Project Design Phase-II
Technology Stack (Architecture & Stack)

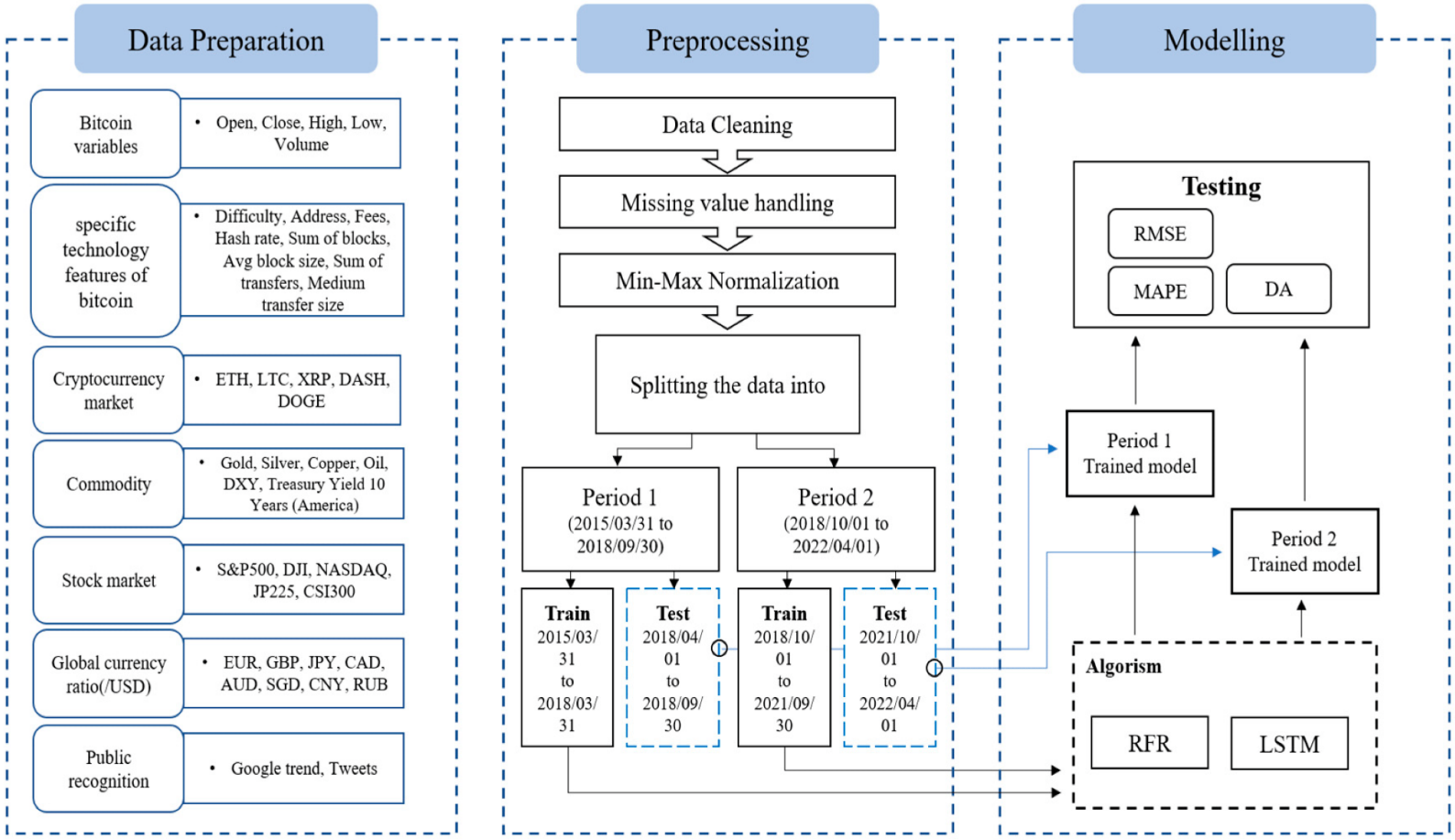
Date	27 October 2023
Team ID	Team - 592087
Project Name	Project - **Time Series Analysis For Bitcoin Price Prediction**
Maximum Marks	4 Marks

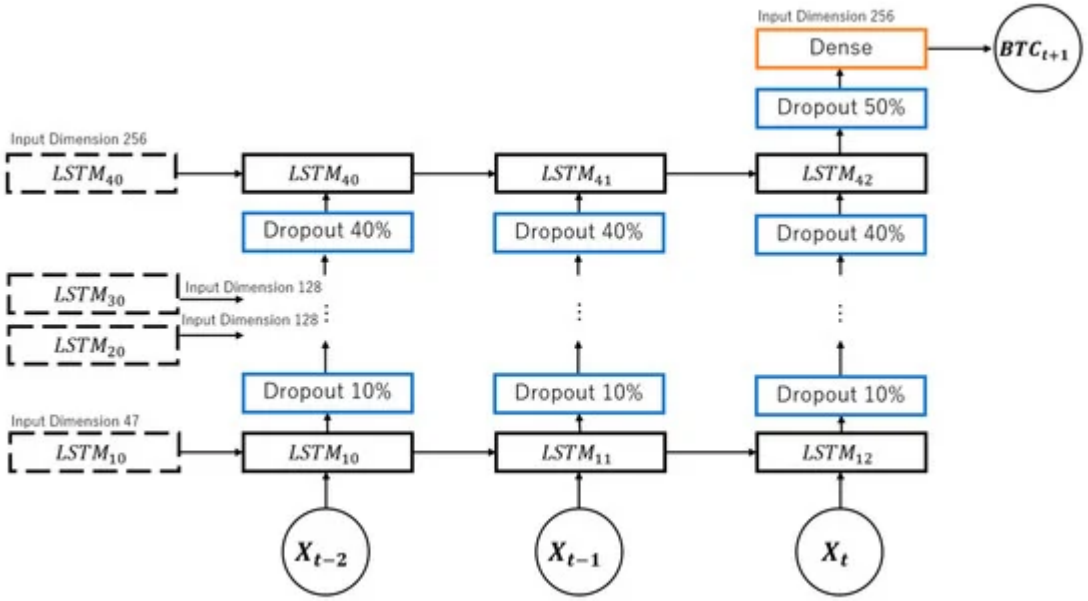
Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table

2

Reference: https://www.mdpi.com/1911-8074/16/1/51


Data Preparation
Bitcoin variables
• Open, Close, High, Low, Volume
specific technology features of bitcoin
• Difficulty, Address, Fees, Hash rate, Sum of blocks, Avg block size, Sum of transfers, Medium transfer size
Cryptocurrency market
• ETH, LTC, XRP, DASH, DOGE
Commodity
• Gold, Silver, Copper, Oil, DXY, Treasury Yield 10 Years (America)
Stock market
• S&P500, DJI, NASDAQ, JP225, CSI300
Global currency ratio(/USD)
• EUR, GBP, JPY, CAD, AUD, SGD, CNY, RUB
Public recognition
• Google trend, Tweets
Preprocessing
Data Cleaning
Missing value handling
Min-Max Normalization
Splitting the data into
Period 1 (2015/03/31 to 2018/09/30)
Period 2 (2018/10/01 to 2022/04/01)
Train 2015/03/31 to 2018/03/31
Test 2018/04/01 to 2018/09/30
Train 2018/10/01 to 2021/09/30
Test 2021/10/01 to 2022/04/01
Modelling
Testing
RMSE
MAPE
DA
Period 1 Trained model
Period 2 Trained model
Algorism
RFR
LSTM



Guidelines:-

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Interface for users to interact with the time series analysis for Bitcoin price prediction application	Web Application Framework (e.g., Flask, Django)
2.	Application Logic-1	Responsible for data collection and preprocessing of historical Bitcoin price data	Python (Pandas, NumPy), Web Scraping, APIs
3.	Application Logic-2	Manages time series decomposition, feature engineering, and model training	Python (statsmodels, TensorFlow, scikit-learn)
4.	Application Logic-3	Implements backtesting, model evaluation, and prediction generation	Python (Backtrader, custom algorithms)
5.	Database	Stores cleaned and preprocessed historical Bitcoin price data, model parameters, and evaluation metrics	Relational Database (e.g., MySQL, PostgreSQL)
6.	Cloud Database	Provides a scalable and accessible storage solution for large datasets and model artifacts	Cloud-based Database Services (e.g., AWS RDS, Google Cloud Firestore)
7.	File Storage	Stores datasets, model parameters, and other necessary files	Local File System, Cloud Storage (e.g., AWS S3)
8.	External API-1	Fetches additional data relevant to Bitcoin price prediction from external sources (e.g., economic indicators, news sentiment)	HTTP requests, RESTful APIs
9.	External API-2	Retrieves supplementary information from external APIs, such as social media sentiments or blockchain data	HTTP requests, RESTful APIs
10.	Machine Learning Model	Encompasses the trained models used for time series forecasting and prediction of Bitcoin prices	Encompasses the trained models used for time series forecasting and

			prediction of Bitcoin prices
11.	Infrastructure (Server / Cloud)	Hosts the application logic, database, and other components, either on local servers or in the cloud	On-Premises Servers, Cloud Platforms (e.g., AWS, Google Cloud)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source frameworks for flexibility, community support, and cost-effectiveness	Python (Pandas, NumPy, TensorFlow, scikit-learn), Apache Kafka, Prophet
2.	Security Implementations	Implements robust security measures to protect sensitive data, model parameters, and user information	Encryption protocols, SSL/TLS, Secure APIs
3.	Scalable Architecture	Adopts a scalable architecture to handle growing datasets and increasing computational demands	Microservices Architecture, Docker, Kubernetes
4.	Availability	Ensures high availability for the system, minimizing downtime and ensuring continuous data analysis	Load Balancing, Redundancy, Failover Mechanisms
5.	Performance	Focuses on optimizing system performance for efficient data processing and real-time analysis	Caching Mechanisms, Data Indexing, GPU Acceleration

References:

https://www.mdpi.com/1911-8074/16/1/51
https://www.jetir.org/papers/JETIR2106299.pdf
https://www.irjmets.com/uploadedfiles/paper/issue_1_january_2022/18213/final/fin_irjmets1641909207.pdf