Journal of Soft Computing Paradigm (ISSN: 2582-2640)
www.irojournals.com/jscp/

Bitcoin Price Prediction Using Machine Learning Technique

Sujith V.[1], Sree Yogeswari S.[2], Subha Sri R.[3]

[1]Assistant Professor, [2,3] Student, Department of Computer Science and Engineering,

NPR College of Engineering and Technology, Dindigul, India.

E-mail: [1]sujitha@gmail.com, [2]asyogaprabha@gmail.com,[3]subasri@gmail.com

Abstract

Bitcoin price forecasting is challenging because of nonlinear market behaviour and rapid price fluctuations. This study presents an integrated machine-learning framework for one-step-ahead Bitcoin price prediction using Random Forest (RF), Support Vector Machine (SVM), and XGBoost. The framework incorporates preprocessing, comparative model evaluation, prediction fusion, and directional trend classification within a unified predictive workflow. Model performance is assessed using MAE, MSE, RMSE, and R^2, while an equal-weight fusion mechanism combines the individual prediction outputs. A separate RF classifier generates UP, DOWN, and NEUTRAL market-direction states. Experimental results demonstrate that XGBoost provides the strongest predictive performance, achieving a reported accuracy of 96.5% and an R^2 of 0.94, with lower prediction errors than the other evaluated models. The complete framework is deployed through a Flask-based web application supporting interactive prediction, graphical visualization, trend analysis, and prediction-history management.

Keywords: Bitcoin Price Forecasting, XGBoost Regression, Support Vector Regression, Random Forest Ensemble, Prediction Fusion, Market Trend Classification, Flask-Based Predictive Analytics.

1. Introduction

Bitcoin has emerged as a prominent digital asset whose market value is characterized by substantial volatility and rapid price variation. Unlike conventional financial assets governed by centralized institutions, Bitcoin operates within a decentralized ecosystem in which price

DOI: https://doi.org/10.36548/jscp.2026.4.005
Extended version of article received from **6th International Conference on Smart Innovations in Engineering Technology (ICSIET 2026)**
Received: 17.08.2026, received in revised form: 19.09.2026, accepted: 30.09.2026, published: 17.10.2026
© 2026 Inventive Research Organization. This is an open access article under the Creative Commons Attribution-NonCommercial International (CC BY-NC 4.0) License

movements are influenced by changing market demand, investor behaviour, regulatory conditions, and broader economic factors. These characteristics produce nonlinear and highly dynamic market patterns, making accurate price forecasting a challenging computational problem. The availability of historical market information, including Open, High, Low, Close, and Trading Volume, provides an opportunity to investigate these patterns through data-driven machine-learning approaches.

Traditional forecasting approaches based on simplified linear assumptions may have limitations when applied to rapidly changing cryptocurrency markets. Machine-learning models provide an alternative by learning relationships between historical market variables and subsequent price values without requiring an explicitly defined linear relationship. However, the use of an individual prediction model can restrict the representation of different nonlinear patterns within the market data. In addition, a prediction system intended for practical use requires more than offline model estimation; data preprocessing, comparative model evaluation, prediction visualization, user interaction, and real-time access are also required. These limitations motivate the development of an integrated prediction framework combining multiple supervised learning models with an operational web-based interface.

The proposed system addresses these requirements through a machine-learning framework incorporating Random Forest, Support Vector Machine, and XGBoost for Bitcoin price prediction. Historical market observations are subjected to preprocessing before being supplied to the predictive models, and their outputs are evaluated using MAE, MSE, RMSE, and R^2. The framework additionally incorporates prediction fusion and a Random Forest-based trend-classification component to provide both continuous price estimation and directional market information. The resulting prediction workflow is integrated into a Flask-based web application, enabling market parameters to be processed and prediction results to be visualized through an interactive interface.

2. Related Works

2.1 Machine Learning-Based Cryptocurrency Prediction

Early machine-learning approaches to Bitcoin forecasting established the applicability of supervised learning to historical cryptocurrency market data. Conventional machine-learning

models have been investigated for learning nonlinear relationships between market variables and future Bitcoin prices, demonstrating the potential of data-driven forecasting over purely statistical approaches [1], [2], [3]. Ensemble-based strategies have further been explored to improve predictive stability by combining multiple learners and reducing the limitations associated with individual estimators [4], [5]. These studies indicate that model selection and feature representation play an important role in capturing the highly variable behavior of cryptocurrency markets.

2.2 Ensemble and Gradient-Boosting Approaches

Tree-based ensemble methods have received considerable attention because of their ability to model nonlinear relationships and interactions among financial variables. Random Forest provides an ensemble of decision trees that can improve robustness through aggregated predictions, while gradient-boosting approaches progressively minimize prediction errors through sequential weak learners [6], [7]. XGBoost extends this boosting principle with an optimized implementation suitable for large-scale learning and has been applied to cryptocurrency forecasting tasks [2], [4]. These approaches provide an effective basis for comparing individual learners and constructing combined prediction mechanisms.

2.3 Support Vector and Time-Series-Based Prediction

Support Vector Regression provides another approach for nonlinear financial forecasting by mapping input observations into higher-dimensional feature spaces and minimizing prediction error under regularization constraints [8]. Time-series forecasting methods provide complementary mechanisms for modelling temporal dependencies and sequential market behavior [9]. However, cryptocurrency prices are affected by rapidly changing market conditions, making the direct representation of temporal and nonlinear relationships an important consideration when developing predictive systems.

2.4 Deep Learning-Based Forecasting

Deep-learning approaches have also been investigated for financial and cryptocurrency prediction because sequential neural architectures can represent temporal dependencies within historical observations. Long Short-Term Memory networks are particularly relevant to sequential forecasting because their recurrent structure enables information from previous observations to influence subsequent predictions [10]. General deep-learning frameworks further provide flexible

architectures for learning complex representations from high-dimensional data, although their computational requirements and training complexity can increase compared with conventional tree-based models [5], [10].

2.5 Research Gap

The reviewed research demonstrates the applicability of machine learning, ensemble learning, support-vector methods, time-series techniques, and deep-learning architectures to cryptocurrency forecasting [1], [2], [4]. However, the existing approaches considered in this study primarily emphasize individual prediction models or specific algorithmic strategies rather than integrating multiple tree-based regressors, prediction fusion, market-trend classification, and an interactive deployment framework within a single operational system. In addition, practical prediction platforms require mechanisms for data acquisition, preprocessing, model execution, result visualization, and prediction-history management rather than limiting the contribution to offline model evaluation [3], [6], [9]. Therefore, the proposed framework integrates complementary machine-learning predictors with a fusion mechanism and Flask-based application layer to provide an end-to-end Bitcoin price prediction workflow.

3. Proposed Methodology

3.1 System Overview and End-to-End Workflow

The proposed system is developed as an end-to-end machine-learning framework for Bitcoin price prediction, integrating market-data acquisition, data preprocessing, predictive model development, prediction fusion, trend analysis, and web-based deployment. Historical Bitcoin market observations consisting of open, high, low, close, and trading-volume attributes are first acquired and prepared for model processing. The processed data are subsequently supplied to multiple machine-learning models to generate individual price predictions. The proposed framework incorporates Random Forest, Decision Tree, and XGBoost-based prediction components, followed by a fusion mechanism for obtaining the consolidated price prediction. In parallel, a classification component is incorporated to provide directional market information. The resulting predictive outputs are integrated into a Flask-based web application, where prediction results can be accessed, visualized, and stored for subsequent analysis.

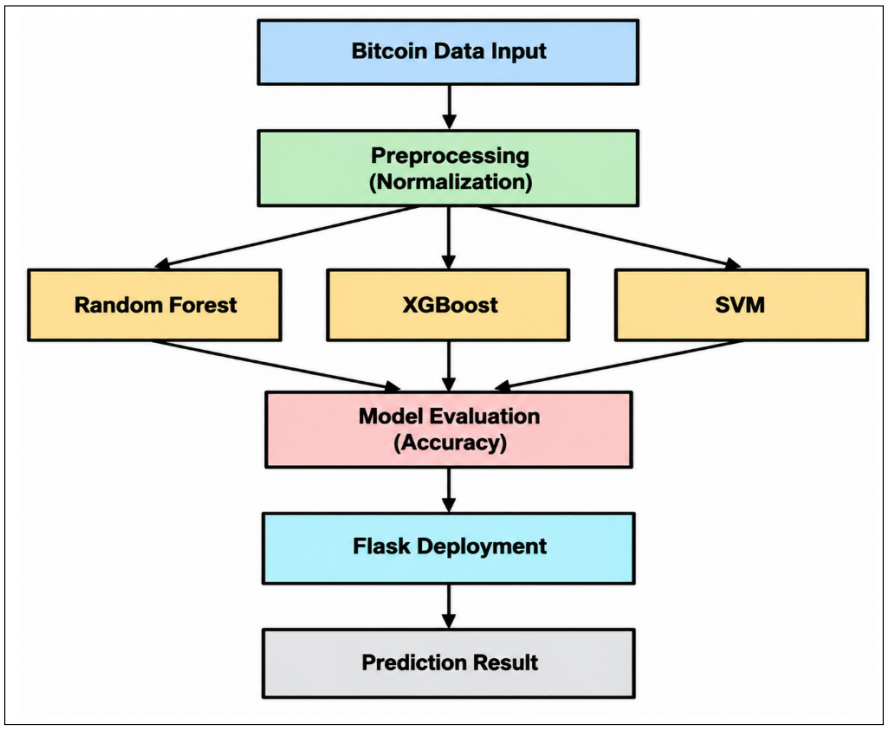

Figure 1. End-to-End machine learning framework for bitcoin price prediction

As illustrated in Figure 1, the framework follows a sequential processing structure in which market data are transformed into predictive outputs through successive data preparation, model execution, and application-integration stages. The architecture separates the analytical processing from the user-facing application while maintaining a continuous flow from input data to prediction output. This organization provides the basis for the subsequent dataset preparation, feature construction, model development, fusion, and deployment procedures described in the following subsections.

3.2 Dataset Acquisition and Preprocessing

The dataset used for model development consists of five principal Bitcoin market attributes: Open, High, Low, Close, and Trading Volume. Historical observations are acquired through the Yahoo Finance API [11], providing the market variables required for regression-based Bitcoin price prediction. The Close variable is treated as the principal price variable, while Open, High, Low, and Volume provide complementary market information for model input. The dataset structure is therefore represented by four price-related variables and one trading-activity variable, forming a 5-dimensional market feature representation [11].

Prior to model development, the collected observations undergo three preprocessing

operations: missing-value handling, normalization, and feature selection. Missing observations are treated before model execution to prevent incomplete records from affecting the learning process. Normalization is subsequently applied to the numerical attributes to establish a comparable scale among the input variables. Feature selection is then performed to retain the relevant market attributes for predictive modelling. These operations produce the final feature matrix supplied to the machine-learning models.

For experimental evaluation, the available observations are divided using an 80:20 training-testing configuration. The training portion is used for model development, whereas the held-out portion is used to evaluate predictive performance. Model performance is assessed using four regression measures: MAE, MSE, RMSE, and R^2.

3.3 Feature Representation and Target Construction

The processed market observations are represented as a feature vector for machine-learning prediction. For an observation at time t_r, the input representation is defined as

$$X_t = \left[O_t, H_t, L_t, C_t, V_t\right] \tag{1}$$

where O_t, H_t, L_t, C_t, and V_t denote the Open, High, Low, Close, and Trading Volume, respectively. The corresponding target variable represents the subsequent Bitcoin price used for the regression task.

In parallel with continuous price estimation, a separate classification component is incorporated to identify the market direction using three categorical states: UP, DOWN, and NEUTRAL. A Random Forest classifier is used for the trend-classification task.

3.4 Machine Learning Prediction Models

Three regression models are considered for continuous Bitcoin price estimation: Random Forest (RF), Decision Tree (DT), and XGBoost (XGB). The models provide independent estimates of the target price, allowing their predictive behaviour to be examined under the same feature representation. The manuscript specifies RF, DT, and XGB as the regression components for next-day price prediction.

3.4.1 Random Forest Regression

Random Forest regression constructs an ensemble of decision trees and combines their individual outputs to obtain a continuous prediction. In the proposed framework, RF is used to model nonlinear relationships among the Bitcoin market variables. Its ensemble structure provides multiple tree-based estimates rather than relying on a single decision path. The manuscript reports that the RF model is also used separately as the trend-classification component.

3.4.2 Decision Tree Regression

Decision Tree regression estimates the target price through a sequence of feature-based decision rules. Each internal node applies a splitting condition, while terminal nodes provide the corresponding regression value. Within the proposed multi-model configuration, the Decision Tree provides an independent price estimate that can subsequently be considered alongside the other regression outputs.

3.4.3 XGBoost Regression

XGBoost is employed as a gradient-boosting regression model. It builds a sequence of decision trees in which subsequent trees are used to improve the errors produced by preceding learners. The manuscript incorporates XGBoost as one of the principal prediction models and reports its performance using MAE, MSE, RMSE, and R^2.

3.5 Prediction Fusion and Trend Classification

3.5.1 Prediction Fusion

The multi-model configuration generates three independent continuous estimates for the same prediction instance through Random Forest, Decision Tree, and XGBoost regression. Let the outputs of the three regressors be P_{RF}, P_{DT}, and P_{XGB}, respectively. A simple arithmetic fusion mechanism is applied to these outputs:

$$P_F = \frac{P_{RF}\ P_{DT}\ P_{XGB}}{3} \tag{2}$$

where P_F denotes the fused Bitcoin price estimate. The equal-weight formulation assigns identical contribution to each regression model and avoids introducing an additional parameter for model weighting. Consequently, the final continuous prediction represents the central estimate

of the three independently generated model outputs. The implementation specifically describes averaging as the mathematical operation used to combine the RF, DT, and XGBoost predictions.

The fusion mechanism is applied after individual model inference rather than during model training. Hence, each regressor retains its independent decision structure, while the averaging stage operates only on their numerical prediction outputs. This provides a single price estimate for subsequent presentation through the application.

3.5.2 Trend Classification

The framework additionally performs a separate classification operation to represent the direction associated with the predicted price. A Random Forest classifier is used independently from the regression models, producing three discrete market-direction states: UP, DOWN, and NEUTRAL.

This dual-output configuration separates continuous price estimation from directional categorization. The regression branch produces the numerical price value, whereas the classification branch converts the market condition into a discrete directional state. Therefore, the system provides both a quantitative prediction and a categorical trend indication rather than treating price estimation and trend identification as a single modelling task.

3.6 System Architecture and Application Integration

The proposed system follows a layered architecture integrating the presentation layer, Flask application layer, machine-learning layer, and MySQL data-storage layer. The presentation layer manages user interaction and prediction visualization, while the Flask application layer coordinates input processing and model execution. The machine-learning layer contains the RF, DT, and XGBoost regression models together with the RF classification component. MySQL maintains user, dataset, and prediction-related records. Yahoo Finance integration provides access to market data for prediction operations.

The architecture establishes a direct flow between user input, predictive processing, model output, and result presentation. Prediction requests received through the web interface are processed by the Flask backend, where the trained models generate price estimates and directional outputs. The processed results are returned to the interface for visualization and are also associated with the corresponding prediction records in the database. The architectural arrangement is presented in Figure 2.

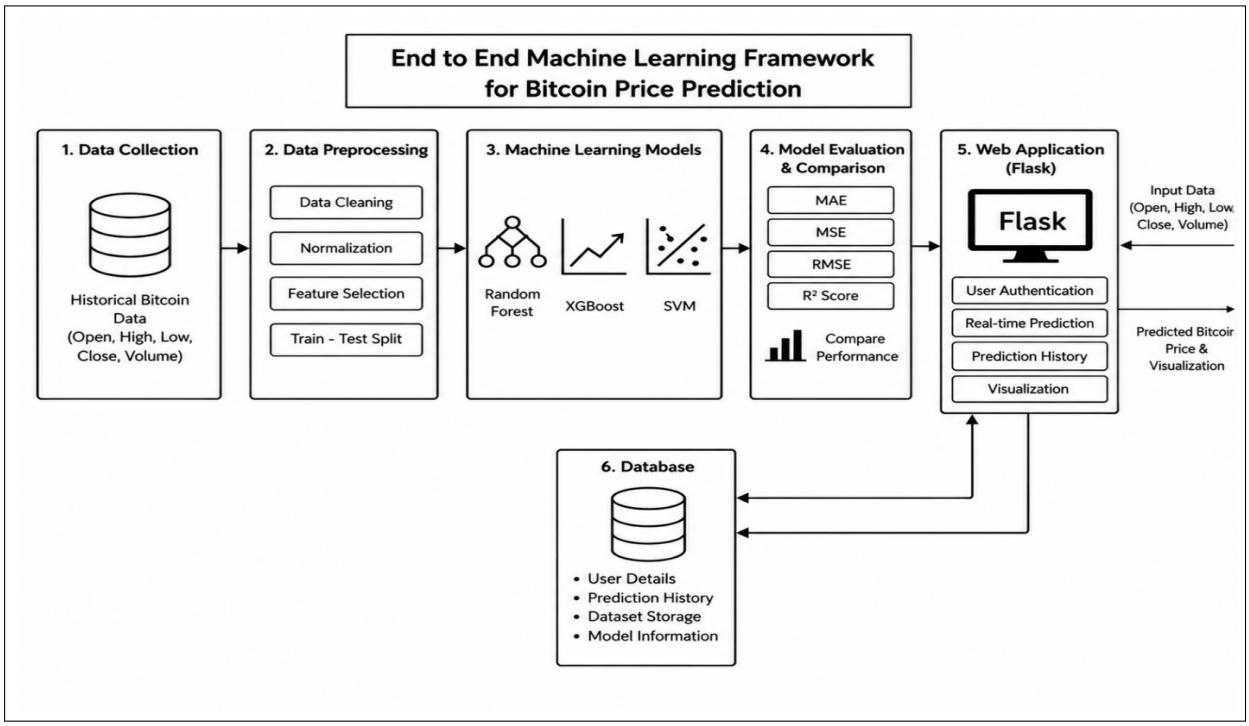

Figure 2. Web application architecture of the proposed bitcoin price prediction system

3.7 Web Application, Database and Operational Implementation

The proposed system is implemented using Python and Flask, with HTML/CSS and Flask templates providing the web interface. The application supports user registration, authentication, Bitcoin parameter input, prediction generation, graphical visualization, and prediction-history management.

The backend coordinates the submitted market parameters with the trained prediction models and manages the resulting outputs. MySQL is used for persistent storage of user registration details, uploaded market data, and prediction-analysis records. The application also incorporates Yahoo Finance retrieval for obtaining market data and supports PDF report generation for prediction results.

The operational flow consists of data acquisition, input processing, model execution, prediction fusion, result visualization, and record storage. As illustrated in Figure 3, these components are integrated into a layered implementation structure that connects the user interface with the application backend, prediction models, and persistent data storage. This integration enables the trained models to operate through a web interface rather than as an isolated prediction script.

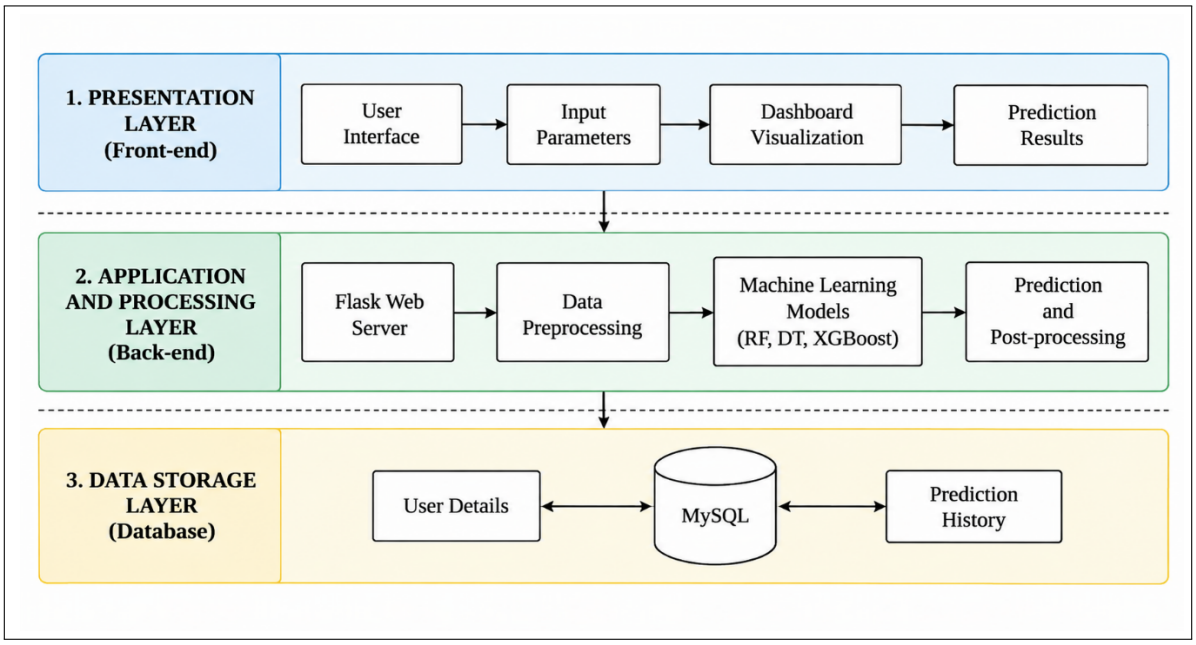

Figure 3. Layered architecture of the flask-based bitcoin price prediction framework

3.8 Performance Evaluation

The predictive models are evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score. These metrics are used to quantify prediction error and the agreement between predicted and observed Bitcoin prices. Evaluation is performed on the reserved 20% test portion of the dataset, which is kept separate from model development.

The evaluation process compares the regression outputs generated by the implemented models using the same test observations and evaluation criteria. Lower MAE, MSE, and RMSE values indicate reduced prediction error, while R^2 represents the proportion of variation in the target variable accounted for by the regression model.

4. Results and Discussion

4.1 Experimental Configuration and Evaluation Setup

The proposed Bitcoin price prediction system is evaluated using historical market observations consisting of Open, High, Low, Close, and Trading Volume attributes. Three machine-learning models, namely Random Forest (RF), Support Vector Machine (SVM), and XGBoost (XGB), are considered for comparative prediction. The dataset is divided into training 80% and testing 20% portions, with of the observations reserved as the test set for evaluating

model performance on unseen data.

The predictive performance is evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score. These metrics provide complementary measures of prediction error and regression fit, while the reported accuracy percentage is used for comparative presentation of the implemented models. The same evaluation configuration is applied to the individual models to maintain consistency in performance comparison. The obtained results are analysed based on both error magnitude and the agreement between predicted and observed Bitcoin prices.

4.2 Comparative Model Performance

The comparative results of the three prediction models are presented in Table 1. The evaluation shows clear differences in predictive behaviour across Random Forest, SVM, and XGBoost. XGBoost exhibits the most favourable error profile and the strongest regression fit among the evaluated models, indicating better alignment between the predicted and observed Bitcoin prices. Random Forest demonstrates comparatively stable performance, while SVM shows relatively higher prediction error under the same test configuration.

The observed performance variation can be associated with the different learning mechanisms employed by the models. The ensemble structure of Random Forest provides multiple tree-based estimates, whereas SVM models the feature–target relationship through its kernel-based formulation. XGBoost progressively refines tree-based predictions through gradient boosting, enabling it to capture nonlinear relationships within the market features.

Table 1. Comparative performance of the machine learning models

Model	MAE	MSE	RMSE	R^2	Accuracy (%)
Random Forest	2.15	8.45	2.90	0.91	94.2
SVM	2.48	9.72	3.11	0.88	92.8
XGBoost	1.95	7.10	2.66	0.94	96.5

4.3 Actual–Predicted Price Analysis

The relationship between the observed and predicted Bitcoin prices is illustrated in Figure 4. The comparison provides a direct view of how closely the model-generated price estimates follow the actual market values across the evaluated observations. The proximity between the predicted and actual price trajectories indicates the consistency of the regression output, while deviations between the two curves represent prediction errors associated with variations in the market data.

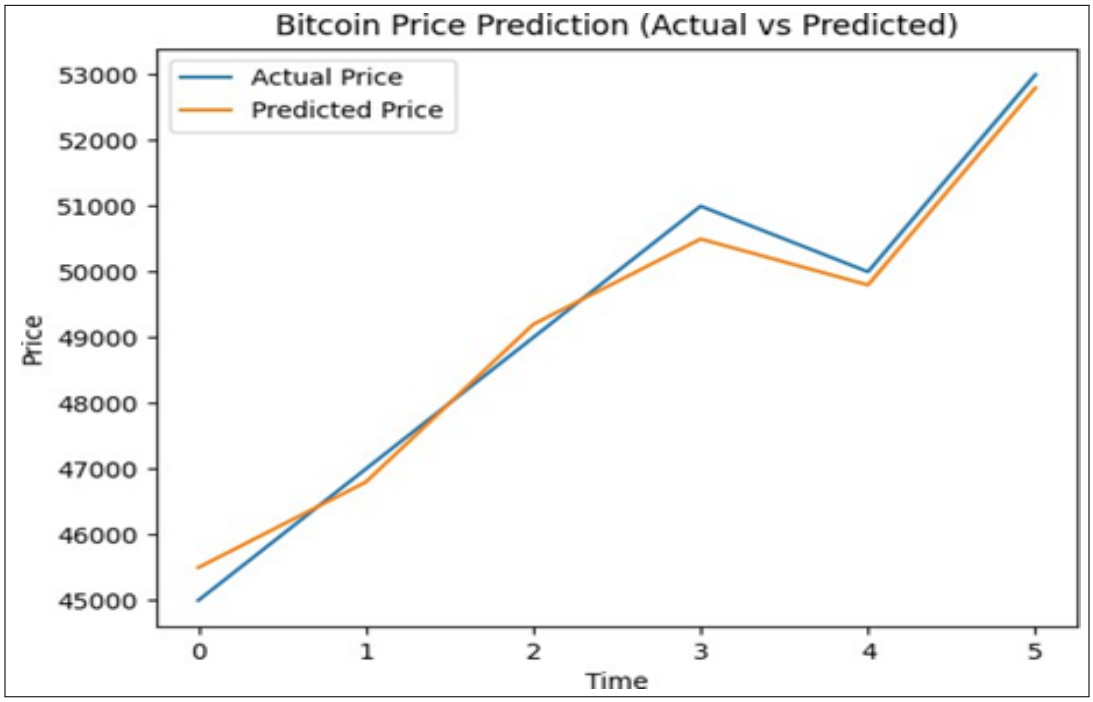

Figure 4. Actual and predicted bitcoin price comparison

The graphical comparison complements the statistical evaluation by showing the prediction behaviour at the individual observation level rather than only through aggregate error metrics. This representation also helps identify regions where the prediction follows the underlying price movement and regions where greater deviation occurs. The actual-predicted analysis therefore provides an additional assessment of the practical behaviour of the implemented prediction models.

4.4 Prediction Fusion and Trend Classification

The proposed system combines the outputs of the individual prediction models to generate a consolidated price estimate. In parallel, a Random Forest classifier is incorporated to categorize

the predicted market direction into UP, DOWN, and NEUTRAL classes. The operational components associated with prediction fusion and trend classification are summarized in Table 2.

Table 2. Prediction fusion and trend classification components

Component	**Input**	**Processing**	**Output**
RF Regression	Market features	Random Forest regression	Predicted Bitcoin price
SVM Regression	Market features	Support Vector Machine regression	Predicted Bitcoin price
XGBoost Regression	Market features	Gradient-boosting regression	Predicted Bitcoin price
Prediction Fusion	Individual model predictions	Combined prediction outputs	Consolidated price estimate
Trend Classification	Market features	Random Forest classification	UP / DOWN / NEUTRAL

The fusion component provides an integrated prediction output from the implemented regression models, while the classification component adds directional information to the continuous price estimate. This combination enables the system to represent both estimated price magnitude and market direction within the prediction workflow.

4.5 Web Application and Real-Time Prediction Analysis

The developed web application provides an interactive interface for accessing the prediction framework and monitoring the operational status of the implemented components. Figure 5 presents the dashboard interface, where the current market trend, active prediction models, data-source status, recent market movement, and prediction history are displayed within a unified view.

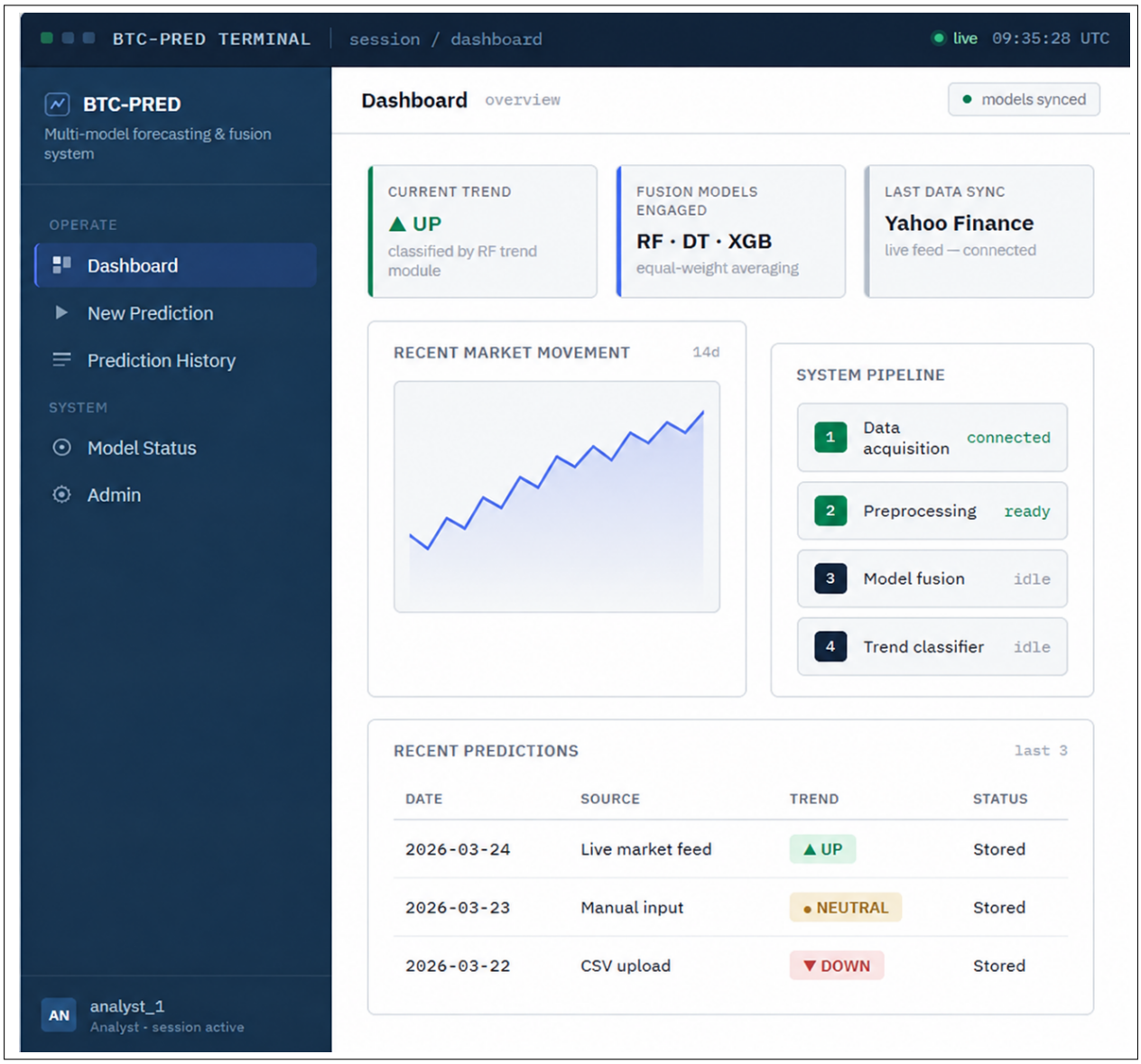

Figure 5. Dashboard interface of the proposed bitcoin price prediction system

The dashboard integrates Random Forest, Decision Tree, and XGBoost through an equal-weight prediction-fusion configuration, while the Random Forest trend module provides the directional classification. The interface also indicates the connection with the Yahoo Finance data source for live market-feed acquisition. The system pipeline displays the operational stages from data acquisition and preprocessing to model fusion and trend classification, providing a direct representation of the prediction workflow.

The prediction-history panel records outputs obtained from different input sources, including live market feed, manual input, and CSV upload, together with their corresponding

directional classifications. The displayed records demonstrate the integration of prediction generation, trend classification, and result storage within the web interface. Thus, the application provides an operational layer through which the developed prediction components can be accessed and monitored in a unified environment.

5. Conclusion and Future Work

The proposed Bitcoin price prediction framework integrates machine-learning regression, prediction fusion, trend classification, and web-based deployment into a unified forecasting system. The comparative evaluation demonstrates that the integrated modelling strategy can capture complex relationships within Bitcoin market variables and generate consistent price predictions. The fusion mechanism further provides a consolidated predictive output, while the directional classification component extends the system beyond numerical forecasting by identifying market movement states. Flask-based deployment enables interactive prediction, visualization, and prediction-history management, supporting practical use of the developed framework. Future work will focus on incorporating additional market indicators, sentiment and blockchain-derived features to improve contextual representation. Advanced deep-learning and temporal models can also be investigated for sequential dependency modelling. Further development can include adaptive online learning, uncertainty estimation, and real-time model updating for changing cryptocurrency market conditions.

Reference

[1] Velankar, Siddhi, Sakshi Valecha, and Shreya Maji. "Bitcoin price prediction using machine learning." In 20th International Conference on Advanced Communication Technology (ICACT), IEEE, 2018, 144-147.

[2] Srivastava, Vibha, Vijay Kumar Dwivedi, and Ashutosh Kumar Singh. "Cryptocurrency price prediction using enhanced PSO with extreme gradient boosting algorithm." Cybernetics and Information Technologies, 2023, vol. 23, no. 2: 170-187.

[3] Aggarwal, Divya, Shabana Chandrasekaran, and Balamurugan Annamalai. "A complete empirical ensemble mode decomposition and support vector machine-based approach to predict Bitcoin prices." Journal of Behavioral and Experimental Finance, 2020, vol. 27: 100335.

[4] Ganesan, Shanmuga Canaraj, Saripelli Chaitanya Bangarraju, Kishan Tiwari, Katakam Jaswanthi, and Amiya Bhaumik. "Bitcoin Price Prediction Using XGBoost Classifier: A Machine Learning Approach." Soft Computing and Signal Processing: Proceedings of 7th ICSCSP, 2024, vol. 1, no. 1: 91.

[5] Nielsen, Michael A. "Neural networks and deep learning." (2015).

[6] Wang, Sidan, Zidong Huang, and Zhong-Wei Liao. "Multi-scale decomposition for deep learning-based Bitcoin price forecasting." Finance Research Letters, 2026, vol. 96: 109776.

[7] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, 785-794.

[8] Box, George EP, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[9] Brownlee, Jason. Machine learning mastery. Machine Learning Mastery, 2022.

[10] Géron, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.", 2022.

[11] Yahoo Finance API, Dataset, Accessed 2025. Available: https://finance.yahoo.com/