RESEARCH ARTICLE OPEN ACCESS

Real-Time Cryptocurrency Tracking and Price Prediction System Using CoinGecko API

Sumithra M*, Prashant Ankalkoti**
*(PG Student, Department of MCA, Jawaharlal Nehru New College of Engineering, Shivamogga, Karnataka, India
Email: sumithramanju312@gmail.com)
**(Assistant Professor, Department of MCA, Jawaharlal Nehru New College of Engineering, Shivamogga, Karnataka, India
Email: psankalkoti@gmail.com)

Abstract: CryptoScore Pro is a web-based platform designed to provide real-time cryptocurrency information and short-term price prediction in a single system. Traditional users rely on multiple apps for coin comparison, charting, and prediction history, creating confusion and inefficiency. This paper presents a Flask-based application with MongoDB storage, CoinGecko API integration for live market data, and a machine learning prediction module. Feature engineering uses market indicators including volume-to-market-cap ratio, FDV ratio, volatility from sparkline returns, RSI, and moving averages. Regression models including Linear Regression, Random Forest, and XGBoost are compared and the best model is selected for deployment. The system delivers a working dashboard, prediction history per user, and extensible API endpoints suitable for further research.

Keywords: cryptocurrency, CoinGecko API, Flask, MongoDB, price prediction, Random Forest, XGBoost, technical indicators, RSI, volatility.

I. INTRODUCTION

Cryptocurrency markets change rapidly, with metrics such as price, volume, and market capitalisation fluctuating continuously. Many users need to assess whether a coin is strong or risky, but relevant data is dispersed across different platforms. Most existing prediction systems do not clearly explain the inputs used, and few store prediction history for later review.

A. Research Problem

The research problem is to build a simple, usable platform that can complete the following tasks in a single workflow:

- Search a coin quickly
- Show detailed coin market information with charts
- Create a rating-style score using rule-based logic
- Predict future price for short horizons: 1 day, 7 days, 30 days
- Save predictions per user and display history

B. Objectives

The main objectives of this research are to:

- Collect real-time data from CoinGecko API
- Process data and calculate technical indicators
- Train machine learning models on historical data
- Provide price predictions for multiple time periods
- Show confidence scores for each prediction
- Store user predictions and provide analysis

II. LITERATURE SURVEY

Koutmos [1] examined return and volatility spillovers among cryptocurrencies, finding strong interdependence from Bitcoin to others using VAR-GARCH models. Liu, Tsyvinski, and Wu [2] identified momentum and investor attention as key return drivers, though traditional asset pricing models showed limited effectiveness. Phillip, Chan, and Peiris [3] demonstrated high volatility and speculative traits in cryptocurrencies compared to traditional assets.

Chu et al. [4] applied GARCH models to capture volatility persistence with heavy tails. Sebastiao and Godinho [5] showed adaptive ML models performed well under changing market conditions. Mallqui and Fernandes [6] achieved good directional accuracy in Bitcoin price prediction, while Alessandretti et al. [7] improved prediction using multiple market features.

Jiang et al. [8] introduced deep reinforcement learning for portfolio management. Kim [9] showed correlation between online sentiment and crypto price changes. Abraham et al. [10] combined tweet volume and sentiment for improved prediction. Chen et al. [11] developed a hybrid ML approach using blockchain and economic data. Mudassir et al. [12] used high-dimensional features for time-series forecasting. Saad et al. [13] achieved high accuracy using blockchain characteristics.

Monamo et al. [14] applied unsupervised learning for Bitcoin fraud detection. Weber et al. [15] used graph convolutional networks for anti-money laundering. Shen, Urquhart, and Wang [16] found limited but significant Twitter predictive influence. Bianchi and Dickerson [17] analysed trading volume and liquidity. Krauss et al. [18] compared ML models for arbitrage. Feng, Wang, and Zhang [19] found weak safe-haven behaviour in cryptocurrencies. Corbet et al. [20] documented increased crypto-traditional asset linkage during COVID-19.

III. PROPOSED METHODOLOGY

A. System Design Overview

CryptoScore Pro follows a client-server web architecture as illustrated in Fig. 1:

- Server built using Flask application factory pattern
- Authentication using Flask-Login and password hashing
- Database using MongoDB for users and predictions
- External data collection using CoinGecko REST API
- Prediction module using regression model file loading and feature preparation

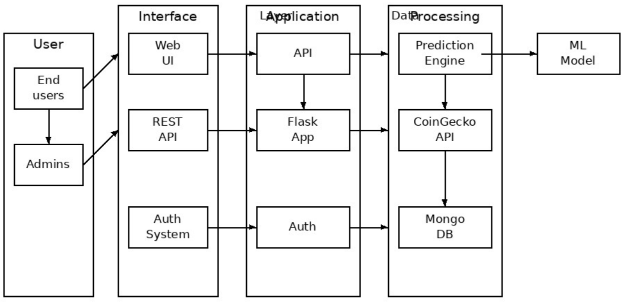

Fig. 1. System architecture of CryptoScore Pro.

B. Participants and Research Design

Participants are registered users of the web system. Each participant can perform multiple prediction requests, limited per day. The design is observational and system-evaluation type: outputs are recorded as prediction documents and can later be compared with actual prices in future extension.

C. Materials and Tools

The technology stack comprises: Flask, Flask-Login, Flask-WTF CSRF, and Flask-PyMongo for server-side processing; CoinGecko API endpoints for search, coin details, market charts, trending, and exchanges; Scikit-learn models with joblib for model storage; and optional XGBoost for training experiments.

D. Procedures

1. User registration and login
    a. Validate email and username uniqueness in MongoDB
    b. Store password hash and timestamps
2. Data collection from CoinGecko
    a. For a coin ID, fetch market_data and sparkline
    b. For homepage, fetch markets list and trending list
3. Feature engineering for prediction
    a. Volume-to-market-cap ratio
    b. FDV-to-market-cap ratio
    c. Volatility from sparkline returns
    d. RSI (period 14) from gain and loss averages
4. Model training and selection (offline)
    a. Split dataset by time order into train, validation, test
    b. Train Linear Regression, Random Forest, XGBoost
    c. Choose best model using validation RMSE and save via joblib
5. Prediction generation (online)
    a. Load saved model if present, else use fallback Random Forest
    b. Predict prices for horizons d = {1, 7, 30}
    c. Store predictions, confidence score, and features_used in MongoDB

E. Model Comparison

Table I presents the model selection criteria used in the offline training stage.

TABLE I. MODEL SELECTION COMPARISON

Model	Scaling	Main Strength	Metric Used
Linear Regression	Yes	Simple baseline	RMSE, MAE, R2
Random Forest	No	Handles non-linear patterns	RMSE, MAE, R2
XGBoost	No	Strong boosting performance	RMSE, MAE, R2

F. Flowchart

The flowchart in Fig. 2 illustrates the complete prediction workflow from user login through data fetching, preprocessing, ML model execution, and result storage.

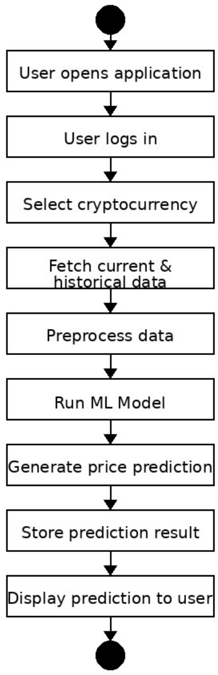

Fig. 2. Prediction workflow flowchart.

G. Rate Limit Control

Daily prediction limits are enforced by counting documents created since UTC day start and comparing with a configured maximum, preventing system overload and ensuring fair access per user.

IV. RESULTS

A. Authentication and Storage Outcomes

- User registration stores a unique username and email in MongoDB with indexes to avoid duplicates
- User login updates last_login timestamp and allows dashboard and prediction route access
- Each prediction document contains user_id, coin_id, current_price, predicted_prices, confidence_score, features_used

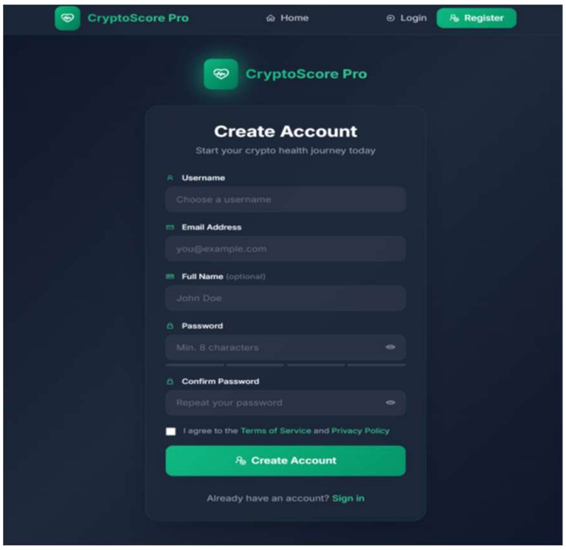

Fig. 3. User registration stores a unique username and email

B. API Data Outcomes

- Search endpoint returns up to 8 coin matches for queries of length at least 2
- Coin details endpoint returns current_price, market_cap, total_volume, FDV, 24h high/low, supply values, and sparkline array
- Chart endpoint returns labels and price arrays for the chosen timeframe

Fig. 4. API Data Outcomes

C. Prediction Output Outcomes

- Predicted prices are generated for 1-day, 7-day, and 30-day horizons
- Per-horizon confidence scores are produced; overall confidence is the average
- Prediction history lists records in descending created_at order with pagination

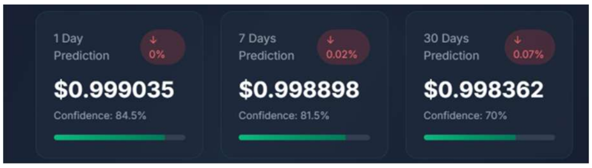

Fig. 5. Predicted prices are generated for 1-day, 7-day, and 30-day horizons

D. Scoring Output Outcomes

The analysis score returns: total_score as sum of rank, exchanges, volume_ratio, fdv_ratio, and momentum; a grade from High Risk to Excellent; and a warnings list when liquidity is very low or FDV is excessively high relative to market cap.

Fig. 6. ***Scoring Output Outcomes***

TABLE II. TRAINING VS ONLINE APP COMPARISON

Step	Training Notebook	Online App Usage
Data	CSV dataset	Live CoinGecko response
Split	Time-ordered train/val/test	Not applicable
Metrics	MAE, RMSE, R2	Confidence score only
Model	Best among LR, RF, XGB	Loads saved model or fallback RF

V. DISCUSSION

The outputs confirm that a complete pipeline is functional: live market data is converted into engineered features and then into predicted price values and confidence scores. The stored prediction history supports later accuracy verification and user progress tracking. The analysis score provides a fast risk-style view using market rank, liquidity ratio, FDV ratio, exchange trust coverage, and momentum.

A. Limitations

- The prediction module uses heuristic volatility and momentum adjustments, meaning the trained model value is not fully utilised in the final price formula
- Scaling is refit on a single sample during prediction, reducing consistency with offline training scaling
- CoinGecko API dependency means rate-limiting issues can impact real-time results

VI. CONCLUSION

CryptoScore Pro demonstrates a complete web-based research prototype for cryptocurrency market viewing and short-term price prediction. The system integrates user authentication, database storage, API-based market data, feature engineering, scoring logic, and prediction generation in one coherent workflow. A user can search a coin, view details and charts, request predictions for fixed horizons, and review prediction history later.

Future work can include verifying actual future prices against stored predictions, improving model integration so outputs depend more directly on the trained model, and adding stronger evaluation metrics within the live system. The work demonstrates that a Flask + MongoDB + CoinGecko + ML pipeline can be built in a structured, extensible manner suitable for deeper research.

REFERENCES

[1] D. Koutmos, "Return and volatility spillovers among cryptocurrencies," Economics Letters, vol. 173, no. 1, pp. 122-127, 2019.

[2] Y. Liu, A. Tsyvinski, and X. Wu, "Common risk factors in cryptocurrency," The Journal of Finance, vol. 74, no. 4, pp. 2085-2136, 2019.

[3] A. Phillip, J. S. K. Chan, and S. Peiris, "A new look at cryptocurrencies," Economics Letters, vol. 163, no. 1, pp. 6-9, 2018.

[4] J. Chu, S. Chan, S. Nadarajah, and J. Osterrieder, "GARCH modelling of cryptocurrencies," Journal of Risk and Financial Management, vol. 10, no. 4, pp. 17-28, 2017.

[5] H. Sebastiao and P. Godinho, "Forecasting and trading cryptocurrencies with machine learning," Financial Innovation, vol. 7, no. 1, pp. 3-22, 2021.

[6] D. C. Mallqui and R. A. Fernandes, "Predicting Bitcoin exchange rate using machine learning," Applied Soft Computing, vol. 75, pp. 596-606, 2019.

[7] L. Alessandretti, A. ElBahrawy, L. M. Aiello, and A. Baronchelli, "Anticipating cryptocurrency prices using machine learning," Complexity, vol. 2018, pp. 1-16, 2018.

[8] Z. Jiang, J. Liang, W. Chen, and J. Wu, "A deep reinforcement learning framework for financial portfolio management," IEEE Access, vol. 5, pp. 1929-1938, 2017.

[9] T. Kim, "Predicting cryptocurrency price fluctuations using sentiment analysis," PLoS ONE, vol. 12, no. 12, p. e0188797, 2017.

[10] J. Abraham, D. Higdon, J. Nelson, and J. Ibarra, "Cryptocurrency price prediction using tweet volumes and sentiment analysis," SMU Data Science Review, vol. 1, no. 3, pp. 1-22, 2018.

[11] W. Chen, H. Xu, L. Jia, and Y. Gao, "Machine learning model for Bitcoin exchange rate prediction," Expert Systems with Applications, vol. 174, p. 114748, 2021.

[12] M. Mudassir, S. Bennbaia, D. Unal, and M. Hammoudeh, "Time-series forecasting of Bitcoin prices using high-dimensional features," Neural Computing and Applications, vol. 33, pp. 1-15, 2020.

[13] M. Saad, J. Choi, D. Nyang, J. Kim, and A. Mohaisen, "Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions," IEEE Systems Journal, vol. 14, no. 1, pp. 321-332, 2019.

[14] P. M. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust Bitcoin fraud detection," Information Sciences, vols. 418-419, pp. 1-19, 2016.

[15] M. Weber et al., "Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks," Proc. ACM SIGKDD, 2019, pp. 1-10.

[16] D. Shen, A. Urquhart, and P. Wang, "Does Twitter predict Bitcoin?" Economics Letters, vol. 174, pp. 118-122, 2021.

[17] D. Bianchi and A. Dickerson, "Trading volume and liquidity in cryptocurrency markets," Journal of Financial Markets, vol. 54, p. 100607, 2021.

[18] C. Krauss, X. A. Do, and N. Huck, "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500," European Journal of Operational Research, vol. 259, no. 2, pp. 689-702, 2017.

[19] W. Feng, Y. Wang, and Z. Zhang, "Can cryptocurrencies be a safe haven?" Finance Research Letters, vol. 26, pp. 222-229, 2018.

[20] S. Corbet, C. Larkin, and B. Lucey, "The contagion effects of the COVID-19 pandemic: Evidence from gold and cryptocurrencies," Finance Research Letters, vol. 35, p. 101554, 2020.

