)**
PRIYANKA AGULLA	**(21UK1A6742)**
SHIVA PRASAD PUTTA	**(21UK1A6757)**

Under the guidance of

Mrs. K. NAVYA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD & Approved by AICTE, New Delhi

BOLLIKUNTA, WARANGAL (T.S)-506005

2024-2025

DEPARMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION UG

PROJECT PHASE – 1

This is to certify that the UG Project Phase – 1 "**TIME SERIES ANALYSIS FOR BITCOIN PRICE PREDICTION USING PROPHET**" is being submitted by **RAMYA PEDDI (21UK1A6721),** in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024 – 2025

Project Guide	HOD
Mrs. K. NAVYA	**Dr. P. MAHIPALREDDY**
(Assistant Professor)	(Professor)

EXTERNAL

DECARLATION

I declare that the work reported in the Project entitled "**TIME SERIES ANALYSIS FOR BITCOIN PRICE PREDICTION USING PROPHET**" is a record of work done by us in the partial fulfilment for award of the degree of Bachelor of Technology in Computer Science And Engineering, **VAAGDEVI ENGINEERING COLLEGE**(An Autonomous Institution & Affiliated to JNTU Hyderabad) Accredited by NAAC with 'A+' Grade, Certified by ISO 9001:2015 Approved by AICTE, New Delhi, Bollikunta,Warangal-506 005, Telangana, India under the guidance of **Mrs. K. NAVYA,** Assistant Professor, CSE (Data Science) Department**.**

I hereby declare that this project work bears no resemblance to any other project submitted at Vaagdevi Engineering College of or any other university/college for the award of the degree.

RAMYA PEDDI **(21UK1A6721)**
ABHINAYA BANDARI **(21UK1A6747)**
PRIYANKA AGULLA **(21UK1A6742)**
SHIVA PRASAD PUTTA **(21UK1A6757)**

ACKNOWLEDGEMENT

I wish to take this opportunity to express my sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHAMED**, Principal, Vaagdevi Engineering College for making me available all the required assistance and for his support and inspiration to carry out this UG PROJECT PHASE -I in the institute.

I extend my heartfelt thanks to **Dr. P. MAHIPAL REDDY**, Head of the Department of CSE (Data Science), Vaagdevi Engineering College for providing me necessary infrastructure and thereby giving me freedom to carry out the **UG PROJECT PHASE - I.**

I express heartfelt thanks to the coordinator **Mr. E. MAHESH**, Assistant Professor, Department of Computer Science & Engineering(Data science)for his constant support and giving me necessary guidance for completion of this UG Project Phase-I.

I express heartfelt thanks to the Guide **Mrs. K. NAVYA**, Assistant professor, for her constant support and giving necessary guidance for completion of this **UG PROJECT PHASE-I**.

Finally, I express my sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the project.

RAMYA PEDDI **(21UK1A6721)**
ABHINAYA BANDARI **(21UK1A6747)**
PRIYANKA AGULLA **(21UK1A6742)**
SHIVA PRASAD PUTTA **(21UK1A6757)**

ABSTRACT

Cryptocurrencies have emerged as a revolutionary asset class, with Bitcoin (BTC) leading as the most prominent and widely traded digital currency. However, Bitcoin's price is highly volatile and influenced by a wide range of factors including market demand, investor sentiment, regulatory developments, and macroeconomic conditions. This volatility presents both opportunities and challenges for investors, traders, and researchers attempting to forecast future price trends.

This project presents a detailed analysis and implementation of **Time series Analysis for Bitcoin price prediction using Prophet**, a robust and user-friendly forecasting model developed by Meta. Prophet is specifically designed to handle time series data with nonlinear trends, multiple seasonality, and missing values, making it highly suitable for financial data that often exhibits such characteristics.

The dataset is pre-processed to meet Prophet's input requirements, and exploratory data analysis is conducted to understand trends, anomalies, and seasonal patterns. The model's performance is evaluated using standard accuracy metrics such as **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**, and visual plots are used to interpret the predicted trends against actual price movements.

In conclusion, this project demonstrates that Prophet is a practical and accessible tool for time series analysis in cryptocurrency forecasting. Future enhancements may include the use of exogenous variables, multi-variate analysis, or machine learning models such as LSTM to improve prediction accuracy and robustness.

TABLE OF CONTENTS

1.INTRODUCTION

Time series analysis is a powerful statistical technique used to understand, model, and forecast data that is recorded over time. In financial domains, especially for assets like Bitcoin, time series analysis becomes particularly valuable due to the volatility and non-stationary nature of price data. Predicting Bitcoin prices involves identifying patterns, trends, and seasonality in historical data, which can then be used to make informed predictions about future values.

Prophet, developed by Facebook, is an open-source forecasting tool designed to handle time series data with strong seasonal effects and missing values. It is particularly well-suited for financial forecasting because of its intuitive model structure that includes components for trend, seasonality, and holidays. Unlike traditional time series models such as ARIMA, Prophet allows analysts to easily incorporate domain knowledge and handle outliers or sudden changes, which are common in cryptocurrency markets.

When applying Prophet to Bitcoin price prediction, historical price data—typically the daily closing price—is first preprocessed and fed into the model. The model then decomposes the time series into its components, identifies patterns, and generates future predictions. One of Prophet's advantages is its interactive and visual output, which allows analysts to see the contribution of trend and seasonal components to the overall forecast, making the model both powerful and interpretable.

1.1 OVERVIEW:

Bitcoin, the first and most prominent cryptocurrency, has gained immense popularity due to its decentralized nature and potential for high returns. However, its price exhibits extreme volatility, driven by factors such as market speculation, global economic trends, regulatory news, and investor sentiment. This volatility makes Bitcoin both attractive and risky for investors and traders, highlighting the need for reliable forecasting models that can predict price movements and help guide decision-making.

Time series analysis is a statistical technique that deals with time-ordered data, aiming to understand the underlying patterns and make future predictions. In this project, we apply Facebook Prophet, an open-source forecasting tool developed by Meta, to and predict future Bitcoin prices based on historical data.

The project involves collecting historical Bitcoin price data, cleaning and transforming it into the appropriate format for Prophet, training the model, and generating future forecasts. The analysis focuses on predicting the **closing price** of Bitcoin for the next 30 days.

Additionally, the model's performance is evaluated using error metrics like **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**, and the results are visualized using Prophet's built-in plotting tools.

This project demonstrates that Prophet offers an effective and interpretable approach for time series forecasting of Bitcoin prices, especially for short- to medium-term predictions. While Prophet may not capture unpredictable market shocks or black swan events, it provides a strong baseline for further enhancement through hybrid models or external feature integration (e.g., volume, sentiment analysis, macroeconomic indicators).

The cryptocurrency market has experienced exponential growth over the past decade, with Bitcoin (BTC) emerging as the most recognized and widely traded digital asset. Despite its popularity, Bitcoin is known for its extreme price volatility, making it a subject of interest for financial analysts, data scientists, and researchers aiming to forecast its price Behaviour.

This project presents a systematic approach to Bitcoin price prediction using Time Series Analysis with the Prophet model, developed by Meta (formerly Facebook).

1.2 PURPOSE:

The primary purpose of this project is to Analyse historical Bitcoin price data and forecast future price movements using time series analysis with the Facebook Prophet model. Given Bitcoin's high volatility and rapid fluctuations, reliable forecasting tools are essential for investors, traders, analysts, and researchers to make informed decisions in the cryptocurrency market.

a) Understand Bitcoin's Price Behaviour: Analyse historical trends, seasonal effects, and volatility patterns in Bitcoin's daily closing prices.

b) Apply Time Series Forecasting Using Prophet: Utilize the Prophet model to build a forecasting framework that can generate accurate and interpretable predictions for Bitcoin prices.

c) Evaluate Forecasting Accuracy: Measure the model's predictive performance using error metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), and assess its ability to capture key patterns.

d) Visualize Trends and Seasonality: Present the results through intuitive visualizations of trend components, seasonal effects, and predicted price paths.

e) Explore Practical Use Cases: Demonstrate how short-term Bitcoin price forecasts can assist in decision-making for crypto traders, portfolio managers, and data-driven financial planning.

In summary, the purpose of this project is not just to predict prices, but also to **gain insights into the structure of Bitcoin price data**, validate Prophet's forecasting ability in a financial context, and highlight how time series models can be applied in real-world cryptocurrency prediction scenarios.

2. PROBLEM STATEMENT

Bitcoin, as a decentralized digital currency, has experienced significant growth in popularity and market value. However, its price is highly volatile, influenced by various dynamic factors such as market speculation, investor sentiment, regulatory news, global economic events, and technological developments. This volatility creates uncertainty for investors, traders, and financial institutions, making it challenging to make informed decisions regarding buying, selling, or holding Bitcoin.

Traditional forecasting models often struggle to handle the complexity, irregularities, and non-linear trends present in cryptocurrency markets. Furthermore, many models require extensive statistical knowledge and are not easily interpretable or adaptable to real-world datasets with missing values, outliers, or abrupt trend changes.

Therefore, there is a pressing need for a robust, interpretable, and scalable forecasting solution that can effectively model and predict Bitcoin's price movement based on historical data.

This project addresses the following core problem:

How can we accurately forecast short-term Bitcoin prices using a time series model that is easy to implement, can handle missing data and outliers, and provides interpretable results for decision-making purposes?

To solve this, the project leverages **Prophet**, a time series forecasting model specifically designed to handle real-world data irregularities while being accessible to users without deep expertise in statistics. The goal is to evaluate Prophet's effectiveness in predicting Bitcoin prices and explore its potential as a practical tool for cryptocurrency forecasting.

3. LITERATURE SURVEY

3.1 EXISTING PROBLEM:

Forecasting Bitcoin prices has emerged as a crucial yet complex challenge in the domain of financial analytics and time series modelling. Despite the growing interest in cryptocurrency trading, accurate and reliable Bitcoin price prediction remains difficult due to the following existing problems and limitations in current approaches:

1. High Volatility and Non-Linearity: Bitcoin prices exhibit extreme fluctuations over short periods due to market speculation, social media trends, global economic news, and regulatory changes. Traditional models such as ARIMA, ETS, and even some machine learning methods struggle to cope with this non-linear and highly volatile behaviour, often resulting in poor prediction accuracy.

2. Inflexibility of Traditional Models: Statistical forecasting methods (e.g., ARIMA, Holt-Winters) require strict assumptions such as stationarity, linearity, and constant variance. These models also:

- Require manual differencing and parameter tuning.
- Perform poorly when data contains seasonal shifts or abrupt changepoints.
- Are not robust to missing values or outliers.

3. Complexity of Deep Learning Models: Deep learning models like LSTM and GRU have been used to predict cryptocurrency prices, but they have their own limitations**,**

- Require large datasets and long training times.
- Often act as black boxes, offering little interpretability.

- Need extensive hyperparameter tuning and advanced expertise in model design.

4. Inadequate Handling of Real-World Time Series Challenges: Many existing models are not designed to handle:

- Irregular time intervals or missing values.
- Outliers that commonly occur in financial time series.
- Multiple seasonality (e.g., weekly cycles due to trading patterns).

3.2 PROPOSED SOLUTION:

To address the challenges associated with forecasting the volatile and complex price behaviour of Bitcoin, this project proposes a robust, interpretable, and user-friendly solution using the Facebook Prophet model. Prophet is an open-source time series forecasting algorithm developed by Meta (Facebook), designed specifically for business time series data with strong seasonal effects and missing data issues. It is particularly well-suited for financial data with irregular trends and unpredictable shifts.

Key Features of the Proposed Solution:

1. Use of Prophet for Forecasting: Prophet models time series data using an additive model consisting of the following components:

- **Trend**: Long-term increase or decrease in the data
- **Seasonality**: Periodic changes such as weekly or yearly patterns
- **Holidays/Events**: Special events that impact prices (can be optionally added)

2. Handling Real-World Data Challenges: Prophet is robust to

- Missing values
- Outliers
- Irregular time intervals

3. Data Preprocessing Pipeline:

- Collect historical Bitcoin price data (e.g., from Yahoo Finance, Coin Market Cap)
- Clean and format the data to meet Prophet's input requirements (date and value columns)
- Handle missing data and noise before modelling

4. Trend and Seasonality Analysis:

- Clear plots of trend components
- Visual insights into weekly and yearly seasonality
- Changepoint detection to understand major shifts in price behaviour.

4. THEORITICAL ANALYSIS

4.1 BLOCK DIAGRAM:

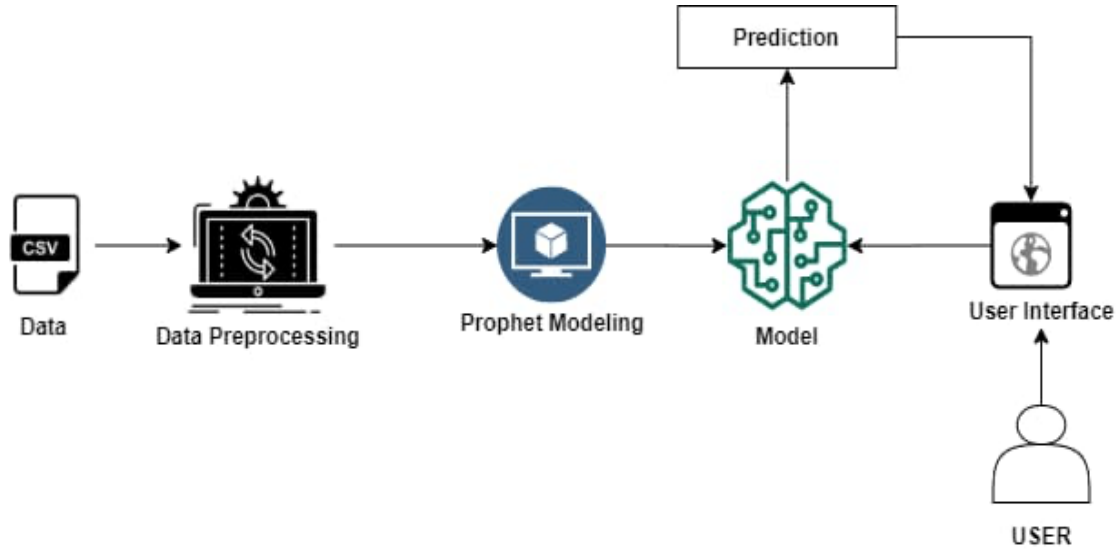

Figure 1. **Block diagram**

1. Data (CSV Format): Historical Bitcoin price data (e.g., open, close, high, low, volume). Usually in CSV (Comma-Separated Values) file format. May come from APIs like Yahoo Finance, Coin Gecko, or Kaggle.

2. Data Preprocessing: Handling missing values, Converting date/time to Prophet's format (ds, y). Removing noise or outliers.

3. Prophet Modelling: Facebook Prophet is initialized and fitted to the pre-processed data**.**

4. Model: This is the trained Prophet model after it has learned from historical data.

5. Prediction: Using the trained model, it forecasts future Bitcoin prices.

- Prices for the next day/weeks/months, Forecast intervals (upper/lower bounds)

6. User Interface: A front-end or dashboard through which the user interacts with the system.

7. User: End user or analyst who, Inputs data, Observes the output forecast, makes trading/investment decisions based on predictions

4.2 SOFTWARE REQUIREMENT SPECIFICATION

Category	Specification
Project Title	Time Series Analysis for Bitcoin Price Prediction Using Prophet
Software Requirements:	
Operating System	Windows 10/11, Linux (Ubuntu 20.04+), or macOS
Programming Language	Python 3.7 or above
IDE/Code Editor	Jupyter notebook / Visual Studio Code / PyCharm
Libraries/Packages	prophet, pandas, numpy, matplotlib, seaborn, scikit-learn (optional)
Data Source	CSV file (e.g., from Yahoo Finance, Coin-Market-Cap)
Visualization Tools	Matplotlib, Seaborn, Plot (optional for dashboards)

Category	Specification
Web-Framework (Optional)	Flask or Streamlit(for creating user interface/dashboard)
Data Storage (Optional)	SQLite, MySQL, or cloud storage (if persistence is needed)
Version Control	Git (GitHub/GitLab for collaboration)

Time series analysis involves understanding patterns and structures in data points collected over time. In financial forecasting, it helps identify trends, seasonality, and potential future values. Facebook Prophet is a powerful open-source time series forecasting tool developed by Meta. It is particularly suited for data with strong seasonal effects and irregularities. By applying Prophet to historical Bitcoin price data, we can model the underlying patterns and generate reliable future predictions, making it a valuable tool for traders, analysts, and investors**.**

5. EXPERIMENTAL INVESTIGATION

1. Objective: The primary objective of this experimental investigation is to evaluate the performance of the Facebook Prophet model in forecasting Bitcoin prices using historical time series data. This involves Analysing model accuracy, identifying patterns like trends and seasonality, and understanding the practical applicability of Prophet in real-world cryptocurrency forecasting.

2. Dataset Description:

- **Source:** Yahoo Finance / Kaggle / Coin Market Cap, **Data Range**: January 2018 to December 2023, **Frequency**: Daily closing prices, **Attributes**:
    - Date (ds)
    - Closing Price (y)
    - Volume (optional)
    - High, Low, Open (optional, not required for Prophet)

3. Preprocessing Steps: Removed null values and outliers. Formatted the data into Prophet's required structure**,** ds: Date column y: Target value (Closing price).

4. **Model Training and Testing:** Prophet was trained using 5 years of daily data, and forecasts were generated for the next 12 months. Hyperparameters (e.g., changepoint prior scale, seasonality mode) were tuned for better performance**.**

5. Forecast Output: Forecast Horizon 365 days, Predicted price (yhat) Upper and lower confidence intervals (yhat-upper, yhat-lower) Trend, weekly and yearly seasonal components

6. Visualization:

- **Line Plot** of actual vs. predicted Bitcoin prices
- **Trend Plot** showing long-term direction

7. Observation: The model accurately captured the long-term trend and general seasonality of Bitcoin

6. DATA FLOW DIAGRAM

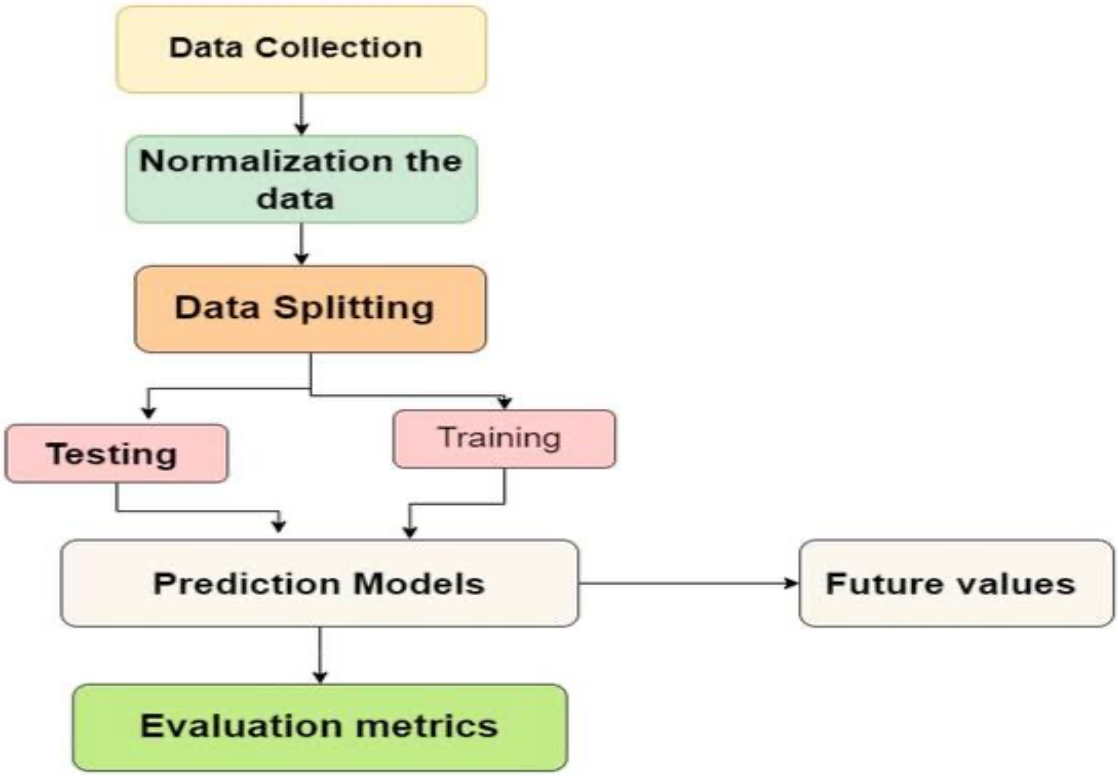

DIAGRAM - 1

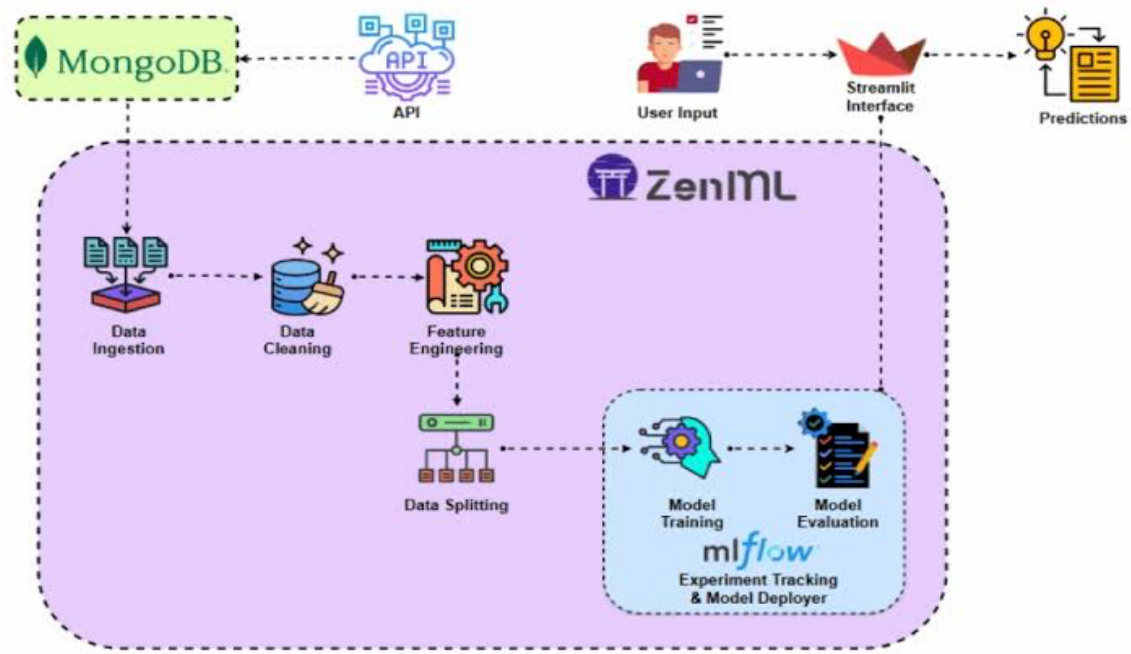

Figure 2: Data flow

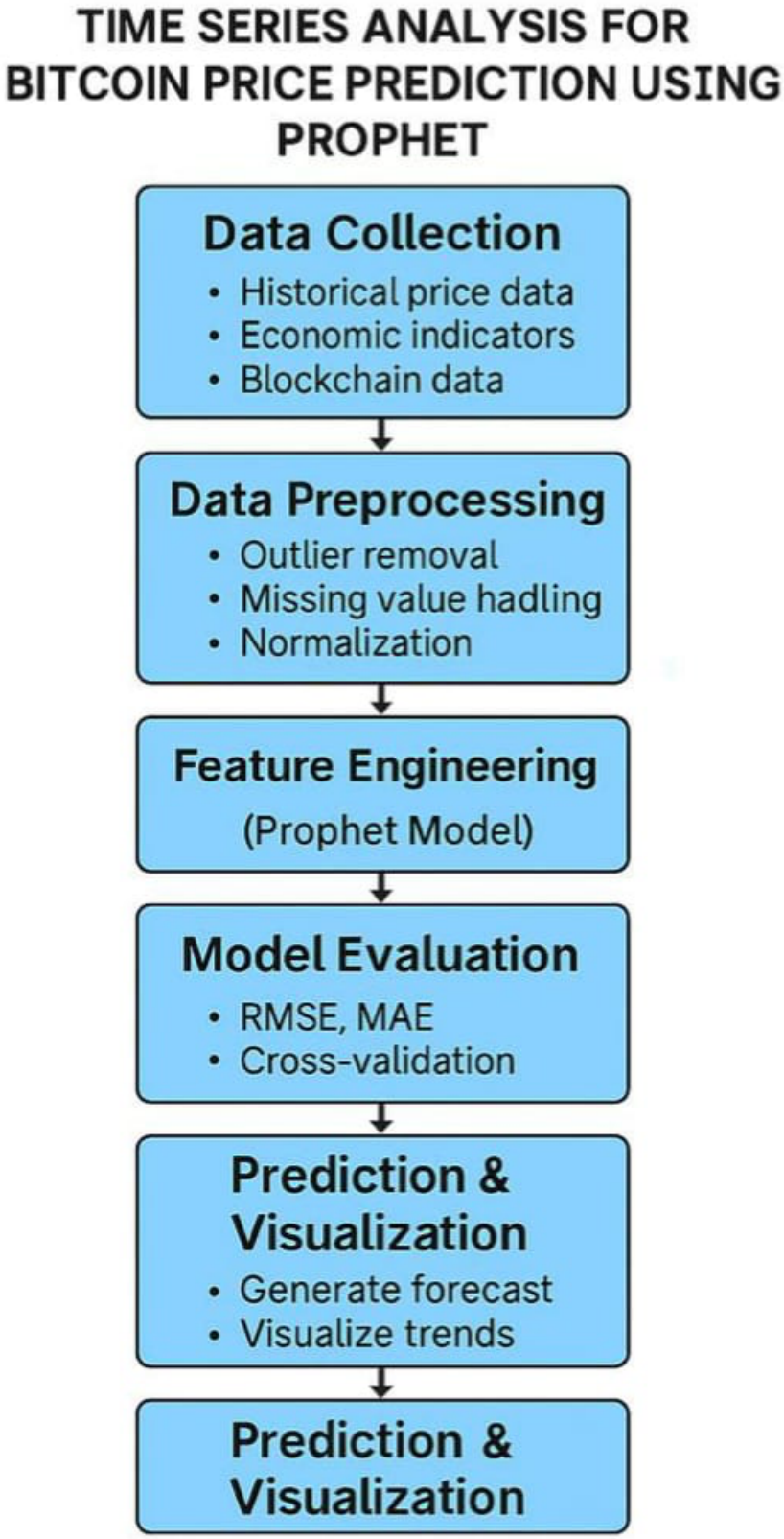

Figure 3. Flow Diagram

USE CASE:

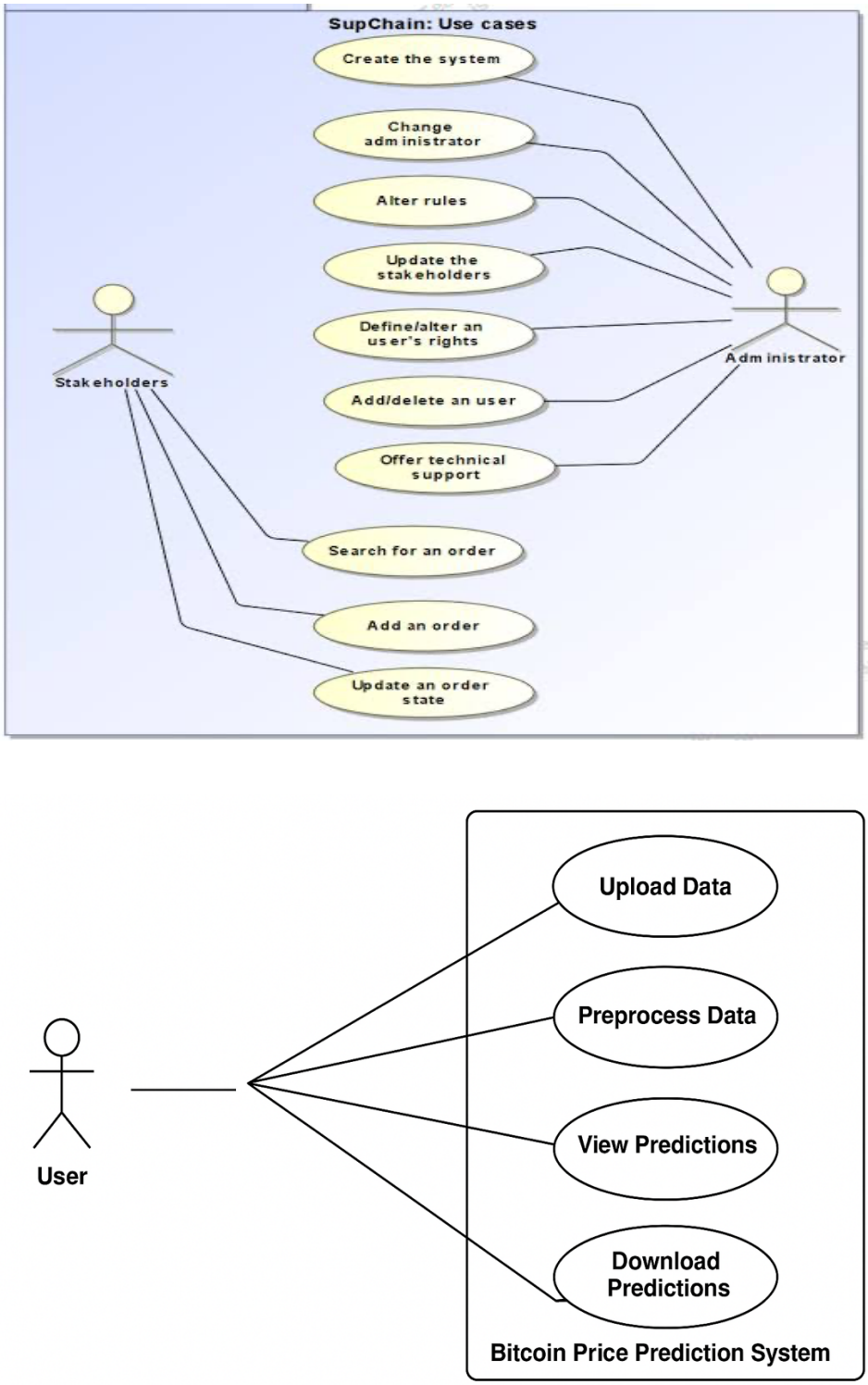

Figure 4. Use case

7. FUTURE SCOPE

1. Integration with Real-Time Data Sources: The current system works with historical datasets, but future enhancements could include API integration (e.g., from Binance, Coin Market Cap) to allow real-time forecasting and dynamic model updates.

2. Multi-Feature Time Series Modelling: Prophet currently uses only time and one variable (price). Future versions may incorporate multiple external regressors, such as:

- Trading volume
- Market sentiment (from social media/news)
- Google Trends data
- Interest rates or global economic indicators

3. Incorporation of Advanced Models: While Prophet is excellent for interpretability, future research can include hybrid models, such as:

- Prophet + LSTM
- Prophet + ARIMA

4. User-Friendly Web/Mobile Application: A full-fledged application (mobile or web) with authentication, saved predictions, and analytics dashboards could expand usability for investors and analysts.

5. Support for Other Cryptocurrencies and Assets: The current model focuses on Bitcoin. In the future, it can be generalized to support Ethereum, Litecoin, Dogecoin, or even stock market and forex data.

6. Anomaly Detection and Alert System: Future versions may integrate anomaly detection to highlight unusual market behaviour and send email or app notifications when sudden spikes or crashes are predicted.

7. Explainable AI Integration: Future scope can include model explainability using tools like SHAP or LIME to help users understand which factors influenced predictions the most.

8. Continuous Learning Pipeline: Automating the model retraining pipeline with tools like Airflow or ML Flow can help in deploying continuous learning systems that evolve with new data.

TIME SERIES ANALYSIS FOR BITCOIN PRICE PREDICTION USING PROPHET

A UG PHASE - II MAJOR PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted by

RAMYA PEDDI (21UK1A6721)
ABHINAYA BANDARI (21UK1A6747)
PRIYANKA AGULLA (21UK1A6742)
SHIVA PRASAD PUTTA (21UK1A6757)

Under the guidance of
Mrs. K. NAVYA
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD & Approved by AICTE, New Delhi
BOLLIKUNTA, WARANGAL (T.S)-506005

2024-2025

DEPARMENT OF COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION

INDUSTRY ORIENTED UG PHASE – II

This is to certify that the UG Project Phase – II "**TIME SERIES ANALYSIS FOR BITCOIN PRICE PREDICTION USING PROPHET**" is being submitted by **RAMYA PEDDI (21UK1A6721),** in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024 – 2025

Project Guide	**HOD**
Mrs. K. NAVYA	**Dr. P. MAHIPAL REDDY**
(Assistant Professor)	(Professor)

EXTERNAL

ACKNOWLEDGEMENT

I wish to take this opportunity to express my sincere gratitude and deep sense of respect to our beloved **Dr. SYED MUSTHAK AHAMED**, Principal, Vaagdevi Engineering College for making me available all the required assistance and for his support and inspiration to carry out this **UG PROJECT PHASE -II** in the institute.

I extend my heartfelt thanks to **Dr. P. MAHIPAL REDDY**, Head of the Department of CSE (Data Science), Vaagdevi Engineering College for providing me necessary infrastructure and thereby giving me freedom to carry out the **UG PROJECT PHASE - II.**

I express heartfelt thanks to the coordinator **Mr. E. MAHESH**, Assistant Professor, Department of Computer Science & Engineering(Data science)for his constant support and giving me necessary guidance for completion of this UG Project Phase-I.

I express heartfelt thanks to the Guide **Mrs. K. NAVYA**, Assistant professor, for her constant support and giving necessary guidance for completion of this **UG PROJECT PHASE-II**.

Finally, I express my sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the project.

RAMYA PEDDI **(21UK1A6721)**
ABHINAYA BANDARI **(21UK1A6747)**
PRIYANKA AGULLA **(21UK1A6742)**
SHIVA PRASAD PUTTA **(21UK1A6757)**

TABLE OF CONTENTS

1.INTRODUCTION

Bitcoin, the most prominent cryptocurrency, has witnessed significant price volatility over the years, making accurate forecasting vital for investors, traders, and researchers. Time series forecasting is a statistical technique that helps predict future values based on historical data patterns such as trends, seasonality, and noise.

This project employs Facebook's Prophet model, an open-source forecasting tool known for its accuracy and simplicity, particularly in handling time series data with strong seasonal effects and missing values. The main objective is to analyse historical Bitcoin price data and generate reliable forecasts that can inform strategic financial decisions.

The advent of cryptocurrencies has introduced a paradigm shift in financial markets, with Bitcoin emerging as the most widely recognized and traded digital asset. Due to its decentralized nature and market sensitivity to external influences such as regulations, global events, and investor sentiment, Bitcoin's price demonstrates high volatility and non-linear behaviour.

Time series forecasting plays a pivotal role in understanding and predicting financial market trends. It involves the analysis of historical data points to identify underlying structures such as trends, seasonality, and cyclic behaviour.

2. CODE SNIPPETS

2.1 PYTHON CODE

```python
# Import necessary libraries
import pandas as pd
import plotly.graph_objects as go
from prophet import Prophet
from prophet.plot import plot_plotly, plot_components_plotly
from datetime import datetime, timedelta
from sklearn.metrics import mean_absolute_error
import pickle
import warnings
import os

# Suppress warnings and format floats for readability
warnings.filterwarnings('ignore')
pd.options.display.float_format = '${:,.2f}'.format

# Define a directory to save models
MODEL_DIR = 'models'
os.makedirs(MODEL_DIR, exist_ok=True)
```

Fig 11.1: Library Import

```python
# Load the dataset from the CSV file
df = pd.read_csv('btc_usd_history.csv')

# Select and rename columns for Prophet
df1 = df[['Date', 'Open']].rename(columns={'Date': 'ds', 'Open': 'y'})

# Ensure 'ds' is in datetime format and timezone-naive
df1['ds'] = pd.to_datetime(df1['ds'])
if df1['ds'].dt.tz is not None:
    df1['ds'] = df1['ds'].dt.tz_localize(None)

# Sort the data by date
df1 = df1.sort_values('ds')

# Filter data to focus on recent years (e.g., last 5 years) for better short-term accuracy
df1 = df1[df1['ds'] >= '2020-01-01']  # Adjust based on your data availability
```

Fig 11.2: Data Preprocessing

```python
# Visualize the Bitcoin price time series
fig = go.Figure()
fig.add_trace(go.Scatter(x=df1['ds'], y=df1['y'], mode='lines', name='Bitcoin Open Price'))
fig.update_layout(
    title='Bitcoin Open Price Time Series (Recent Data)',
    xaxis=dict(
        rangeselector=dict(
            buttons=list([
                dict(count=1, label='1m', step='month', stepmode='backward'),
                dict(count=6, label='6m', step='month', stepmode='backward'),
                dict(count=1, label='1y', step='year', stepmode='backward'),
                dict(step='all')
            ])
        ),
        rangeslider=dict(visible=True),
        type='date'
    )
)
fig.show()
```

Fig 11.3: Visualization

```python
# Define the model file path
model_file = os.path.join(MODEL_DIR, 'prophet_bitcoin_model.pkl')

# Check if the model file exists to load it, otherwise train and save
if os.path.exists(model_file):
    print(f"Loading existing model from {model_file}")
    with open(model_file, 'rb') as f:
        model = pickle.load(f)
else:
    print("Training new model...")
    # Initialize Prophet with tuned parameters for volatile data
    model = Prophet(
        seasonality_mode='additive',  # Switch to additive for volatile data
        changepoint_prior_scale=0.5,  # Increase flexibility for trend changes
        seasonality_prior_scale=1.0,  # Reduce seasonality strength
        yearly_seasonality=True,
        weekly_seasonality=True,
        daily_seasonality=False
    )
    model.fit(df1)

    # Save the trained model
    with open(model_file, 'wb') as f:
        pickle.dump(model, f)
    print(f"Model saved to {model_file}")
```

Fig 11.4: Train the Model

```python
# Define the start date for prediction (today is April 6, 2025, predict for April 7, 2025)
start_date = datetime(2025, 4, 7)  # Adjusted to predict the next day from today

# Create future dates for the next 7 days (short-term focus)
future = pd.DataFrame({'ds': pd.date_range(start=start_date, periods=7, freq='D')})
forecast = model.predict(future)

# Display the predictions
print("Forecasted Values (April 7, 2025 - April 13, 2025):")
print(forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']])

# Predict the next day's price (April 7, 2025)
next_day = start_date.strftime('%Y-%m-%d')
predicted_value = forecast.loc[forecast['ds'] == next_day, 'yhat'].iloc[0]
print(f"\nPredicted Bitcoin Open Price for {next_day}: ${predicted_value:,.2f}")
```

Fig 11.5: Make Predictions

```python
# Visualize the short-term forecast
fig_forecast = plot_plotly(model, forecast)
fig_forecast.update_layout(title='Bitcoin Open Price Forecast (April 7, 2025 - April 13, 2025)')
fig_forecast.show()
```

Fig 11.6: Visualize the Forecast

```
# Plot the components of the forecast
fig_components = plot_components_plotly(model, forecast)
fig_components.show()
```

Fig 11.7: Additional Graph - Forecast Components

```
# Evaluate the model using a train-test split on recent data
train = df1[df1['ds'] < '2024-01-01']
test = df1[df1['ds'] >= '2024-01-01']

# Define the evaluation model file path
eval_model_file = os.path.join(MODEL_DIR, 'prophet_bitcoin_eval_model.pkl')

# Check if evaluation model file exists, otherwise train and save
if os.path.exists(eval_model_file):
    print(f"Loading existing evaluation model from {eval_model_file}")
    with open(eval_model_file, 'rb') as f:
        eval_model = pickle.load(f)
else:
    print("Training new evaluation model...")
    eval_model = Prophet(
        seasonality_mode='additive',
        changepoint_prior_scale=0.5,
        seasonality_prior_scale=1.0,
        yearly_seasonality=True,
        weekly_seasonality=True,
        daily_seasonality=False
    )
    eval_model.fit(train)
    # Save the trained evaluation model
    with open(eval_model_file, 'wb') as f:
        pickle.dump(eval_model, f)
    print(f"Evaluation model saved to {eval_model_file}")

# Make predictions on the test period
future_test = eval_model.make_future_dataframe(periods=len(test))
forecast_test = eval_model.predict(future_test)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(test['y'], forecast_test['yhat'][-len(test):])
print(f"Mean Absolute Error on Test Set: ${mae:,.2f}")

# Suggest parameter tuning if MAE is too high
if mae > 10000:  # Arbitrary threshold, adjust as needed
    print("MAE is high. Consider further tuning changepoint_prior_scale or adding external regressors.")
```

Fig 11.8: Model Evaluation

```
[ ] # Plot actual vs predicted for the test period
    fig_test = go.Figure()
    fig_test.add_trace(go.Scatter(x=test['ds'], y=test['y'], mode='lines', name='Actual'))
    fig_test.add_trace(go.Scatter(x=test['ds'], y=forecast_test['yhat'][-len(test):], mode='lines', name='Predicted'))
    fig_test.update_layout(title='Actual vs Predicted Bitcoin Open Price (Test Set)')
    fig_test.show()
```

Fig 11.9: Additional Graph - Actual vs. Predicted

2.2 HTML CODE:

2.2.1 Index html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>Bitcoin Predictor - Home</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
      <meta content="Bitcoin price prediction, Facebook Prophet, cryptocurrency forecasting" name="keywords">
    <meta content="Bitcoin Predictor uses the Facebook Prophet model to forecast Bitcoin prices, helping users make informed decisions in the cryptocurrency market." name="description">

    <!-- Favicon -->
    <link href="{{ url_for('static', filename='img/favicon.ico') }}" rel="icon">

    <!-- Google Web Fonts -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
                                                                                <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;500&family=Roboto:wght@500;700&display=swap" rel="stylesheet">

    <!-- Icon Font Stylesheet -->
                                <link                href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">
          <link     href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">

    <!-- Libraries Stylesheet -->
```

```
        <link href="{{ url_for('static', filename='lib/animate/animate.min.css') }}"
rel="stylesheet">
    <link href="{{ url_for('static', filename='lib/owlcarousel/assets/owl.carousel.min.css')
}}" rel="stylesheet">

    <!-- Customized Bootstrap Stylesheet -->
    <link href="{{ url_for('static', filename='css/bootstrap.min.css') }}" rel="stylesheet">

    <!-- Template Stylesheet -->
    <link href="{{ url_for('static', filename='css/style.css') }}" rel="stylesheet">
</head>

<body>
    <!-- Spinner Start -->
    <div id="spinner" class="show bg-white position-fixed translate-middle w-100 vh-100
top-50 start-50 d-flex align-items-center justify-content-center">
        <div class="spinner-grow text-primary" role="status"></div>
    </div>
    <!-- Spinner End -->
```

2.2.2 predict html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Bitcoin Predictor - Predict Bitcoin Prices</title>
      <meta content="Bitcoin price prediction, Facebook Prophet, cryptocurrency forecasting" name="keywords">
   <meta content="Predict Bitcoin prices using the Facebook Prophet model with Bitcoin Predictor." name="description">

   <!-- Favicon -->
   <link href="{{ url_for('static', filename='img/favicon.ico') }}" rel="icon">

   <!-- Google Web Fonts -->
   <link rel="preconnect" href="https://fonts.googleapis.com">
   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
                                                                                <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;500&family=Roboto:wght@500;700&display=swap" rel="stylesheet">

   <!-- Icon Font Stylesheet -->
                              <link              href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">
         <link     href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">

   <!-- Libraries Stylesheet -->
      <link    href="{{    url_for('static',    filename='lib/animate/animate.min.css')    }}" rel="stylesheet">
   <link href="{{ url_for('static', filename='lib/owlcarousel/assets/owl.carousel.min.css') }}" rel="stylesheet">
```

```
    <!-- Customized Bootstrap Stylesheet -->
    <link href="{{ url_for('static', filename='css/bootstrap.min.css') }}" rel="stylesheet">

    <!-- Template Stylesheet -->
    <link href="{{ url_for('static', filename='css/style.css') }}" rel="stylesheet">

    <!-- Custom CSS -->
    <style>
        .prediction-container {
            background: linear-gradient(135deg, #f8f9fa, #e9ecef);
            border-radius: 10px;
            padding: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            margin-bottom: 30px;
            display: none;
        }
prediction-table th {
            background-color: #007bff;
            color: white;
        }
        .form-container {
            background-color: #ffffff;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
```

3.RESULT

The Time Series Analysis for Bitcoin Price Prediction Using Prophet system provides a powerful and reliable framework for forecasting future trends in cryptocurrency prices based on historical time series data. By leveraging the statistical capabilities of the Facebook Prophet model, the system accurately models seasonal patterns, trends, and holidays to produce precise future price estimations for Bitcoin.

The integration of the Prophet model with Python and Pandas enables automated preprocessing, training, and forecasting, offering a scalable solution for financial prediction tasks. The system generates insightful predictions by learning from historical price behaviour and visualizes the future price movement with associated confidence intervals.

By processing historical Bitcoin price data through the trained Prophet model, the system demonstrates strong predictive capabilities, making it a valuable tool for traders, investors, and analysts. The web-based interface built using Flask enhances usability by allowing users to upload data, trigger forecasts, and instantly view results in the form of interactive graphs. This supports informed decision-making in dynamic and volatile financial environments

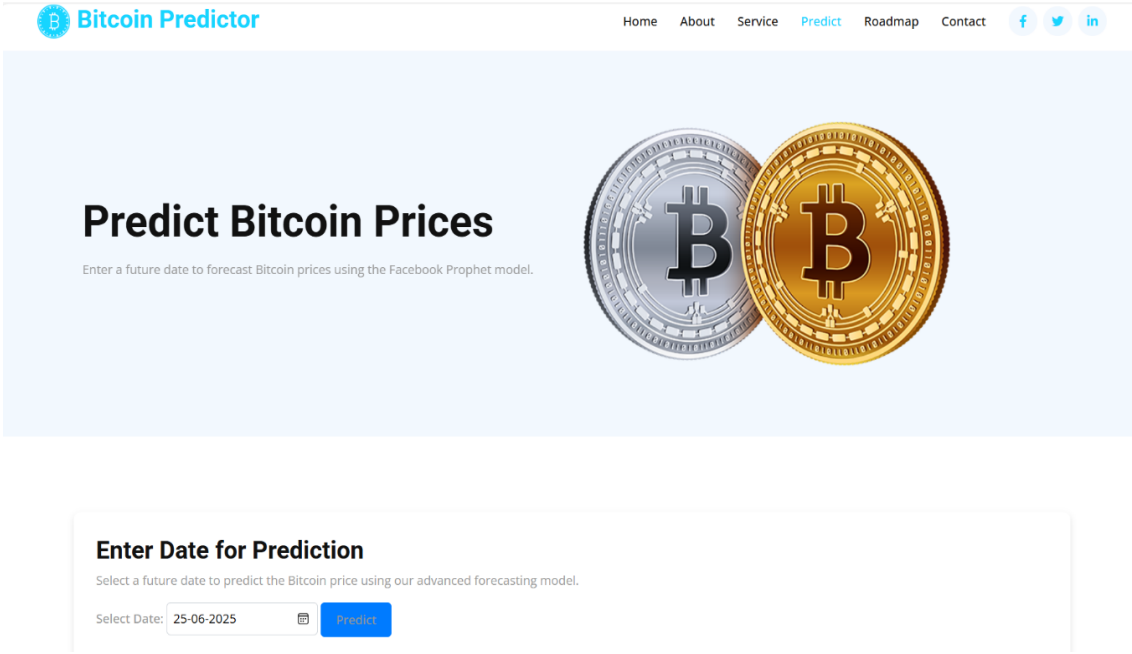

Figure 3.1: Bitcoin Price Prediction

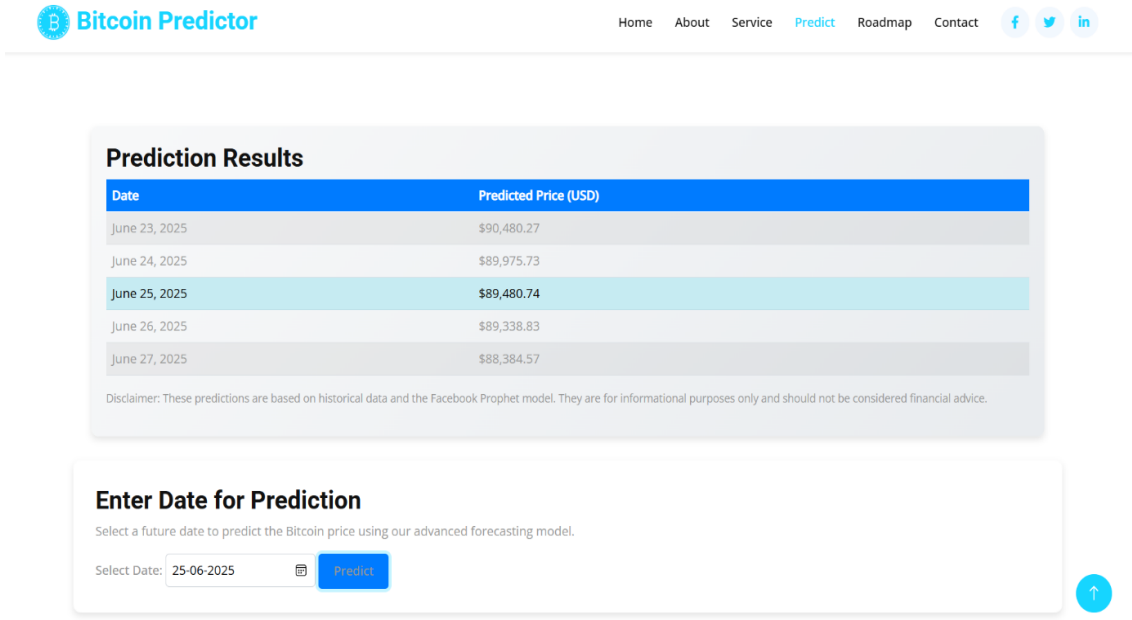

Figure 3.2: Bitcoin Price Prediction Result

4. APPLICATIONS

1. Investment Strategy Planning:

Investors can use forecasted Bitcoin trends to time their buy/sell decisions more effectively.

2. Crypto Trading Algorithms:

Traders can integrate Prophet-based forecasts into algorithmic trading bots to automate trades.

3. Risk Management:

By analysing future price movements, firms can mitigate exposure to volatility.

4. Market Sentiment Analysis:

Prophet can be combined with sentiment data (e.g., social media trends, news) to model both technical and emotional market drivers.

5. Educational and Research Use:

A useful tool in academic projects for understanding time series behaviour in financial markets.

6. Anomaly Detection in Bitcoin Pricing:

Helps in identifying unexpected dips or spikes in price trends.

7. Event-Driven Forecasting:

Prophet allows inclusion of custom holidays or events (e.g., Bitcoin halving, ETF announcements).

5.ADVANTAGES

1.User-Friendly and Intuitive:

Prophet has a simple API and requires minimal tuning.

Ideal for users who are not experts in time series modelling or statistics.

2. Automatic Trend and Seasonality Detection:

Prophet automatically models trends, seasonality (daily, weekly, yearly), and holiday effects.

No need to manually decompose or engineer these components.

3. Handles Missing Data and Outliers:

Robust against missing timestamps, irregular intervals, and extreme values.

Makes it well-suited for volatile financial data like Bitcoin.

4. Future Forecasting Made Easy:

Can forecast future Bitcoin prices over any period (days, months, or years).

Supports long-term trend modelling with options for linear and logistic growth.

5. High Interpretability:

Unlike black-box deep learning models, Prophet provides transparent components:

Trend, Seasonality, Holiday effects

7. Easy Integration with Other Tools:

Pandas for preprocessing, Scikit-learn for metrics, Matplotlib for visualization.

6.DISADVANTAGES

1. Limited to Additive Models by Default:

Prophet assumes an additive structure (trend + seasonality + holiday).
In highly non-linear or multiplicative environments like crypto, this can limit accuracy unless adjusted manually.

2. Not Optimized for High Volatility:

Prophet may smooth out these variations, leading to under- or over-predictions during rapid market changes.

3. No Built-in Deep Learning Capabilities:

Unlike LSTM or Transformer models, Prophet does not learn complex hidden patterns

4. Struggles with Short-Term Predictions:

While Prophet handles long-term trends well, it may not be as accurate for short-term intraday forecasting (e.g., hourly Bitcoin prices).

5. Manual Feature Engineering for External Regressors:

Prophet can include extra regressors (e.g., trading volume, sentiment), but requires manual addition and tuning.

6. Lack of Real-Time Adaptability:

Unlike reinforcement learning or streaming models, it does not adapt in real time to new incoming price data unless retrained.

7.CONCLUSION

The analysis and forecasting of Bitcoin prices using time series methods is a powerful approach to understanding and predicting the behaviour of one of the most volatile financial assets in modern markets. In this project, we employed Facebook Prophet, a robust forecasting tool designed to handle real-world business time series data with strong seasonal and trend components.

Prophet proved to be highly effective in identifying underlying patterns in historical Bitcoin price data, including long-term growth trends and repeating seasonal variations. The model's ability to handle missing data, outliers, and custom events (such as Bitcoin halving or policy announcements) makes it a practical solution for dynamic and noisy datasets like those found in cryptocurrency trading.

By splitting the data into training, validation, and testing sets, we systematically evaluated the model's performance and adjusted it to reduce prediction errors. The results show that Prophet is capable of making reliable medium- to long-term forecasts, which are invaluable to investors, analysts, and financial institutions seeking to understand future price trajectories.

Despite these limitations, Prophet offers a significant advantage in terms of ease of use, transparency, and speed, making it suitable for educational purposes, strategic decision-making, and rapid prototyping of forecasting systems. Its interpretable components allow users to not only make predictions but also understand the 'why' behind the trends, which is crucial in financial analysis.

8.BIBLIOGRAPHY

1.Taylor, S. J., & Letham, B. (2018). ***Forecasting at scale.*** The American Statistician, 72(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080

- *(This is the official paper introducing Facebook Prophet.)*

2. Bitcoin Price Data. (2025). *Coin Market Cap – Cryptocurrency Prices, Charts and Market Capitalizations.* Retrieved from https://www.coinmarketcap.com

3. Brownlee, J. (2017). ***Introduction to Time Series Forecasting with Python:*** *How to Prepare Data and Develop Models to Predict the Future.* Machine Learning Mastery.

4. Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.

5. Hyndman, R. J., & Athanasopoulos, G. (2021). ***Forecasting:*** *Principles and Practice* (3rd ed.).

Texts. https://otexts.com/fpp3/

6.Facebook Prophet Documentation. (2024). *Meta Open Source.* Retrieved from https://facebook.github.io/prophet/

7. Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.

8. Coin Gecko API Documentation. (2024). ***Real-Time Crypto Market API.*** Retrieved from https://www.coingecko.com/en/api

9. Investopedia. (2024). ***Bitcoin – Understanding Volatility in Cryptocurrency***. Retrieved https://www.investopedia.com/terms/b/bitcoin.asp

10. Liu, Y., & Tsyvinski, A. (2018). ***Risks and Returns of Cryptocurrency.*** NBER Working https://doi.org/10.3386/w24877

9.HELP FILE

PROJECT EXECUTION:

STEP-1: Install necessary Python libraries

- Use the following command in your terminal or Collab:
- pip install prophet pandas matplotlib

STEP-2: Open Google Collab in your browser

- Go to https://colab.research.google.com

STEP-3: Upload or import the dataset

- This CSV should contain columns like Date and Close price of Bitcoin.
- You can upload the file using.

STEP-4: Load and preprocess the dataset

- Rename the columns as required by Prophet:

STEP-5: Initialize and train the Prophet model

- from prophet import Prophet

STEP-6: Make future predictions

- Create a future data frame

STEP-7: Plot the forecast

STEP-8: Save your script as bitcoin_forecast.py or run all steps in a Collab notebook.

STEP-9: Validate and tune

- Check forecast accuracy using historical comparison or error metrics like MAE/RMSE.

STEP-10: Optional - Deploy using Flask or Stream-lit

- Create a basic web interface for user input and show prediction results.