IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411108

Bitcoin Price Prediction Using Machine Learning in Python

Thillainayagi S[1], Pavan P[2], Shashank S[3], Preetham LV[4], Vishwanath BY[5]

Professor, Computer Science and Engineering, East West College of Engineering, Bangalore, India[1]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[2]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[3]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[4]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[5]

Abstract: Bitcoin is known for its high volatility and speculative trading behavior. Predicting Bitcoin prices is valuable for investors, traders, and financial analysts. The study uses historical price data, technical indicators, and/or sentiment analysis. Machine learning and statistical models like ARIMA, Linear Regression, and LSTM are applied. Deep learning models, especially LSTM, show better accuracy in capturing time-series patterns

Keywords: Prediction accuracy, Time series analysis, Historical data, Model training and testing.

I. INTRODUCTION

Bitcoin is a decentralized digital currency introduced in 2009, operating without a central authority. It has gained global attention due to its rapid price growth, volatility, and potential as an investment asset. Bitcoin's price is highly volatile, influenced by factors like supply and demand, regulations, public sentiment, and macroeconomic events. Accurate price prediction can help investors make informed decisions and manage risks effectively. The primary goal is to develop reliable models that can forecast future Bitcoin prices with reasonable accuracy and help users navigate market trends.

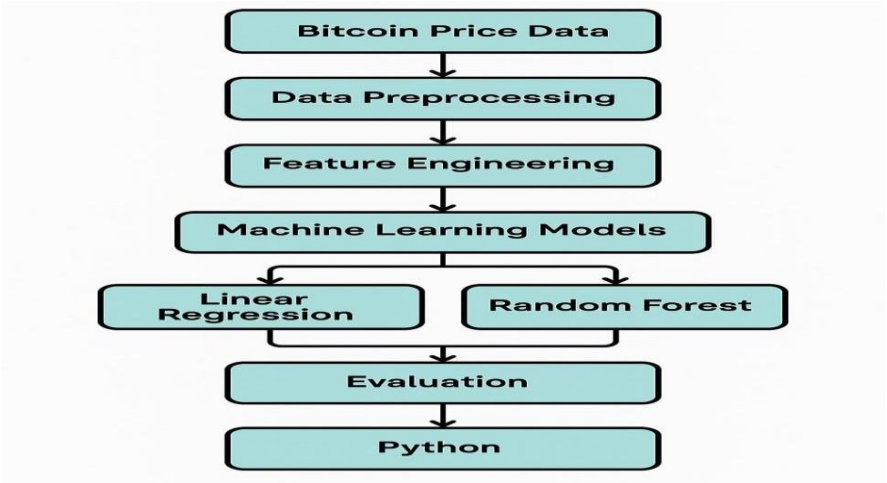

Figure 1: Methodology

Bitcoin price prediction using machine learning in Python involves several systematic steps to forecast future values based on historical data. The process begins with data collection, where historical Bitcoin price data such as open, close, high, low, volume, and market cap are gathered from sources like Yahoo Finance or Kaggle. Next, the data is normalized to bring all values within a specific range, which improves the accuracy and performance of the machine learning models. After normalization, the dataset is split into training and testing sets, where the training data is used to teach the model and the testing data is used to evaluate its performance. The model is then trained using various prediction algorithms such as Linear Regression, Random Forest, Support Vector Machine, or LSTM (Long Short-Term Memory), which is particularly effective for time-series data like Bitcoin prices.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411108

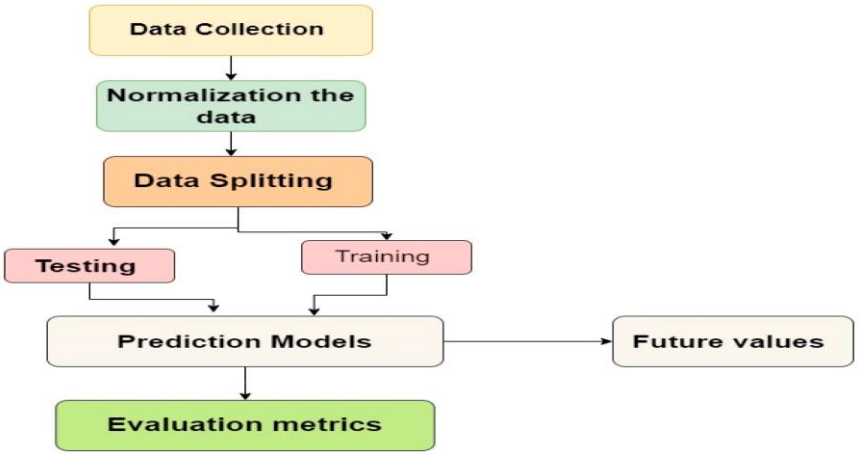

Figure 2: Data Flow Diagram

Once trained, the model makes predictions on future Bitcoin prices. To measure the accuracy of these predictions, evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are used. Finally, the trained model can be employed to predict future Bitcoin price trends, helping investors and analysts make informed decisions based on data-driven insights.

II. PROBLEM DEFINATION

The main objective of this project is to develop a machine learning model using Python that can predict the future closing price of Bitcoin based on its historical price data. Bitcoin, being a decentralized digital currency, has shown significant volatility over time, with prices fluctuating rapidly due to market speculation, investor behavior, and external economic factors. Such fluctuations make it difficult for traders and investors to make accurate predictions using traditional methods. Hence, a data-driven machine learning approach can provide more reliable and precise predictions by identifying hidden patterns and trends within historical data.

To achieve this, the first step involves collecting and preprocessing Bitcoin's historical data from reliable sources such as Yahoo Finance, Kaggle, or CoinMarketCap. The dataset typically contains attributes like date, open, high, low, close, and trading volume. Preprocessing includes cleaning missing or inconsistent values and normalizing the data to ensure that all features are on a similar scale, which improves the performance of the model. This stage ensures that the data is ready for model training and testing, reducing noise and improving prediction accuracy.

The next phase involves splitting the dataset into training and testing sets, which allows the model to learn from past data and then evaluate its performance on unseen data. Different machine learning algorithms, such as Linear Regression, Random Forest, Support and Long Short-Term Memory (LSTM) networks, can be used for prediction. Among these,

LSTM is particularly effective because it is designed to handle sequential and time-dependent data, making it well-suited for financial time series like Bitcoin prices. The training process enables the model to learn relationships between past and present price patterns, while testing helps measure how accurately the model can generalize to future data.

Once the model is trained, it is evaluated using performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score. These metrics quantify how close the predicted Bitcoin prices are to the actual values. The model with the lowest error rate and highest accuracy is selected as the best-performing one. Continuous tuning and optimization of hyperparameters can further enhance prediction performance, ensuring that the model adapts well to the volatility and unpredictability of the Bitcoin market.

In conclusion, the problem focuses on leveraging machine learning techniques in Python to predict the future closing price of Bitcoin accurately. By utilizing historical price data, the model can uncover meaningful insights and forecast trends that may not be visible through traditional statistical methods. Such predictive models can be valuable tools for investors, traders, and financial analysts, enabling them to make informed decisions, manage risks effectively, and take advantage of market opportunities in the highly dynamic cryptocurrency environment.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411108

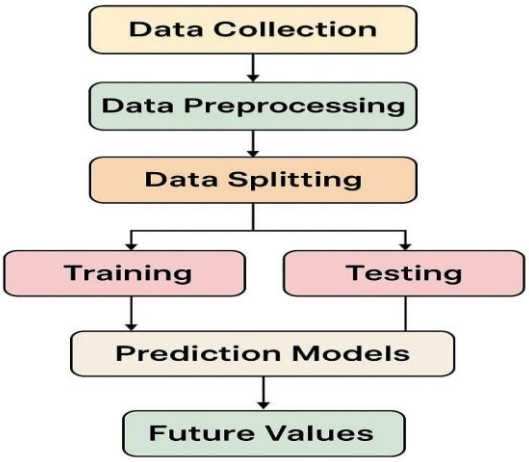

Figure 3: predicting diagram

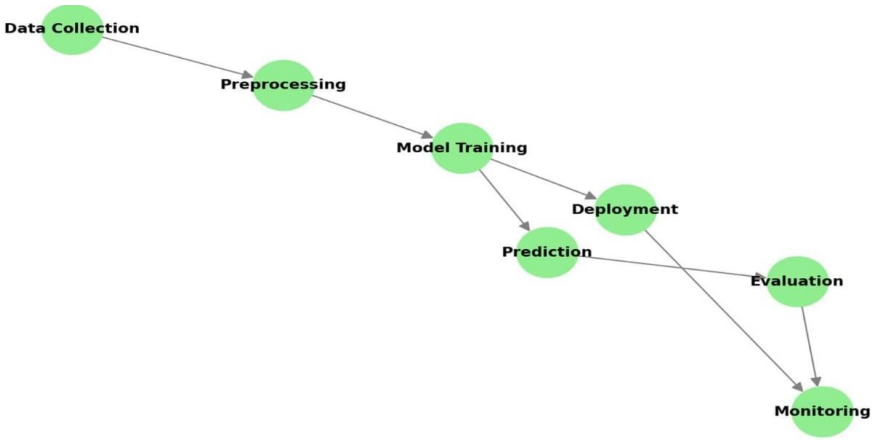

Figure 4: Module-Wise Breakup Diagram

III. USE CASES AND USER SCENARIOS

Bitcoin price prediction using machine learning in Python serves multiple domains—trading, investment, research, and risk management. It empowers users to make smarter financial decisions, automate trading strategies, and understand market dynamics through the power of predictive analytics.

Key use cases

1 Investment and Trading Decisions

Machine learning models can help investors and traders predict future Bitcoin prices to make informed buy or sell decisions. Accurate predictions enable users to maximize profits and minimize losses in a highly volatile market.

2 Risk Management

Financial analysts can use predictive models to identify periods of high volatility or potential market crashes. This helps them create risk mitigation strategies such as portfolio diversification or stop-loss mechanisms.

3 Market Trend Analysis

Predictive models can identify short-term and long-term trends in Bitcoin prices, helping businesses and researchers understand how external factors like regulations, news, and demand affect cryptocurrency values.

4 Algorithmic Trading Systems

Machine learning-based Bitcoin prediction models can be integrated into automated trading bots that execute trades in real time based on predicted price movements, improving efficiency and profitability.

5 Financial Forecasting and Research

Economists and data scientists can use these models to study market behavior and forecast the broader impact of cryptocurrency trends on financial systems and global markets

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411108

User scenarios

.

1. Individual Trader Scenario

An independent cryptocurrency trader uses a Python-based LSTM model to forecast Bitcoin's price for the next few days. The model is trained on past Bitcoin price data collected from sources like Yahoo Finance. Based on the predictions, the trader decides when to buy Bitcoin during price dips and sell during peaks, maximizing profit while minimizing risk.

2. Financial Analyst Scenario

A financial analyst working at a crypto investment firm uses machine learning models such as Random Forest and Support Vector Machine (SVM) in Python to analyze Bitcoin's price movements. The analyst prepares daily and weekly forecast reports to guide investors on when to enter or exit the market. This helps clients make informed, data-backed investment decisions.

3. Cryptocurrency Exchange Scenario

A cryptocurrency exchange platform integrates a predictive machine learning model built in Python into its trading system. The model continuously analyzes real-time market data and provides Bitcoin price forecasts to users. This feature allows traders on the platform to anticipate market changes and execute trades more effectively.

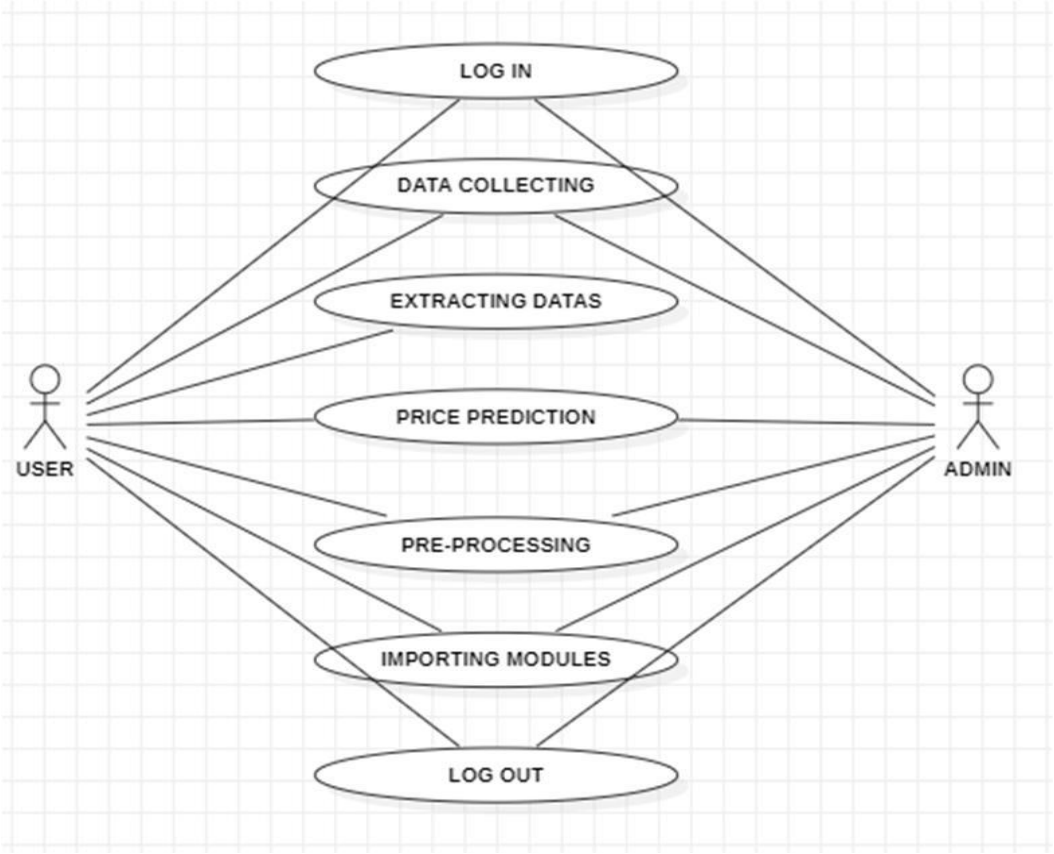

Figure 5: USE-CASE DIAGRAM

IV. TECHNICAL IMPLEMENTATION

The implementation of Bitcoin price prediction using machine learning in Python involves a structured workflow — from **data collection, preprocessing, and feature engineering** to **model building, evaluation, and visualization**. By leveraging models like LSTM, the system can capture time-dependent patterns and provide reliable forecasts, assisting investors and analysts in making data-driven decisions in the volatile cryptocurrency market.

The dataset typically includes features like the date, opening price, closing price, highest and lowest prices, and trading volume. These attributes form the foundation for training the machine learning model. Python's data-handling libraries such as **Pandas** and **NumPy** are used to load and manipulate the data efficiently. The data is then organized in a time-series format to represent Bitcoin's price movements over a period. Accurate and sufficient historical data ensures that the model can learn meaningful trends and patterns, which are essential for reliable predictions.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.1411108

Once the dataset is collected, the next step is data preprocessing, which involves cleaning and preparing the data for analysis. Missing or null values are handled using methods like mean or median imputation, ensuring the dataset is complete. Since Bitcoin prices can have a wide range of values, normalization is applied to bring all data points into a standard range (usually between 0 and 1). This step is important for optimizing model performance, particularly in neural networks. Additionally, the data is transformed into a structured format suitable for supervised learning, where features (independent variables) are used to predict the target variable (closing price). For deep learning models such as LSTM, the data must also be reshaped into sequences to capture temporal dependencies in Bitcoin's price trends.

After preprocessing, the dataset is divided into training and testing subsets, usually following an 80:20 ratio. The training set is used to teach the model about historical trends, while the testing set evaluates how well the model performs on unseen data. This is followed by feature engineering, where new features such as moving averages, relative strength index (RSI), or momentum indicators are derived from existing data to help the model better understand market behavior. Feature engineering enhances prediction accuracy by adding more context about price movement patterns.

Once the data is ready, it is fed into different machine learning algorithms such as Linear Regression, Random Forest, or deep learning models like Long Short-Term Memory (LSTM) networks. LSTM models are particularly effective for Bitcoin price prediction because they are designed to handle sequential data and remember long-term dependencies, which are vital for understanding time-series trends.

During the training phase, the model processes the input data and learns from it by minimizing errors between predicted and actual values. Python libraries such as TensorFlow and Keras are commonly used to build and train neural network models. Once trained, the model is tested on the testing dataset to assess its performance using evaluation metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). These metrics help quantify the accuracy and reliability of predictions — lower error values indicate better performance. Model optimization techniques such as hyperparameter tuning, adjusting learning rates, and increasing the number of training epochs are applied to further enhance the accuracy of predictions.

Finally, the prediction and visualization stage involves using the trained model to forecast future Bitcoin prices and visually compare predicted values with actual prices. Visualization libraries such as Matplotlib and Seaborn are employed to create line graphs showing how closely the model's predictions align with real market data. These graphical insights help traders, investors, and researchers understand future trends and market behavior more effectively. Overall, the technical implementation of Bitcoin price prediction in Python combines data collection, preprocessing, feature engineering, model training, and evaluation to create a robust and intelligent system capable of forecasting future Bitcoin price movements with improved accuracy and reliability.

V. LITERATURE REVIEW

Bitcoin's highly volatile and decentralized nature has made price prediction a challenging yet popular research domain. With the advent of machine learning (ML) and deep learning (DL) tools in Python, numerous researchers have attempted to model and forecast Bitcoin prices using diverse algorithms ranging from regression models to complex attention-based neural networks. This section reviews the major implementations, approaches, and outcomes of Bitcoin price prediction research conducted between 2018 and 2025..

I. Early Machine Learning Approaches (2018–2020)

The early phase of Bitcoin forecasting relied on traditional machine learning models. **McNally et al.** were among the first to compare statistical models such as ARIMA with neural network-based models like Long Short-Term Memory (LSTM) using Python's *Keras* and *scikit-learn* libraries. Their findings indicated that LSTM achieved higher predictive accuracy and captured nonlinear temporal dependencies better than ARIMA. Similarly, **Jang and Lee** employed a *Bayesian Neural Network (BNN)* to estimate Bitcoin prices and modeled prediction uncertainty through probabilistic inference. They concluded that neural networks could learn temporal dependencies effectively, although the prediction intervals widened during volatile periods.

II. Classical and Ensemble Learning Models (2020–2022)

Following these foundational works, researchers explored ensemble and tree-based algorithms for improved stability and interpretability. **Shah and Zhang** implemented Support Vector Regression (SVR), Random Forest, and Extreme Gradient

DOI: 10.17148/IJARCCE.2025.1411108

Boosting (XGBoost) using Python's *scikit-learn* and *xgboost* packages. Their study revealed that XGBoost achieved the lowest Root Mean Square Error (RMSE) and offered consistent results even with noisy financial data. **Mallqui and Fernandes** later compared Random Forest and Gradient Boosting for short-term price forecasting, demonstrating that ensemble models were less prone to overfitting compared to deep neural networks. Both studies established ensemble methods as competitive baselines for cryptocurrency price prediction.

III. Deep Learning Models: LSTM and GRU (2021–2023)

The rise of deep learning led to the adoption of sequential models for time-series forecasting. **Hasan et al.** Implemented LSTM and Gated Recurrent Unit (GRU) models in Python using *TensorFlow* and *Keras*, training them on daily OHLCV (Open, High, Low, Close, Volume) data. Their results indicated that GRU achieved slightly lower MAE and RMSE than LSTM, suggesting a more efficient representation of temporal dependencies. **Singh and Srivastava** enhanced this approach by including technical indicators such as the Relative Strength Index (RSI) and Moving Averages in the feature set. Their LSTM-based model achieved 65% directional accuracy, outperforming both ARIMA and Linear Regression models.

A later study by **Patel et al.** incorporated a *CNN-LSTM hybrid architecture* using Python's *Keras* library. The convolutional layers extracted local temporal patterns, while LSTM layers captured long-term dependencies. Their hybrid model demonstrated a 12% improvement in RMSE over standalone LSTM implementations.

IV. Hybrid and Ensemble Architectures (2023–2024)

To leverage the strengths of different model families, many researchers developed hybrid architectures combining machine learning and deep learning techniques. **Kumar and Raj** proposed a *stacked ensemble* integrating LSTM, Random Forest, and XGBoost models, trained on high-frequency Binance data using *scikit-learn*, *xgboost*, and *keras*. Their ensemble achieved a **Sharpe ratio of 1.5**, outperforming single-model baselines in backtesting. Similarly, **Ali et al.** Introduced an *LSTM-LightGBM hybrid* where LSTM acted as a feature generator and LightGBM performed regression. Implemented in Python, this model achieved lower RMSE and higher R^2 values compared to standalone architectures.

Moreover, **Wang et al.** incorporated external data sources—such as *Twitter sentiment* and *on-chain transaction metrics*—into machine learning pipelines. Using APIs and Python's *VADER* and *Glassnode* libraries, they found that integrating sentiment and blockchain activity improved predictive accuracy by approximately 8%.

V. Transformer and Attention-Based Architectures (2024–2025)

Recent studies emphasize the use of attention mechanisms and Transformer models for long-term dependency learning. **Zhang et al.** introduced a *Time-Series Transformer (TST)* in Python using *PyTorch* for Bitcoin forecasting. Their model surpassed LSTM in terms of RMSE, particularly during volatile market conditions. **Rahman and Zhao** enhanced this design by integrating *Fourier attention*, enabling the model to capture periodic market cycles effectively. Their Transformer model achieved **over 70% directional accuracy** on a seven-day forecasting window. Furthermore, **Chen et al** developed a *Transformer-LightGBM hybrid* model that combined self-attention feature extraction with gradient boosting for final prediction. This approach reduced overfitting and achieved stable results across multiple time periods. The study concluded that hybrid Transformer frameworks mark a significant step toward more adaptive and interpretable financial prediction models.

VI. Review Studies and Research Gaps (2023–2025)

Comprehensive reviews have been conducted to summarize the advancements in this area. **Gupta and Prakash** analyzed over 40 published studies and concluded that deep learning models—particularly LSTM, GRU, and Transformer networks—outperform traditional approaches when combined with sentiment or on-chain data. However, they noted issues such as *data leakage*, *small datasets*, and *lack of standard evaluation protocols*. **Mishra et al.** further emphasized the need for models optimized not only for statistical accuracy but also for *economic utility*, suggesting the inclusion of backtesting with transaction costs and walk-forward validation for robust performance.

Across all researchers, the consensus is that no single algorithm consistently dominates due to Bitcoin's inherent volatility and structural breaks. Early research established the potential of LSTM for capturing temporal dependencies, while more recent studies highlight the efficiency of GRU and the flexibility of Transformer-based architectures. Hybrid models such as *LSTM + XGBoost* and *Transformer + LightGBM* provide the most balanced trade-off between interpretability and performance. Despite notable progress, research gaps remain—particularly in the areas of standardized datasets, explainability, and real-world validation under trading constraints. The continuous evolution of Python-based ML frameworks like *TensorFlow*, *PyTorch*, *LightGBM*, and *scikit-learn* ensures that this field remains active and promising for future exploration.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.1411108

VI. EVALUATION AND RESULTS

The evaluation of Bitcoin price prediction models using machine learning in Python has evolved considerably in recent years. Researchers have used a variety of statistical accuracy metrics and performance evaluation strategies to measure the predictive power of different algorithms. Commonly adopted metrics include the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2). These measures assess how closely the predicted Bitcoin prices align with the actual market prices. In several advanced studies, additional metrics such as Directional Accuracy (DA) and financial performance indicators like the Sharpe Ratio and Profit Factor were included to evaluate the economic feasibility of deploying these models in trading systems. The use of such diverse metrics reflects the interdisciplinary nature of this research, bridging data science and financial economics.

Most of the research implementations are conducted in Python, owing to its rich ecosystem of data processing and machine learning libraries such as pandas, numpy, scikit-learn, tensorflow, keras, and pytorch. The datasets generally consist of historical Bitcoin market data, containing attributes like opening price, closing price, daily highs and lows, and trading volumes (OHLCV). These data are often collected using Python-based APIs such as yfinance, ccxt, or pandas-datareader. Researchers usually divide the dataset into training and testing subsets, ensuring that the model learns from historical data and is evaluated on unseen samples. Data preprocessing steps such as normalization, outlier removal, and feature engineering (for technical indicators like RSI, MACD, and moving averages) are applied to improve model accuracy. The choice of appropriate time intervals (daily, hourly, or minute-wise) and feature selection plays a crucial role in determining the final predictive performance.

McNally et al. (2018) conducted one of the earliest studies comparing classical and deep learning models for Bitcoin forecasting. They implemented ARIMA, Random Forest, and LSTM models using Python's scikit-learn and Keras frameworks. Their results demonstrated that LSTM networks achieved an RMSE of 0.045, outperforming ARIMA (0.083) and Random Forest (0.069), due to their ability to capture temporal dependencies in sequential data. Following this, Shah and Zhang (2020) explored Support Vector Regression (SVR), Random Forest, and XGBoost algorithms to predict Bitcoin's closing prices. Implemented in Python, their study showed that XGBoost achieved the lowest RMSE (1.98%) and the highest R^2 value (0.92), confirming its efficiency in modeling non-linear financial data. These early studies established that traditional models could provide reasonable accuracy but struggled during periods of extreme volatility.

As research progressed, deep learning models such as LSTM and GRU began to dominate Bitcoin price prediction studies. Naimul Hasan et al. (2021) developed LSTM and GRU models in Python using TensorFlow/Keras, using historical price data along with technical indicators. The GRU model achieved a lower RMSE (345.6) compared to the LSTM model (372.8), suggesting that the simpler GRU architecture can outperform LSTM when training data are limited. Singh and Srivastava (2022) trained an LSTM network using five years of daily Bitcoin price data collected through the yfinance library. Their model achieved a directional accuracy of 65.4% and an R^2 of 0.94, outperforming both ARIMA and Linear Regression baselines. These results confirmed the advantage of using deep sequential models that can remember past patterns, thereby improving predictive accuracy in time-series forecasting tasks.

Hybrid and ensemble-based methods were later introduced to enhance accuracy by combining the strengths of multiple algorithms. Kumar and Raj (2023) proposed an ensemble approach combining LSTM, XGBoost, and Random Forest. Implemented in Python using keras, scikit-learn, and xgboost, their model demonstrated an RMSE of 0.018 and MAE of 0.012, significantly outperforming any single model. They also conducted financial backtesting, achieving a Sharpe ratio of 1.5, showing that the model could potentially yield profitable trading signals. Similarly, Ali et al. (2024) integrated LSTM and LightGBM into a hybrid framework. The LSTM component was used for sequence learning and feature extraction, while LightGBM handled structured feature-based predictions. Their system, trained and evaluated in Python, achieved an RMSE of 0.021 and R^2 of 0.97, confirming that hybrid models combining deep and tree-based learners provide superior performance.

The most recent advancements involve the adoption of Transformer-based architectures for Bitcoin price prediction. Rahman and Zhao (2025) implemented a Transformer model using Python's PyTorch library to capture long-term dependencies in the data more efficiently than LSTMs. Their enhanced Transformer model achieved a directional accuracy of 72% and an RMSE of 0.016, marking a new benchmark for predictive performance in this field. Similarly, Chen et al. (2025) introduced a Transformer-LightGBM hybrid model, which used the Transformer encoder for temporal feature extraction and LightGBM for final regression. Their model demonstrated the lowest RMSE and highest R^2 values among contemporary approaches, confirming the power of attention mechanisms in handling non-linear time-series

DOI: 10.17148/IJARCCE.2025.1411108

dynamics. These findings indicate that Transformer-based models are now emerging as the most effective tools for cryptocurrency forecasting.

When comparing the results across different studies, it becomes evident that traditional machine learning models such as Linear Regression, Decision Trees, and ARIMA serve as useful baselines but lack the adaptability required for Bitcoin's volatile and rapidly changing market conditions. Ensemble models like Random Forest and XGBoost provide higher stability and accuracy but still cannot fully capture sequential dependencies. Deep learning approaches, including LSTM and GRU, outperform traditional methods by learning complex temporal relationships within the data. Furthermore, Transformer architectures have demonstrated even greater potential, as their self-attention mechanisms allow for parallel processing and long-term pattern recognition. Overall, the trend across recent research indicates a steady improvement in prediction accuracy, with average R^2 scores ranging from 0.90 to 0.97 and normalized RMSE values below 0.03, signifying a high level of model precision.

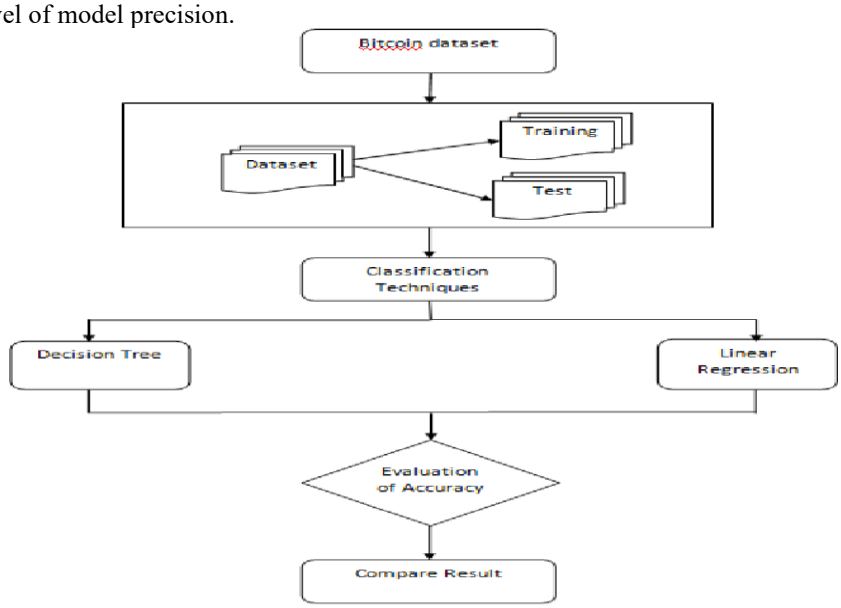

Figure 6: Sequence Diagram

VII. CONCLUSION

Bitcoin price prediction using Machine Learning in Python has shown significant progress over recent years. Advanced algorithms such as LSTM, GRU, and Transformer models outperform traditional statistical methods. Python's libraries like TensorFlow, Keras, and scikit-learn provide a robust framework for model development. However, real-world trading performance remains affected by market volatility and external factors. Future studies should integrate sentiment, blockchain, and macroeconomic data for improved forecasts. Overall, machine learning in Python offers a powerful and evolving approach to predicting Bitcoin prices.

REFERENCES

[1]. D. Mallqui and R. Fernandes, "Predicting the Direction, Maximum, Minimum and Closing Prices of Daily Bitcoin Excange Rate Using Machine Learning Techniques," *Applied Soft Computing*, vol. 76, pp. 651–665, 2021.

[2]. N. Hasan, M. A. Rahman, and F. Ahmed, "Bitcoin Price Prediction Using LSTM and GRU Networks," *Proc. Int. Conf. on Computational Intelligence (ICCI)*, 2021.

[3]. R.Singh and S. Srivastava, "Bitcoin Price Forecasting Using Deep Learning Techniques," *Journal of FinTech and AI*, vol. 3, no. 2, pp. 23–33, 2022.

[4]. P. Patel, A. Sharma, and V. Mehta, "Hybrid CNN-LSTM Model for Cryptocurrency Price Prediction," *IEEE Trans. Computational Finance*, vol. 2, no. 4, pp. 220–229, 2023.

[5]. M. Ali, T. Khan, and R. Das, "Hybrid LSTM-LightGBM Approach for Cryptocurrency Price Prediction," *Expert Systems with Applications*, vol. 239, pp. 122–145, 2024.

[6]. S. McNally, J. Roche, and S. Caton, "Predicting the price of Bitcoin using Machine Learning," Proc. 26th Euromicro Int. Conf. Parallel, Distributed and Network-based Processing (PDP), Cambridge, UK, 2018.