Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

On Integrating Technical Analysis with Machine Learning for Cryptocurrency Price Forecasting

Steve Karam

College of Engineering and Technology, American University of the Middle East, Egaila, 54200, Kuwait
E-mail: steve.karam@aum.edu.kw

Received: 15 November 2024; **Revised:** 14 January 2025; **Accepted:** 15 January 2025

Abstract: This study explores the integration of technical indicators, specifically Exponential Moving Averages (EMA) and Volume Weighted Average Price (VWAP), into machine learning models for cryptocurrency price forecasting. Our findings reveal that including these indicators can complicate the modeling process without necessarily improving performance. Support Vector Regression (SVR) and Random Forest Regressor (RFR) models outperform deep learning approaches such as Long Short-Term Memory (LSTM) networks, demonstrating higher predictive accuracy with simpler feature sets. These findings emphasize the challenges of high-dimensional data and the critical role of rigorous feature selection and preprocessing in financial forecasting. Practical implications and trade-offs between model complexity and prediction accuracy are discussed, providing valuable insights for researchers and practitioners in financial analytics.

***Keywords*:** machine learning, neural networks, price prediction, financial time series deep learning, market forecasting, technical analysis

MSC: 91G70, 68T01, 62M45, 91B84

1. Introduction

Accurately forecasting cryptocurrency prices remains one of the most significant challenges in financial decision-making, particularly given the inherent volatility and complexity of these markets. Bitcoin, as the leading cryptocurrency, offers an ideal testing ground for predictive models due to its substantial price fluctuations and rich historical data. Despite considerable advancements in both machine learning techniques and technical analysis, achieving reliable prediction accuracy continues to pose significant challenges for researchers and practitioners alike.

This study investigates a critical question in cryptocurrency price forecasting: does the integration of traditional technical indicators enhance the predictive power of machine learning models? Specifically, we examine the impact of incorporating Exponential Moving Averages (EMA) and Volume Weighted Average Price (VWAP) into various machine learning frameworks. Our research reveals an unexpected finding: the inclusion of these widely-used technical indicators often complicates the modeling process without necessarily improving predictive performance. In fact, our results demonstrate that Support Vector Regression (SVR) and Random Forest Regressor (RFR) models consistently outperform more complex deep learning approaches such as Long Short-Term Memory (LSTM) networks, particularly when utilizing simpler feature sets.

DOI: https://doi.org/10.37256/cm.6120256080

The contributions of this study are threefold. First, we provide empirical evidence demonstrating how technical indicators affect machine learning model performance in cryptocurrency markets, offering insights into the relationship between feature complexity and prediction accuracy. Second, we present a comprehensive analysis of the trade-offs between feature inclusion, data sparsity, and model generalization, providing practical guidance for practitioners in the field. Third, we develop and validate a framework for evaluating the effectiveness of different machine learning approaches in cryptocurrency price prediction, considering both traditional and deep learning methods.

This work extends the current literature on machine learning applications in finance, building upon recent developments in both technical analysis and algorithmic trading. While previous studies such as Kelly and Xiu [1] and Coqueret and Guida [2] have explored machine learning applications in financial forecasting broadly, our research specifically examines the intersection of technical indicators and machine learning in the context of cryptocurrency markets.

The remainder of this paper is structured as follows: Section 2 presents a comprehensive review of related literature, examining both traditional technical analysis approaches and recent machine learning applications in financial forecasting. Section 3 details our methodology, including data preparation, feature selection, and the implementation of various machine learning models. Section 4 presents our empirical results. Section 5 discusses our conclusions and directions for further research. Finally, Section 6 provides information regarding data availability.

2. Literature review

Machine learning applications in financial forecasting have evolved significantly, driven by the need for advanced tools to navigate volatile and complex markets. Recent reviews by Coqueret and Guida [2] and Kelly and Xiu [1] provide comprehensive overviews of the field, highlighting the transformative role of machine learning in tasks such as stock return prediction and portfolio optimization. While these studies underscore the potential of machine learning, they also emphasize the critical importance of feature selection and preprocessing in achieving robust model performance.

Deep learning models, particularly Long Short-Term Memory (LSTM) networks, have been shown to achieve notable success in stock market prediction tasks by capturing temporal dependencies in price data Adil Moghar and Hamiche [3]. These models leverage their ability to process sequential data, making them suitable for financial forecasting, especially in volatile markets. Fischer and Krauss [4] also demonstrated the potential of LSTM networks for directional predictions, capturing temporal dependencies in cryptocurrency prices. However, they highlighted challenges such as overfitting and high data requirements, especially in sparse datasets.

Specific advancements in return predictability have been demonstrated by Chen et al. [5], who validated deep learning models for asset pricing, and Zhou et al. [6], who explored neural networks' capacity to model nonlinear relationships in financial data. Despite their successes, these studies reveal challenges related to overfitting and data sparsity, which are common hurdles in high-dimensional datasets.

The integration of classification approaches has also gained attention. Labiad et al. [7] applied machine learning techniques to classify short-term stock price movements in emerging markets. Their work highlights the importance of feature selection and preprocessing in minimizing noise and improving classification accuracy in these volatile environments. Similarly, ensemble methods like Random Forests have been extensively validated in financial contexts. Illa et al. [8] demonstrated the applicability of Random Forests and Support Vector Machines in predicting stock prices, emphasizing their ability to model non-linear relationships while reducing overfitting risks.

The integration of technical indicators with machine learning models has also garnered attention. Fieberg et al. [9] demonstrated that trend-based indicators enhance cryptocurrency return models, while Raza and Akhtar [10] explored their effectiveness in emerging stock markets. These studies highlight the delicate balance between enriching models with technical indicators and avoiding unnecessary complexity. Volume-weighted indicators, such as the Volume Weighted Average Price (VWAP), have proven effective in cryptocurrency trading. Zarattini and Aziz [11] emphasized the role of VWAP in day trading strategies, where it is used to assess market sentiment and identify overbought or oversold conditions.

In the cryptocurrency domain, Fischer and Krauss [4] illustrated the potential of Long Short-Term Memory (LSTM) networks for directional predictions, capturing temporal dependencies in price data. However, ensemble methods like

Support Vector Machines (SVM) and Random Forests, as demonstrated by Liew and Mayster [12], offer alternative approaches with robust performance in diverse financial contexts. Zhu et al. [13] extended these findings by proposing hybrid neural networks integrating convolutional and LSTM layers, showcasing innovative architectures for temporal and spatial feature extraction.

By situating this study within the existing literature, we address a specific gap: the unexplored impact of combining EMA and VWAP indicators with machine learning models for cryptocurrency price forecasting. This research builds on prior work by evaluating the trade-offs between feature inclusion, model complexity, and predictive performance in a high-volatility domain.

3. Methodology

The dataset used in this study spans Bitcoin prices from September 17, 2014, to November 30, 2023. This period was specifically chosen as it captures multiple market cycles, including the notable bull runs of 2017 and 2021, as well as subsequent bear markets, providing a comprehensive view of Bitcoin's price dynamics across different market conditions. The raw dataset includes daily observations of four features: Open Price, High Price, Low Price, and Trading Volume. The Close Price, expressed in USD, serves as our target variable and represents Bitcoin's final trading value for each day. Trading Volume is measured in BTC, reflecting the total number of Bitcoin traded daily.

Table 1 shows a snapshot of our dataset, including the original features (Open Price, High Price, Low Price, and Trading Volume) along with additional features used in our analysis, which are introduced and discussed in detail later in the paper.

Table 1. Head of Bitcoin price data and all derived features. The table displays daily Bitcoin price data, including Open, Low, High, Volume (in BTC), and Close prices. Derived technical indicators include EMA20, EMA50, and EMA100 (20-day, 50-day, and 100-day exponential moving averages) and VWAP (Volume Weighted Average Price). Data is based on daily observations from Bitcoin trading history

Date	Open	Low	High	Volume	Close
2014-12-25	322.286011	316.958008	322.670013	9,883,640	319.007996
2014-12-26	319.152008	316.627014	331.424011	16,410,500	327.924011
2014-12-27	327.583008	312.630005	328.911011	15,185,200	315.863007
2014-12-28	316.160004	311.078003	320.028015	11,676,600	317.239014
2014-12-29	317.700989	312.307007	320.266998	12,302,500	312.670013
Date	Open	Low	High	Volume	Close
2014-12-25	337.460187	353.084240	366.468409	367.618761	
2014-12-26	336.551980	352.097565	365.705154	367.336745	
2014-12-27	334.581601	350.676602	364.718181	367.041155	
2014-12-28	332.929926	349.365324	363.777999	366.802144	
2014-12-29	331.000411	347.926292	362.765960	366.547651	

To provide a comprehensive overview of the dataset used in this study, Table 2 presents the summary statistics of key features, including the Close, Open, High, Low, and Volume.

Table 2. Summary statistics of bitcoin price data (2014-2023)

	Close	Open	High	Low	Volume
Mean	14,325.23	14,315.47	14,652.26	13,951.70	16,486,916,451
Std Dev	16,037.74	16,037.46	16,426.40	15,599.49	19,149,538,687
Min	178.10	176.90	211.73	171.51	5,914,570
Max	67,566.83	67,549.73	68,789.62	66,382.06	350,967,941,479

The summary statistics reveal significant variability in Bitcoin prices and trading activity over the study period. The average closing price of Bitcoin was 14,325.23, but the values ranged widely, from a minimum of 178.10 to a maximum of 67,566.83, reflecting the extreme volatility inherent to cryptocurrency markets. The standard deviation of 16,037.74 further underscores the substantial price fluctuations within the dataset. A similar pattern is observed for trading volume, which exhibits a large mean value of approximately 16.49 billion BTC but fluctuates between a minimum of around 5.91 million BTC and a maximum of 350.97 billion BTC. These statistics highlight the challenges of forecasting in such a volatile environment, where extreme highs and lows can occur within short periods.

The data summarized in Table 2 illustrates the dynamic nature of the cryptocurrency market and the inherent complexities in developing robust prediction models. The high degree of variability, as evidenced by the wide range between minimum and maximum values and the large standard deviations, emphasizes the need for sophisticated methodologies to account for these fluctuations in price and trading activity.

To enhance our predictive capabilities, we incorporated two widely used technical indicators: Exponential Moving Averages (EMA) and Volume Weighted Average Price (VWAP). These indicators were selected based on their prevalent use in cryptocurrency trading and their demonstrated effectiveness in previous studies.

For the EMAs, we implemented three specific periods: 20-day, 50-day, and 100-day-to capture short-term, medium-term, and long-term price trends respectively. These specific periods were chosen as they represent standard intervals used by traders and have shown effectiveness in previous cryptocurrency studies (cite recent study). The EMA is calculated as:

$$\text{EMAlen}(t) = (V_t \times \alpha) + (\text{EMAlen}(t-1) \times (1-\alpha))$$

Where:

- len is the length of the EMA (20, 50 and 100 in this paper).
- EMAlen(t) is the EMAlen at time t.
- V_t is the value at time t.
- α is the smoothing factor, typically between 0 and is inversely proportional to the length of the EMA.
- The initial EMAlen (0), is simple moving average (SMA) of the first len closing prices.

The VWAP indicator complements these trend indicators by providing a volume-weighted perspective of price movements, offering insights into market sentiment and trading activity.

VWAP is an indicator that calculates the average price of a security over a specific period, weighted by the volume traded at each price level. VWAP is often used by institutional investors to execute large orders with minimal market impact. It can also be used by traders to gauge the overall market sentiment towards a security and to identify potential overbought or oversold conditions. It is calculated as:

$$\text{VWAP} = \frac{\Sigma\ (\text{Price} \times \text{Volume})}{\Sigma\ \text{Volume}},$$

VWAP provides a comprehensive measure of market sentiment by combining price and trading volume over the day.

Our data preprocessing pipeline began with thorough cleaning procedures, including the removal of the initial 100 rows to eliminate null values resulting from EMA calculations. Additionally, the Date column was transformed into a datetime object series. This transformation facilitates temporal analysis by enabling a more intuitive and standardized representation of time. It streamlines chronological order and time-based calculations. This approach empowers us to extract valuable insights related to trends, seasonality, and temporal dependencies within the Bitcoin price data. We implemented StandardScaler for numerical features while carefully preserving the temporal ordering of our data. This scaling approach ensures that all features contribute proportionally to the model training process without introducing future information bias. To evaluate the impact of technical indicators systematically, we created progressive feature sets. The base set included the fundamental price and volume data, while subsequent sets incorporated the EMAs and VWAP indicators. This approach allowed us to assess how the addition of technical indicators affected model performance. The dataset was then chronologically split into training (80%) and testing (20%) sets, maintaining the temporal nature of the data crucial for financial forecasting applications.

For model evaluation, we selected Mean Absolute Error as our primary evaluation metric due to its interpretability and robustness to outliers, which is particularly important given Bitcoin's price volatility. MAE provides a clear, direct measure of prediction accuracy in the same unit as the target variable (USD). The MAE is calculated as:

$$\mathrm{MAE} = \frac{1}{n} \sum_{j=1}^{n} \left| y_j - \hat{y}_j \right|,$$

Where:
- $\hat{y}_j$ represents the model's predicted Bitcoin price.
- y_j is the actual observed price.
- n is the number of samples in the test set.

The Mean Absolute Error (MAE) metric offers several advantages that make it highly suitable for Bitcoin price prediction. One key benefit is its direct interpretability. MAE values are expressed in the same units as the target variable, which in this case is US dollars. This makes it easier to understand the average magnitude of prediction errors and assess model performance in a practical and intuitive way. Another important advantage of MAE is its robustness. In the volatile cryptocurrency market, where sudden and extreme price fluctuations are common, MAE performs well by being less sensitive to outliers compared to metrics like Root Mean Squared Error (RMSE). This robustness ensures that the metric provides a reliable evaluation of model accuracy even in the presence of sharp price changes. Finally, MAE provides consistency by treating all error magnitudes linearly. Unlike squared error metrics, which place more emphasis on larger errors, MAE gives equal weight to all errors. This characteristic ensures a stable and straightforward evaluation of model performance across different market conditions, making it a reliable choice for assessing predictive accuracy. These qualities of interpretability, robustness, and consistency make MAE an effective and practical metric for evaluating the performance of Bitcoin price prediction models in volatile market environments.

Our modeling approach involved three distinct architectures: Random Forest Regressor (RFR), Support Vector Regression (SVR), and Long Short-Term Memory (LSTM) networks. Each model was selected based on its unique strengths in handling complex and nonlinear relationships in the dataset.

3.1 *Random forest regressor (RFR)*

The Random Forest Regressor (RFR) is an ensemble learning method that constructs multiple decision trees and aggregates their predictions to improve accuracy and prevent overfitting. It is particularly effective for datasets with nonlinear relationships, and interactions between features. This robustness makes RFR a reliable choice for financial datasets, where noise and outliers are common. As noted by Fama et al. [14], Random Forest models have been validated for stock price forecasting and perform well in high-volatility environments.

To optimize the performance of the RFR, we employed GridSearchCV, a systematic method for hyperparameter tuning. This process evaluates various values for critical parameters, such as the number of estimators (trees), tree depth, and splitting criteria, to identify the best combination. For instance, the number of estimators influences the ensemble size and directly affects model performance. Rigorous tuning with GridSearchCV and 5-fold cross-validation ensured that the selected hyperparameters maximized predictive accuracy while maintaining generalizability.

3.2 *Support vector regression (SVR)*

Support Vector Regression (SVR) serves as a robust regression strategy, preserving the core principles of Support Vector Machine (SVM) while adeptly adapting to the nuances of regression analysis. SVR, akin to SVM, is predicated on the concept of maximizing the margin between data points and concurrently minimizing the prediction error within an established tolerance threshold. SVR is designed to handle continuous numerical outcomes, making it particularly effective for predictive modeling in various domains. Chen et al. [15] and Henrique et al. [16] demonstrated that Support Vector Machines (SVM) and Support Vector Regression (SVR) are effective tools for predicting stock prices, particularly in volatile markets.

The dataset was preprocessed with a Standard Scaler to standardize features by centering them around zero and scaling them to unit variance. This preprocessing step is crucial for SVR, as the algorithm is sensitive to the scale of input data. We performed extensive hyperparameter tuning using GridSearchCV with 5-fold cross-validation to identify the best configurations for each feature set size (4, 7, and 8 features). This included fine-tuning key parameters such as:

• C: Controls the trade-off between fitting the data and maintaining a simple model. A larger C gives more weight to minimizing errors.

• ε (epsilon): Defines the margin of tolerance for prediction errors.

• Gamma: For RBF kernels, gamma determines the influence of individual data points.

SVR models were trained with two kernel functions. The Radial Basis Function (RBF) kernel, which maps inputs into higher-dimensional space, allowed the model to capture nonlinear relationships, making it ideal for financial data. The Polynomial kernel, which captures interactions between features, required careful tuning to avoid overfitting.

3.3 *Long short-term memory (LSTM) networks*

LSTM networks, a type of deep learning model, are particularly promising for time-series data due to their ability to retain relevant information from past inputs and use it for future predictions. This internal memory mechanism makes them well-suited for tasks involving temporal dependencies and long-term relationships within data. Singh et al. [17] highlighted their potential for forecasting trends in volatile markets, such as Bitcoin.

Building on these strengths, we developed two distinct LSTM model architectures, Model 1 and Model 2, each designed with varying configurations to evaluate their performance. Both models share the same input and output layers but differ in their hidden layer configurations. The input layer, with a dimensionality of 50, incorporates 50 days of historical trading data, while the output layer predicts the closing price of Bitcoin for the following day. This 50-day window balances the need to capture relevant historical context while avoiding excessive noise from short-term fluctuations. For model training, we employed a batch size of 32, facilitating a thorough search for optimal solutions by the optimization algorithm. To mitigate overfitting, we implemented a dropout rate of 20 percent, randomly deactivating neurons during each training iteration. Additionally, an early stopping mechanism was incorporated to halt training if the validation error did not decrease for 15 consecutive epochs.

A critical aspect of training these LSTM models was the selection of activation functions and optimizers, both of which play a central role in the learning process of neural networks. Activation functions determine the output of a neuron based on its input. Rectified Linear Unit (ReLU) is defined as $f(x) = \max(0,\ x)$. It introduces non-linearity and is computationally efficient, helping to mitigate the vanishing gradient problem in deep networks. ReLU activates neurons only when their input is positive, making it suitable for sparse activation. Hyperbolic Tangent (Tanh) maps inputs to a range of $(-1,\ 1)$, making it useful for scenarios where outputs are centered around zero. It can model stronger relationships by considering both positive and negative inputs but is prone to vanishing gradients in deep networks. The choice between ReLU and Tanh depends on the task and the data. ReLU is generally preferred for its simplicity and performance, while Tanh might be more appropriate for tasks requiring balanced output distributions.

Equally important to the training process is the choice of optimizer, which determines how the model's weights are updated to minimize the loss function. Adam, a widely used optimizer, adapts learning rates for each parameter, providing robust performance across a variety of tasks. SGD updates weights iteratively, making it computationally efficient but potentially slower in convergence, particularly in complex models. RMSprop adjusts learning rates based on gradient variance, making it especially suitable for non-stationary objectives like financial data. The selection of an optimizer can significantly impact model performance, with Adam offering robust and adaptive behavior, while SGD and RMSprop are often chosen for specific scenarios requiring finer control over learning dynamics.

Model 1 configuration: The architecture of Model 1, a foundational configuration for our LSTM-based approach, is outlined in Table 3. This model is designed to process sequential input data and predict the next-day closing price of Bitcoin. The input layer accepts data with a dimensionality of 50, corresponding to 50 days of historical trading data and the number of features (N). This is followed by a single LSTM layer, which learns temporal dependencies in the data. A dropout layer is applied to mitigate overfitting by randomly deactivating neurons during training. Finally, the dense output layer generates a single scalar prediction for the next day's closing price.

Table 3. Architecture of Model 1: This table describes the sequential layers of the model. The input layer processes 50 days of historical trading data with N features. The LSTM layer captures temporal dependencies, while the dropout layer reduces overfitting. The dense layer outputs a single prediction for the next day's Bitcoin closing price

Layer (type)	Output Shape
input_layer (InputLayer)	(None, 50, N)
LSTM1 (LSTM)	(None, 50)
Dropout1 (Dropout)	(None, 50)
Dense1 (Dense)	(None, 1)

To evaluate the impact of different hyperparameter choices, we trained six variations of Model 1. Each variation used a different combination of activation function (ReLU or Tanh) and optimizer (Adam, SGD, or RMSprop) for the LSTM layer. These variations allowed us to assess the sensitivity of the model's performance to the choice of activation functions and optimization strategies. Table 4 details these configurations.

Table 4. Model 1 Variations: Six configurations of Model 1, each with a unique combination of optimizer and activation function for the LSTM layer. This diversity enables a comprehensive analysis of how these hyperparameters influence predictive performance

Model	Optimizer	Activation function for hidden layers
1.0	Adam	ReLU
1.1	Adam	Tanh
1.2	SGD	ReLU
1.3	SGD	Tanh
1.4	RMSprop	ReLU
1.5	RMSprop	Tanh

Model 2 Configuration: Building upon Model 1, we designed Model 2 to further explore the performance impact of increased complexity in LSTM architectures. The architecture of Model 2 is described in Table 5, which includes an additional LSTM layer and dropout layer compared to Model 1. This second LSTM layer is intended to capture deeper temporal dependencies in the data, while the second dropout layer further mitigates overfitting by increasing regularization. The dense output layer remains unchanged, producing a single prediction for the next day's Bitcoin closing price.

Table 5. Architecture of Model 2: This table details the sequential layers of Model 2, which extends Model 1 by including a second LSTM layer and dropout layer. The second LSTM layer captures deeper temporal dependencies, while the dropout layers prevent overfitting. The dense layer outputs a single prediction for the next day's Bitcoin closing price

Layer (type)	Output shape	Parameters
LSTM 1 (LSTM)	(None, 50, 50)	11,000
Dropout 1 (Dropout)	(None, 50, 50)	0
LSTM 2 (LSTM)	(None, 50)	20,200
Dropout 2 (Dropout)	(None, 50)	0
Dense 1 (Dense)	(None, 1)	51

To assess the effect of these architectural enhancements, we trained six variations of Model 2, each using a unique combination of activation function (ReLU or Tanh) and optimizer (Adam, SGD, or RMSprop). These variations were systematically evaluated to identify the configurations that yielded the best predictive performance. Table 6 summarizes the six configurations of Model 2.

Table 6. Configurations of Model 2 Variants: This table lists six variations of Model 2, detailing the combinations of optimizers and activation functions used. Each configuration was evaluated to assess its impact on predictive performance

Model	Optimizer	Activation function for hidden layers
2.0	Adam	ReLU
2.1	Adam	Tanh
2.2	SGD	ReLU
2.3	SGD	Tanh
2.4	RMSprop	ReLU
2.5	RMSprop	Tanh

4. Results

Table 7 summarizes the performance of the machine learning models evaluated in this study, measured using Mean Absolute Error (MAE).

Our analysis reveals significant differences in performance across the machine learning models evaluated in this study. As shown in Table 7, the Support Vector Regressor (SVR) with the RBF kernel at $N = 4$, $C = 50$, and $\varepsilon = 0.001$ emerged as the best-performing model overall, achieving the lowest MAE of 384.7 USD. This result underscores the RBF kernel's ability to capture complex nonlinear relationships in Bitcoin price data.

The Random Forest Regressor (RFR) configurations also demonstrated competitive performance among the simpler models, particularly with $N = 4$ features and 20 estimators (MAE = 563.25 USD). This lower-dimensional configuration suggests that RFR is robust to noise and can effectively handle implicit feature selection, especially in high-volatility financial datasets where irrelevant features often exacerbate overfitting.

By contrast, SVR with polynomial kernels generally performed poorly, with MAE values exceeding 14,000 USD, likely due to overfitting and the inability to generalize in higher-dimensional feature spaces.

Turning to the LSTM-based neural networks, their performance tended to degrade as the feature set size increased. Model 2.5 (RMSprop, tanh) achieved an MAE of 862.2 USD with $N = 4$ features but performed worse with higher N. This trend illustrates the challenges posed by high-dimensional inputs, including increased susceptibility to overfitting and higher computational cost. The additional LSTM layer in Model 2 added capacity for learning complex patterns but also introduced greater vulnerability to noise when trained on smaller datasets or large feature sets.

Table 7. Model performance summary

Model	Number of features N	Configuration details	MAE
Random forest regressor	4	20 estimators	563.25
	7	10 estimators	1,610.42
	8	10 estimators	2,432.99
SVR (Polynomial kernel)	4	degree 2	14,316.53
	4	degree 3	18,545.12
	7	degree 2	13,570.89
	7	degree 3	14,840.51
	8	degree 2	60,056.22
	8	degree 3	22,986.56
SVR (RBF Kernel)	4	C = 50, $\varepsilon = 0.001$	384.7
	7	C = 100, $\varepsilon = 0.001$	786.1
	8	C = 50, $\varepsilon = 0.001$	857.98
1.0	4	Adam, ReLU	1,527.92
	7	Adam, ReLU	2,832.8
	8	Adam, ReLU	6,888.7
1.1	4	Adam, tanh	1,223.04
	7	Adam, tanh	2,752.53
	8	Adam, tanh	2,185.7
1.2	4	SGD, ReLU	4,536.86
	7	SGD, ReLU	8,759.3
	8	SGD, ReLU	12,247.7
1.3	4	SGD, tanh	2,656.3
	7	SGD, tanh	2,539.22
	8	SGD, tanh	2,464.25
1.4	4	RMSprop, ReLU	1,199.14
	7	RMSprop, ReLU	2,159.6
	8	RMSprop, ReLU	3,576.28
1.5	4	RMSprop, tanh	716.79
	7	RMSprop, tanh	1,320.54
	8	RMSprop, tanh	1,892.91
2.0	4	Adam, ReLU	3,643.46
	7	Adam, ReLU	6,046.53
	8	Adam, ReLU	9,707.9
2.1	4	Adam, tanh	1,450.5
	7	Adam, tanh	3,350.9
	8	Adam, tanh	3,599.9
2.2	4	SGD, ReLU	8,757.6
	7	SGD, ReLU	8,098.8
	8	SGD, ReLU	7,258.8
2.3	4	SGD, tanh	3,639.49
	7	SGD, tanh	4,499.3
	8	SGD, tanh	3,867.4
2.4	4	RMSprop, ReLU	1,192.48
	7	RMSprop, ReLU	3,212.87
	8	RMSprop, ReLU	5,365.62
2.5	4	RMSprop, tanh	862.2
	7	RMSprop, tanh	856.3
	8	RMSprop, tanh	1,339.7

5. Conclusions and further research

This study demonstrates that simpler machine learning models, particularly Random Forest Regressors and Support Vector Regression (SVR) with Radial Basis Function (RBF) kernels, outperform more complex deep learning approaches in cryptocurrency price forecasting. The findings emphasize the importance of balancing model complexity with the quality and dimensionality of the available data. Effective noise management and focused feature selection were shown to be critical for achieving superior performance, particularly in the volatile and high-noise environment of cryptocurrency markets.

Several limitations of this study should be acknowledged. The analysis focused solely on Bitcoin price data, which, while significant due to Bitcoin's market dominance, may not fully capture the dynamics of other cryptocurrencies that exhibit different behaviors. The temporal scope of the dataset, though extensive, might not encompass all possible market scenarios, such as extreme economic events or atypical market cycles. Additionally, the technical indicators used in this study, specifically Exponential Moving Averages (EMAs) and Volume Weighted Average Price (VWAP), represent only a small subset of the tools available for financial forecasting.

These limitations highlight opportunities for future research. Expanding the analysis to include a wider variety of cryptocurrencies could help uncover asset-specific patterns and improve the generalizability of the findings. Exploring additional technical indicators, such as momentum oscillators, Bollinger Bands, or the Relative Strength Index (RSI), could lead to more effective feature sets for prediction. Future studies could also investigate hybrid approaches that combine traditional machine learning models with deep learning architectures. For example, ensemble strategies or attention-based models might address some of the limitations observed in pure Long Short-Term Memory (LSTM) implementations. Furthermore, applying advanced regularization techniques and experimenting with alternative optimization strategies could reduce the risk of overfitting, particularly in high-dimensional datasets.

Alternative data sources, such as social media sentiment, blockchain network activity, and macroeconomic indicators, could further enhance predictive models by offering additional insights into market behavior and external factors influencing price movements. Techniques designed to improve model interpretability, such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), can help clarify the decision-making processes of machine learning models. SHAP assigns each input feature a contribution value, showing how much it influences the model's prediction for a specific outcome. LIME approximates complex models locally with simpler ones, providing insights into the factors most important for individual predictions. These tools enable researchers and practitioners to better understand how models arrive at their results, promoting greater transparency and building confidence in the reliability of predictive systems.

The findings of this study underline the practical value of simpler, well-tuned machine learning models in certain contexts, especially when compared to more complex deep learning approaches. Simpler models, such as Random Forest Regressors, provide robust performance and computational efficiency, making them suitable for real-world applications where resources or data availability may be limited. The sensitivity of model performance to feature set size further underscores the importance of disciplined feature engineering and selection processes to achieve optimal accuracy and efficiency.

In conclusion, while cryptocurrency price forecasting remains a challenging and evolving field, this study demonstrates that traditional machine learning models, when properly implemented, can deliver robust and effective solutions. These findings contribute to a deeper understanding of the trade-offs between model complexity and predictive accuracy, offering valuable guidance for theoretical advancements and practical applications in financial forecasting.

Data availability

The dataset analyzed during the current study is available on Yahoo Finance, [https://finance.yahoo.com/quote/BTC-USD/history].

Conflict of interest

The author declares no competing interests.

References

[1] Kelly B, Xiu D. Financial machine learning. *Foundations and Trends in Finance*. 2023; 13(3-4): 205-363. Available from: https://doi.org/10.1561/0500000064.

[2] Coqueret G, Guida T. *Machine Learning for Factor Investing: R Version*. 1st ed. New York: Chapman and Hall/CRC; 2020.

[3] Moghar A, Hamiche M. Stock market prediction using lstm recurrent neural network. *Procedia Computer Science*. 2020; 170: 1168-1173.

[4] Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. 2018; 270(2): 654-669.

[5] Chen L, Pelger M, Zhu J. Deep learning in asset pricing. *Management Science*. 2024; 70(2): 714-750. Available from: https://doi.org/10.1287/mnsc.2023.4695.

[6] Zhou X, Zhou H, Long H. Forecasting the equity premium: Do deep neural network models work? *Modern Finance*. 2023; 1(1): 1-11. Available from: https://doi.org/10.61351/mf.v1i1.2.

[7] Labiad B, Berrado A, Benabbou L. Machine learning techniques for short term stock movements classification for Moroccan stock exchange. In: *Intelligent Systems: Theories and Applications (SITA), 11th International Conference*. Mohammedia, Morocco: IEEE; 2016. p.1-6.

[8] Illa PK, Parvathala B, Sharma AK. Stock price prediction methodology using random forest algorithm and support vector machine. *Materials Today: Proceedings*. 2022; 56(4): 1776-1782. Available from: https://doi.org/10.61351/mf.v1i1.2.

[9] Fieberg C, Liedtke G, Poddig T, Walker T, Zaremba A. A trend factor for the cross-section of cryptocurrency returns. *Journal of Financial and Quantitative Analysis*. 2024.

[10] Raza H, Akhtar Z. Predicting stock prices in the Pakistan market using machine learning and technical indicators. *Modern Finance*. 2023; 2(2): 46-63. Available from: https://doi.org/10.61351/mf.v2i2.167.

[11] Zarattini C, Aziz A. Volume weighted average price (VWAP): The holy grail for day trading systems. *Social Science Research Network*. 2023.

[12] Liew JKS, Mayster B. Forecasting ETFs with machine learning algorithms. *Journal of Alternative Investments*. 2017; 20(3): 58-78.

[13] Zhu R, Zhong GY, Li JC. Forecasting price in a new hybrid neural network model with machine learning. *Expert Systems with Applications*. 2024; 249: 123697. Available from: https://doi.org/10.1016/j.eswa.2024.123697.

[14] Fama EF. Efficient capital markets: II. *Journal of Finance*. 1991; 46(45): 1575-1617.

[15] Chen W, Shih J, Wu S. Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *International Journal of Electronic Finance*. 2006; 1(1): 49-67.

[16] Henrique BM, Sobreiro VA, Kimura H. Stock price prediction using support vector regression on daily and up to the minute prices. *Journal of Finance and Data Science*. 2018; 4(3): 183-201. Available from: https://doi.org/10.1016/j.eswa.2024.123697.

[17] Singh S, Pise A, Yoon B. Prediction of bitcoin stock price using feature subset optimization. *Heliyon*. 2024; 10(7): e28415.