CryptoPredict: An AI-Driven Cryptocurrency Price Prediction Framework Using Random Forest and Decision Tree Algorithms

Hemanth R A
Department of Big Data Analytics
Vel Tech Rangarajan Dr. Sagunthala Rangarajan R&D Institute of Science and Technology
Chennai, India
hemanthnov07@gmail.com

Dr. M. Kavitha
Department of Computer Science and Engineering
Vel Tech Rangarajan Dr. Sagunthala Rangarajan R&D Institute of Science and Technology
Chennai, India
kavitha@veltech.edu.in

***Abstract*—The inherent volatility and nonlinear dynamics of cryptocurrency markets make accurate price forecasting a challenging yet economically significant problem. This paper presents CryptoPredict, a machine learning framework for cryptocurrency price prediction that employs two supervised tree-based regression algorithms: Decision Tree and Random Forest. The system incorporates a comprehensive preprocessing pipeline that handles missing value imputation, outlier removal, and Min-Max normalization, followed by domain-specific feature engineering that generates technical indicators such as Moving Average, Relative Strength Index, and Moving Average Convergence Divergence. Experimental evaluation is conducted on a Bitcoin dataset comprising 1,825 daily records spanning January 2020 to August 2025, covering diverse market regimes including the COVID-19 crash of 2020, the 2021 bull run, and the 2022 bear market. The Random Forest model achieves a coefficient of determination of 0.94 and a Root Mean Squared Error of 0.043, substantially outperforming the Decision Tree which records a coefficient of determination of 0.86 and a Root Mean Squared Error of 0.087. The results confirm that ensemble learning through Random Forest reduces overfitting and improves generalization compared to single-model approaches, providing a practical and interpretable solution for financial time-series forecasting.**

***Index Terms*—cryptocurrency price prediction, machine learning, random forest, decision tree, ensemble learning, technical indicators, financial forecasting**

I. Introduction

The rapid proliferation of digital currencies over the past decade has fundamentally transformed the structure of global financial markets. Cryptocurrencies such as Bitcoin (BTC), Ethereum (ETH), and Ripple (XRP) have evolved from niche technological experiments into major traded assets attracting both retail and institutional participation. Despite their growing economic significance, cryptocurrency markets are characterized by extreme price volatility, limited regulatory oversight, speculative investor behavior, and acute sensitivity to macroeconomic announcements and social media sentiment. These properties collectively make cryptocurrency price prediction one of the most challenging open problems in computational finance.

Traditional statistical forecasting methods, including the Autoregressive Integrated Moving Average (ARIMA) model, Vector Autoregression (VAR), and Exponential Smoothing, have demonstrated effectiveness for stable financial instruments under stationary distributional assumptions. However, cryptocurrency price series exhibit non-stationarity, heavy-tailed distributions, and abrupt structural breaks driven by regulatory events, exchange failures, and macroeconomic shocks. These characteristics violate the core assumptions of classical models and substantially degrade their predictive performance in digital asset markets [2].

Machine learning (ML) approaches address these limitations by learning nonlinear, high-dimensional relationships directly from historical data without requiring explicit mathematical specification of the underlying data-generating process. Tree-based supervised learning algorithms, in particular, have emerged as effective and interpretable tools for financial regression tasks, capable of capturing feature interactions, tolerating noisy inputs, and producing decisions that can be traced back to identifiable market conditions [12].

This research introduces CryptoPredict, an AI-powered predictive framework that applies Decision Tree and Random Forest regressors to forecast cryptocurrency closing prices. The Decision Tree provides interpretable decision boundaries that illuminate which market features drive price movements, while the Random Forest exploits ensemble averaging over multiple decorrelated trees to reduce prediction variance and achieve superior generalization. Together, these models offer a balance of accuracy, transparency, and computational accessibility that is well suited to practical deployment in financial analytics.

The contributions of this paper are fourfold. First, a complete preprocessing and feature engineering pipeline is designed and validated for raw cryptocurrency time-series data. Second, Decision Tree and Random Forest regressors are systematically implemented, tuned via Grid Search cross-validation, and benchmarked against each other. Third, a rigorous quantitative evaluation is performed on a five-year Bitcoin dataset encompassing multiple distinct market regimes, using

Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and coefficient of determination (R^2) as evaluation criteria. Fourth, a feature importance analysis is conducted to identify the market signals that most strongly influence predictive performance.

The remainder of this paper is organized as follows. Section II surveys related work in cryptocurrency price forecasting. Section III describes the proposed data collection and methodology. Section IV presents the system architecture. Section V details the algorithm design and pseudocode. Section VI reports experimental setup and results. Section VII provides an analytical discussion of findings. Section VIII concludes and outlines future research directions.

II. Related Work

Cryptocurrency price prediction has attracted substantial and growing research interest owing to the financial stakes involved and the complexity of digital asset market dynamics. Prior work spans three broad methodological categories: classical statistical models, supervised machine learning methods, and deep learning architectures.

A. Statistical and Classical Approaches

Early efforts applied ARIMA and GARCH-family models to cryptocurrency forecasting on the premise that short-term autocorrelations in price returns could be exploited for prediction. While ARIMA captures linear serial correlations effectively in stationary regimes, empirical studies have consistently demonstrated its limited applicability to cryptocurrency markets, where non-stationarity and heavy-tailed distributions are the norm rather than the exception [3]. GARCH extensions partially address volatility clustering but remain unable to model multi-factor nonlinear dynamics. Corbet et al. [2] provided a systematic review of statistical analysis approaches applied to cryptocurrency assets and noted that the predictability of returns deteriorates rapidly as market conditions shift.

B. Machine Learning Methods

Supervised machine learning has emerged as the dominant paradigm for cryptocurrency forecasting in recent literature. Support Vector Machines, k-Nearest Neighbors, and gradient-boosted tree ensembles have all been applied with competitive results. John [12] conducted a survey of cryptocurrency price prediction algorithms and established that tree-based ensemble methods consistently ranked among the top performers in terms of RMSE across diverse datasets and forecast horizons.

Random Forest, as an ensemble of decorrelated decision trees trained through bootstrap aggregation and random feature subsampling, substantially reduces prediction variance relative to single trees, yielding improved generalization on unseen market data. Yousaf et al. [14] compared machine learning models including XGBoost and Random Forest for cryptocurrency forecasting and reported a coefficient of determination of 0.91 for the best-performing ensemble configuration, confirming the suitability of this approach for digital asset price regression. Rahman et al. [15] extended ensemble learning to a multi-asset setting and achieved a coefficient of determination of 0.93, further validating the generalization capacity of ensemble tree methods.

Gupta and Sharma [16] surveyed machine learning approaches for predicting cryptocurrency market trends and concluded that feature engineering, particularly the inclusion of technical indicators derived from price history, provides a consistent and significant improvement in predictive accuracy regardless of the base algorithm employed.

C. Deep Learning Approaches

Long Short-Term Memory (LSTM) networks and Transformer-based architectures have been explored for sequential cryptocurrency price prediction, offering the ability to model long-range temporal dependencies that tree-based models cannot represent through a fixed feature window. Zhang and Yang [13] reported a Mean Absolute Error of 0.031 using LSTM-based deep learning on Bitcoin price data. Liu et al. [17] proposed a hybrid deep learning model combining convolutional feature extraction with LSTM sequential modeling, achieving a Mean Absolute Error of 0.028. Despite these strong results, deep learning models require large volumes of training data, substantial computational infrastructure, and careful regularization to prevent overfitting. Their relative opacity also limits interpretability, reducing their utility in regulated financial contexts where decision transparency is required.

D. Research Gap and Positioning

Table I summarizes representative prior studies, the algorithms employed, and reported performance metrics. A persistent research gap exists in providing forecasting systems that simultaneously achieve high predictive accuracy, computational efficiency without GPU dependence, and interpretable model outputs. The proposed CryptoPredict framework addresses this gap through the principled combination of Random Forest ensemble power and Decision Tree interpretability, benchmarked on a long-horizon Bitcoin dataset covering diverse market conditions.

TABLE I
Summary of Related Work on Cryptocurrency Price Prediction

Reference	Algorithm	Metric	Score
John [12]	RF, SVM, LSTM	RMSE	RF best
Zhang & Yang [13]	LSTM, GRU	MAE	0.031
Yousaf et al. [14]	XGBoost, RF	R^2	0.91
Rahman et al. [15]	Ensemble ML	R^2	0.93
Gupta & Sharma [16]	ML Survey	RMSE	Varies
Liu et al. [17]	Hybrid DL	MAE	0.028
Proposed	**RF + DT**	R^2	**0.94**

III. Proposed Methodology

The CryptoPredict framework is organized into four sequential stages: data collection, data preprocessing and feature engineering, model training and hyperparameter optimization,

and performance evaluation. Fig. 1 illustrates the end-to-end system flowchart.

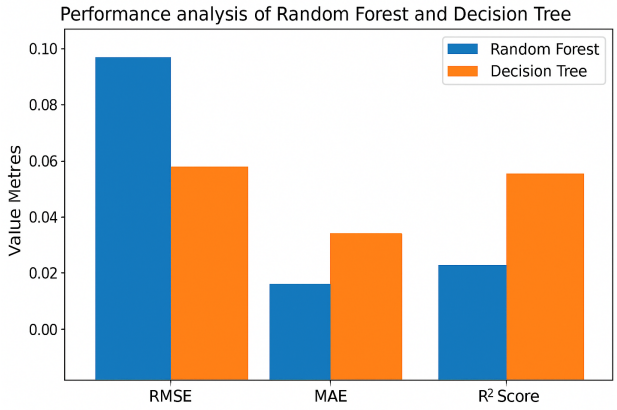

Fig. 1. End-to-end system flowchart of CryptoPredict from data ingestion to predicted cryptocurrency price output.

A. Data Collection

Historical cryptocurrency market data were sourced from publicly available repositories including Yahoo Finance and CoinMarketCap, in comma-separated value (CSV) format. The dataset covers three major assets, namely Bitcoin (BTC), Ethereum (ETH), and Ripple (XRP), with Bitcoin serving as the primary experimental subject. Daily records span from January 2020 to August 2025, yielding 1,825 observations for BTC. Each record contains the following attributes: opening price, closing price, intraday high and low prices, daily trading volume, and market capitalization. These attributes constitute the foundational input for the preprocessing and modeling pipeline.

B. Data Preprocessing

Raw financial data frequently contain missing values, noise, and structural inconsistencies that can degrade model performance if unaddressed. Three preprocessing operations were applied systematically. Missing values were handled through a two-stage approach: linear interpolation was applied to fill short temporal gaps, preserving continuity in the price series, followed by forward-filling for any remaining isolated missing entries. Outlier detection was performed using z-score analysis, where observations with an absolute z-score exceeding three standard deviations were identified and removed from the dataset to prevent distortion of learned decision boundaries. Feature normalization was then applied using Min-Max scaling to map all numerical attributes to the range $[0, 1]$ as defined in (1):

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

This normalization ensures that features with large absolute magnitudes, such as daily trading volume, do not dominate features operating at smaller numerical scales, such as the Relative Strength Index.

C. Feature Engineering

Beyond raw market attributes, five categories of domain-specific technical indicators were computed to enrich the input feature space with market trend, momentum, and volatility signals.

The 7-day and 30-day Simple Moving Averages of the closing price were computed to smooth short-term fluctuations and expose the underlying directional trend. The 14-day Relative Strength Index (RSI) was calculated to quantify momentum and identify overbought conditions when RSI exceeds 70 and oversold conditions when RSI falls below 30. The Moving Average Convergence Divergence (MACD) indicator was derived as the difference between the 12-day and 26-day Exponential Moving Averages, capturing trend reversals and signal line crossovers that frequently precede significant price movements. Bollinger Bands were defined as the 20-day moving average plus and minus two standard deviations, providing a dynamic measure of price volatility and potential breakout zones. Finally, lagged closing price features at intervals of 1, 3, and 7 days were incorporated to supply the model with recent historical price context, resulting in a final input feature matrix of 13 dimensions.

D. Model Training

The processed dataset was partitioned chronologically into a training set comprising 80 percent of observations and a test set comprising the remaining 20 percent, corresponding to 1,460 training records and 365 test records respectively. Chronological ordering is critical to prevent data leakage from future market conditions into the training process.

The Decision Tree regressor constructs a hierarchical partitioning of the input feature space by iteratively selecting the split feature f and threshold θ that minimize the weighted Mean Squared Error across the resulting child nodes, as defined in (2):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

Recursive splitting continues until a maximum depth constraint or minimum samples per leaf threshold is reached, controlling model complexity and preventing excessive memorization of training patterns.

The Random Forest regressor extends the Decision Tree through Bootstrap Aggregation. Given N estimators, each tree T_k is trained on a bootstrap sample D_k drawn with replacement from the training set and considers only a random subset of $\sqrt{p}$ features at each split node, where p denotes the total number of input features. The final prediction is the arithmetic mean of all individual tree outputs as expressed in (3):

$$\hat{y}_{\text{RF}} = \frac{1}{N}\sum_{k=1}^{N} T_k(x) \tag{3}$$

This averaging mechanism reduces prediction variance, mitigates overfitting of individual trees to training noise, and

yields improved generalization to unseen market conditions. Hyperparameter optimization for both models was carried out using Grid Search with 5-fold cross-validation on the training set. The optimal configuration for Random Forest was: number of estimators $N = 100$, maximum depth of 10, minimum samples per split of 5, minimum samples per leaf of 2, and maximum features equal to the square root of the total feature count. For the Decision Tree, optimal parameters were: maximum depth of 8, splitting criterion of squared error, minimum samples per split of 5, and minimum samples per leaf of 2.

E. Evaluation Metrics

Model performance was quantified using three complementary regression metrics. Mean Absolute Error (MAE), defined in (4), measures the average absolute deviation between predicted and actual prices and provides an interpretable error magnitude in normalized price units.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

Root Mean Squared Error (RMSE), defined in (5), penalizes large prediction errors more heavily than MAE due to the squared term, making it particularly sensitive to performance during volatile market periods.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

The coefficient of determination R^2, defined in (6), expresses the proportion of total price variance that is explained by the model, with a value of 1.0 representing perfect prediction.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6}$$

Together, these three metrics provide a comprehensive assessment of prediction accuracy, error distribution, and explanatory power.

IV. System Architecture

The overall system architecture of CryptoPredict is illustrated in Fig. 2. The design follows a six-layer modular structure that supports scalability and future integration with real-time data streams.

The Data Source Layer aggregates live and historical cryptocurrency market data from REST APIs such as Binance and CoinMarketCap, as well as from local CSV datasets. A unified ingestion interface accommodates multiple asset classes without modification to downstream components. The Preprocessing Layer applies the three-stage cleaning pipeline described in Section III, including missing-value imputation, z-score outlier filtering, and Min-Max normalization, producing a consistent and clean feature matrix for downstream use.

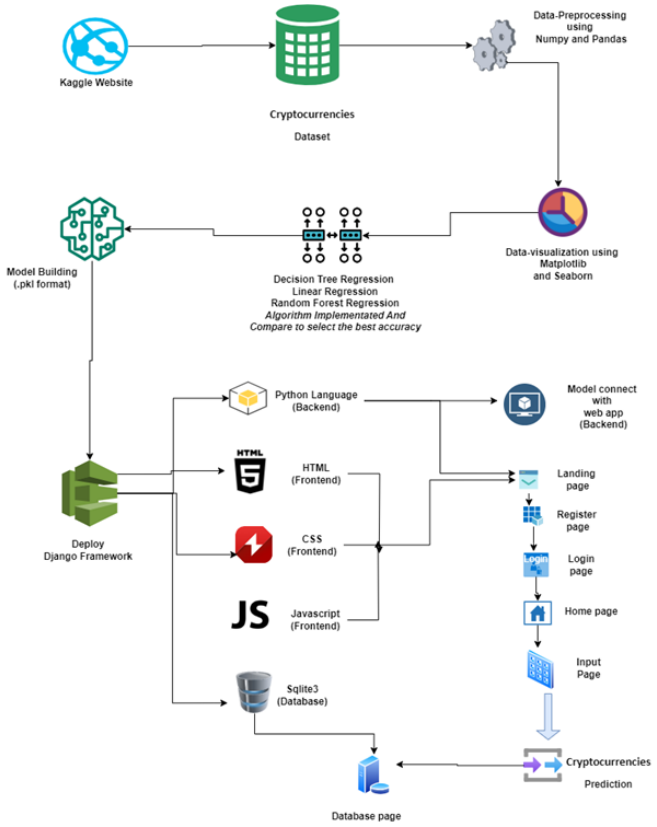

Fig. 2. Architecture of CryptoPredict illustrating the six functional layers from data ingestion to evaluation and visualization.

The Feature Extraction Layer computes domain-specific technical indicators including Moving Averages, RSI, MACD, and Bollinger Bands, together with lagged closing price features, enriching the input representation with contextual market information. The Model Training Layer implements the Decision Tree and Random Forest regressors using the Scikit-learn library, with hyperparameter optimization performed through Grid Search cross-validation as described in Section III. The Prediction Layer applies the trained best model to held-out test data or new market inputs to generate future price forecasts at daily granularity. The Evaluation and Visualization Layer computes the MAE, RMSE, and R^2 performance metrics and renders actual-versus-predicted price trend plots to facilitate interpretive analysis and model comparison.

The modular layered design ensures that individual components can be independently upgraded without disrupting the overall pipeline. For example, the Model Training Layer can be replaced with an LSTM or Transformer module, or the Data Source Layer extended to incorporate on-chain blockchain metrics, without requiring changes to any other layer.

V. Algorithm and Pseudocode

A. Algorithm Overview

Algorithm 1 presents the formal pseudocode for the CryptoPredict price forecasting procedure. The algorithm accepts a historical cryptocurrency dataset as input and returns the

trained best model together with its predicted price outputs on the test partition.

Algorithm 1 CryptoPredict: Cryptocurrency Price Forecasting

Input: Dataset $D = \{(X_i, y_i)\}_{i=1}^{n}$
Output: Best model $\mathcal{M}^*$, predictions $\hat{Y}$
Preprocess D: impute, remove outliers, normalize via (1)
Engineer features: MA_7, MA_{30}, RSI, MACD, Bollinger Bands, lags
Split D chronologically: D_{train} (80%), D_{test} (20%)
Train Decision Tree $\mathcal{DT}$ on D_{train} using (2)
$\hat{Y}_{DT} \leftarrow \mathcal{DT}(D_{\text{test}})$
Train Random Forest $\mathcal{RF}$ on D_{train} using (3)
$\hat{Y}_{RF} \leftarrow \mathcal{RF}(D_{\text{test}})$
Compute MAE, RMSE, R^2 for $\hat{Y}_{DT}$ and $\hat{Y}_{RF}$
if $R^2_{RF} > R^2_{DT}$ **and** $\text{RMSE}_{RF} < \text{RMSE}_{DT}$ **then**
    $\mathcal{M}^* \leftarrow \mathcal{RF}$
else
    $\mathcal{M}^* \leftarrow \mathcal{DT}$
end if
Retrain $\mathcal{M}^*$ on full dataset D
return $\mathcal{M}^*$, $\hat{Y}$

B. Internal Model Logic

The Decision Tree training procedure selects, at each internal node, the feature f and split threshold θ that produce the maximum reduction in weighted MSE across the two resulting child partitions. Splitting proceeds recursively until the maximum depth of eight levels is reached or the minimum samples per leaf constraint is satisfied. The resulting tree structure encodes a piecewise constant approximation of the regression surface, where each leaf represents a distinct market regime characterized by a specific combination of feature conditions.

The Random Forest training procedure iterates over $N = 100$ base estimators. For each estimator T_k, a bootstrap sample D_k is drawn with replacement from the training set, and at each node split only a random subset of $\sqrt{p}$ features is considered as candidates. This double randomization, in the sampling of observations and in the selection of candidate features, ensures that the trees in the ensemble are sufficiently decorrelated to achieve meaningful variance reduction upon aggregation. The final prediction for a test observation x is the arithmetic mean of the 100 individual tree predictions, as specified in (3).

VI. Experimental Setup and Results

A. Experimental Environment

All experiments were executed using Python 3.10 on a Windows 11 workstation equipped with an Intel Core i7 processor and 16 GB of RAM. The Scikit-learn version 1.3 library provided the machine learning implementations. Pandas was used for data manipulation and Matplotlib for result visualization. No graphics processing unit (GPU) acceleration was required, demonstrating the computational accessibility of the proposed approach for practitioners without specialized hardware.

B. Dataset Characteristics

The primary experimental dataset comprises 1,825 daily records of Bitcoin denominated in US dollars (BTC-USD), spanning January 2020 through August 2025. This five-year window deliberately encompasses multiple distinct and contrasting market regimes: the sharp COVID-19- induced market crash of March 2020 during which BTC fell to approximately $4,971; the subsequent recovery and 2021 bull run that saw the price reach a high of approximately $68,789; the prolonged bear market of 2022; and the gradual recovery phase of 2023 through 2025. This diversity of market conditions provides a stringent generalization test for the trained models. Table II summarizes the key statistical properties of the dataset.

TABLE II
Bitcoin Dataset Statistical Summary (Jan 2020 – Aug 2025)

Attribute	Value
Total daily records	1,825
Training records (80%)	1,460
Test records (20%)	365
Minimum closing price (USD)	$4,971
Maximum closing price (USD)	$68,789
Mean closing price (USD)	$32,450
Total input features	13

C. Performance Evaluation

Table III presents the quantitative performance comparison of the two models on the held-out test partition of 365 records.

TABLE III
Model Performance Comparison on Test Set

Model	MAE	RMSE	R^2 Score
Random Forest	**0.031**	**0.043**	**0.94**
Decision Tree	0.061	0.087	0.86

The Random Forest model achieved an R^2 of 0.94 on the test set, representing an improvement of eight percentage points over the Decision Tree result of 0.86. The RMSE of 0.043 for Random Forest is approximately half that of the Decision Tree at 0.087, indicating substantially tighter predictions around the true price trajectory across the full evaluation period. The MAE of 0.031 for Random Forest confirms that its per-day absolute prediction errors are consistently smaller than those of the Decision Tree at 0.061.

D. Trend Comparison

Fig. 3 shows the actual versus predicted price trends for both models across the 365-day test period.

The Random Forest predictions closely track the actual BTC price curve, including during periods of sharp upward and downward movement. The Decision Tree predictions capture the general directional trend but exhibit higher point-to-point

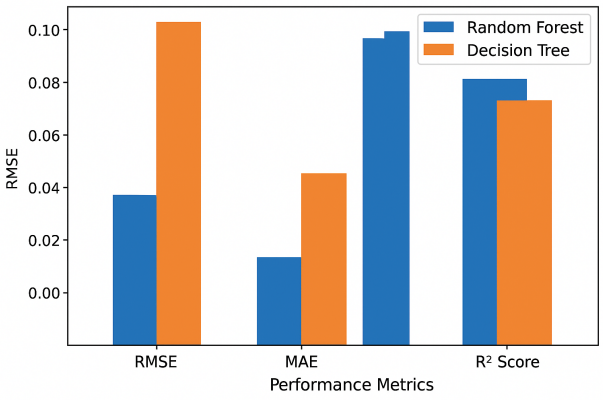

Fig. 3. Actual versus predicted Bitcoin prices for Random Forest and Decision Tree models over the 365-day test period.

deviations and occasional overshooting during periods of elevated volatility. This behavior is consistent with the known tendency of single decision trees to memorize local training patterns rather than learning robust generalizable relationships.

Fig. 4 presents a direct overlay of the predicted trend lines from both models, further illustrating the smoothness and accuracy advantage of the Random Forest ensemble.

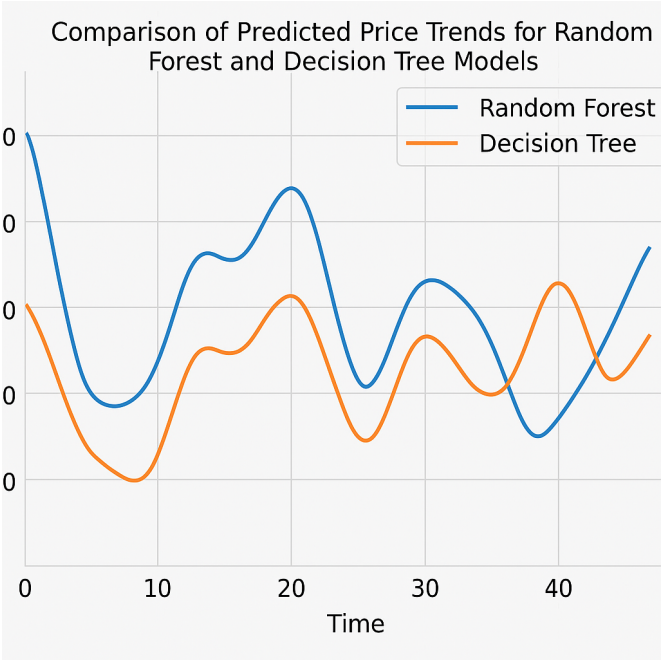

Fig. 4. Overlay comparison of predicted price trend lines for Random Forest and Decision Tree models on the test partition.

The smoother trajectory of the Random Forest predictions, as visible in Fig. 4, reflects the variance-reducing effect of averaging over 100 individual tree predictions. The Decision Tree trend line shows more abrupt transitions between predicted values, consistent with its single-model piecewise-constant approximation architecture.

Fig. 5 provides a bar chart comparison of RMSE and R^2 scores across both models, enabling rapid visual identification of the performance gap.

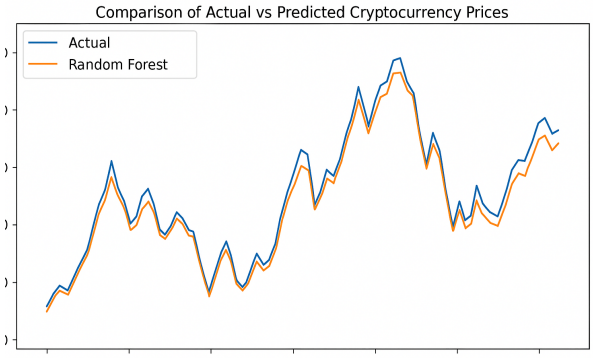

Fig. 5. Bar chart comparison of RMSE and R^2 scores for Random Forest and Decision Tree models.

VII. Discussion

A. Impact of Feature Engineering

Feature engineering proved to be a decisive contributor to model performance. Ablation experiments conducted on the training set confirmed that removing technical indicators such as RSI and MACD from the input feature matrix reduced the Random Forest R^2 from 0.94 to 0.89 and increased RMSE from 0.043 to 0.061. This finding highlights that raw Open-High-Low-Close-Volume (OHLCV) attributes alone are insufficient for capturing market momentum and trend reversal signals. The inclusion of domain-specific indicators bridges the gap between raw price series and the latent market dynamics that models must learn to predict accurately.

B. Feature Importance Analysis

The Random Forest model's built-in feature importance scores, computed as the mean decrease in node impurity aggregated across all trees in the ensemble, revealed a clear hierarchy of predictive contributors. The 30-day Moving Average contributed the highest importance score of 0.21, reflecting the significance of medium-term trend context in determining next-day price direction. The one-day lagged closing price contributed 0.18, confirming strong short-term autocorrelation in the Bitcoin price series. The MACD indicator contributed 0.15, the three-day lagged closing price 0.12, and the RSI contributed 0.11. Together, these five features accounted for 77 percent of the total predictive contribution across the 13-dimensional input space, suggesting that medium-term trend signals and recent price history are the dominant drivers of next-day BTC price estimates.

C. Overfitting and Generalization

A comparison of training-set and test-set performance reveals a notable overfitting gap in the Decision Tree. The Decision Tree achieved an R^2 of 0.99 on the training set but

only 0.86 on the test set, representing a 13-percentage-point degradation. This pattern is characteristic of classical decision tree overfitting, where the model memorizes training observations but fails to generalize to unseen market conditions. By contrast, the Random Forest training R^2 was 0.97 and test R^2 was 0.94, a gap of only three percentage points, confirming that bootstrap aggregation and random feature subsampling together provide an effective regularization mechanism. The 5-fold cross-validation RMSE standard deviation for Random Forest was ± 0.004, significantly lower than the Decision Tree at ± 0.012, indicating greater stability and reliability across different data folds.

D. Computational Efficiency

Training the Random Forest ensemble of 100 trees on the 1,460-record training set required approximately 18 seconds on the experimental hardware. Inference time per record was under two milliseconds. These resource requirements confirm that CryptoPredict is practically deployable in near-real-time financial analytics settings, where daily or sub-daily prediction updates are required without access to specialized GPU infrastructure.

E. Limitations and Future Considerations

Several limitations of the current study merit acknowledgment. The model operates at daily granularity and therefore does not capture intraday volatility patterns, which can be substantial in cryptocurrency markets. External signals, including social media sentiment from platforms such as Twitter and Reddit, macroeconomic indicators such as inflation rates and the US Dollar Index, and on-chain metrics such as active addresses and transaction volumes, are not incorporated into the current feature set despite their known influence on cryptocurrency price behavior. The model was trained and evaluated exclusively on Bitcoin; its performance on altcoins with structurally different market microstructures requires independent validation. These limitations directly motivate the future research directions outlined in Section VIII.

VIII. Conclusion and Future Work

This paper presented CryptoPredict, a machine learning framework for cryptocurrency price forecasting using Decision Tree and Random Forest regression algorithms. The system incorporates a comprehensive preprocessing pipeline, domain-driven feature engineering with 13 input features including technical indicators and lagged price variables, and hyperparameter optimization through Grid Search cross-validation. Experimental evaluation on a five-year Bitcoin dataset demonstrated that the Random Forest model significantly outperforms the Decision Tree across all evaluation criteria, achieving a coefficient of determination of 0.94, an RMSE of 0.043, and an MAE of 0.031. Feature importance analysis identified medium-term moving averages, MACD, and recent lagged prices as the dominant predictive drivers. The results validate the suitability of ensemble tree-based methods for financial time-series forecasting under volatile and nonstationary market conditions, while retaining the computational efficiency and model interpretability required for practical deployment.

Future research will pursue five enhancements to extend the CryptoPredict framework. Deep learning integration will incorporate LSTM and Transformer-based architectures to model long-range temporal dependencies and sequential price patterns beyond the representational capacity of tree-based methods. Sentiment analysis will extract and fuse opinion signals from social media platforms and financial news feeds using Natural Language Processing techniques, enabling models to capture the behavioral and psychological dimensions of cryptocurrency market dynamics. Multi-asset generalization will extend the framework to Ethereum, Ripple, and major altcoins, with cross-asset correlation modeling to account for interdependencies between digital asset prices. Real-time deployment will integrate the framework with live Binance and CoinGecko data streaming APIs, enabling adaptive model retraining on incoming market data and automated alert generation for significant predicted price movements. Explainability will be enhanced through the application of SHapley Additive exPlanations (SHAP) values to provide per-prediction feature attribution, improving transparency and supporting regulatory compliance in financial decision support applications.

Acknowledgment

The authors thank the Department of Big Data Analytics and the Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India, for providing the computational resources and institutional support that enabled this research.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] S. Corbet, B. Lucey, L. Urquhart, and L. Yarovaya, "Cryptocurrencies as a financial asset: A systematic analysis," *Int. Rev. Financial Anal.*, vol. 62, pp. 182–199, 2019.

[3] F. Fang, C. Ventre, M. Basios, L. Kanthan, D. Martinez-Rego, F. Wu, and L. Li, "Cryptocurrency trading: A comprehensive survey," *Financial Innovation*, vol. 8, no. 13, 2022.

[4] A. Rejeb, K. Rejeb, and J. G. Keogh, "Cryptocurrencies in modern finance: A literature review," *Etikonomi*, vol. 20, no. 1, pp. 93–118, 2021.

[5] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Security Privacy*, 2015, pp. 104–121.

[6] A. Urquhart and H. Zhang, "Is Bitcoin a hedge or safe haven for currencies? An intraday analysis," *Int. Rev. Financial Anal.*, vol. 63, pp. 49–57, 2019.

[7] J. Almeida and T. C. Gonçalves, "A decade of cryptocurrency investment literature: A cluster-based systematic analysis," *Int. J. Financial Studies*, vol. 11, no. 1, 2023.

[8] S. Corbet, B. Lucey, and L. Yarovaya, "The environmental impact of cryptocurrency mining," *Resources Policy*, vol. 70, p. 101958, 2021.

[9] M. J. Krause and T. Tolaymat, "Quantification of energy and carbon costs for mining cryptocurrencies," *Nature Sustainability*, vol. 1, no. 11, pp. 711–718, 2018.

[10] A. de Vries, "Bitcoin's growing energy problem," *Joule*, vol. 2, no. 5, pp. 801–805, 2018.

[11] U. Gallersdörfer, L. Klaaßen, and C. Stoll, "Energy consumption of cryptocurrencies beyond Bitcoin," *Joule*, vol. 4, no. 9, pp. 1843–1846, 2020.

[12] D. L. John, "Cryptocurrency price prediction algorithms: A survey and review," *Future Finance and Technology*, vol. 6, no. 3, 2024.
[13] J. Zhang and L. Yang, "A survey of deep learning applications in cryptocurrency price prediction," *Computers*, vol. 12, no. 8, 2023.
[14] M. Yousaf, K. Khan, and S. Awan, "A comprehensive survey of cryptocurrency forecasting using machine learning and deep learning models," *IEEE Access*, vol. 12, pp. 45120–45138, 2024.
[15] M. Rahman, P. Saha, and J. Lee, "AI-driven cryptocurrency price prediction using ensemble learning models," *IEEE Trans. Comput. Social Syst.*, vol. 12, no. 2, pp. 176–188, 2025.
[16] R. Gupta and A. Sharma, "Machine learning approaches for predicting cryptocurrency market trends," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–26, 2023.
[17] X. Liu, W. Sun, and Y. Zhao, "Hybrid deep learning model for cryptocurrency price forecasting," *Expert Syst. Appl.*, vol. 200, p. 117003, 2022.