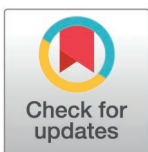


RESEARCH ARTICLE

LSTM-conformal forecasting-based bitcoin forecasting method for enhancing reliability

Xiangyue Zhang¹, Yuyun Kang^{1,2*}, Chao Li³, Wenjing Wang¹, Keqing Wang¹

1 School of Information Science and Engineering, Linyi University, Linyi, Shandong, China, **2** School of Logistics, Linyi University, Linyi, Shandong, China, **3** Daopuyun (Shandong) Intelligent Technology Co., Ltd., Jinan, Shandong, China

* kangyuyun@lyu.edu.cn

Abstract

Cryptocurrency is a new type of asset that has emerged with the advancement of financial technology, creating significant opportunities for research. Bitcoin is the most valuable cryptocurrency and holds significant research value. However, due to the significant fluctuations in Bitcoin's value in recent years, predicting its value and ensuring the reliability of these predictions, which have become crucial, have gained increasing importance. A method that combines Long Short-term Memory (LSTM) with conformal prediction is proposed in this paper. Initially, the high-dimensional features in the dataset are divided using the Spearman correlation coefficient method, and features below 0.75 and above 0.95 are excluded. Subsequently, an LSTM model is built, and data are fed into it and the data is used to train the model to generate predictions. Finally, the predicted values generated by the LSTM are fed into the conformal prediction model, and confidence intervals for these values are generated to verify their reliability. In the conformal prediction model, the quantile loss of the loss function is defined, and an Average Coverage Interval (ACI) predictor is designed to improve the accuracy of the results. The experiments are conducted using data from CoinGecko, which is a publicly available data. The results show that the LSTM-conformal prediction (LSTM-CP) combination improves reliability.

OPEN ACCESS

Citation: Zhang X, Kang Y, Li C, Wang W, Wang K (2025) LSTM-conformal forecasting-based bitcoin forecasting method for enhancing reliability. *PLoS One* 20(5): e0319008. <https://doi.org/10.1371/journal.pone.0319008>

Editor: Yang (Jack) Lu, Beijing Technology and Business University, CHINA

Received: November 26, 2024

Accepted: January 24, 2025

Published: May 2, 2025

Copyright: © 2025 Zhang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data availability statement: All bitcoin files are available from the QDR Main Collection database (accession number(s) 10.5064/F6QF1YQY).

Funding: Shandong Province innovation ability improvement project of science and technology SMEs (No. 2022TSGC2176), Research and application of key technologies of cargo distribution and transportation based on BDS and 5G (No. CYY-TD-2022-004). Yuyun Kang is

1. Introduction

Bitcoin value prediction is crucial for market traders. Recently, Bitcoin has rapidly dominated the trading market. Recently, turmoil in the international financial markets has led to significant fluctuations in Bitcoin market transactions (as shown in [Fig 1](#)). Accurate Bitcoin value predictions can lead to significant returns for investors and businesses. For instance, accurate predictions enable individual investors to make informed decisions. Predicting future price movements allows investors to optimize returns or minimize losses by deciding whether to buy, hold, or sell. Financial analysts and businesses can analyze market trends and devise strategies using Bitcoin value predictions. For example, if forecasts show a rising Bitcoin value, businesses might increase their Bitcoin holdings or offer Bitcoin payments. Additionally, value forecasting serves as a vital risk management tool. Understanding potential price swings enables investors and businesses to plan and protect against extreme market fluctuations. This involves strategies such as stop-loss orders and portfolio diversification.

the funder, he participated in study design and publication decision and paper revision.

Competing interests: The authors have declared that no competing interests exist.

In the early days, scholars extensively researched bitcoin value prediction. Sin et al. [1] introduced a selective neural network ensemble method based on genetic algorithms for predicting bitcoin value. The relationship between bitcoin price changes over the next day and its features is explored, using a multilayer perceptron for each feature in an ensemble neural network. Experimental results showed that this strategy generated nearly 85% returns, outperforming the “previous day trend following” strategy (38% return) and the best single Multi-layer Perceptron (MLP) model strategy (53% return). Afterward, Struga et al. [2] adopted an LSTM-based recurrent neural network for bitcoin price prediction, demonstrating the effectiveness of the LSTM architecture on time series data. Results included bitcoin price predictions for the next 30 and 60 days. Meanwhile, Azari et al. [3] demonstrated the effectiveness of the Autoregressive Integrated Moving Average (ARIMA) model in predicting the value of bitcoin over three years. Empirical studies found it effective in stable periods, especially for short-term, one-day forecasts. However, given bitcoin’s price volatility, training ARIMA for more than three years or long-term forecasts would introduce significant errors.

On this basis, more in-depth research is conducted on previous studies. Poongodi et al. [4] explored the technology of the bitcoin network and diverse machine learning prediction algorithms. The public bitcoin blockchain data set (April 28, 2013 - July 31, 2017) is used [5], and ARIMA is applied for price prediction. Afterward, A deep learning method, the Stacked Denoising Autoencoder (SDAE), is used to predict bitcoin prices by Liu et al. [6]. The results showed that the SDAE model outperformed popular machine learning methods like Back-propagation Neural Network (BPNN) and Support Vector Regression (SVR) in both directional and horizontal predictions. To investigate the dynamic tail risk of the bitcoin market. Gao et al. [7] use the Conditional Autoregressive Value-at-risk (CAViaR) model to jointly estimate Value-at-risk (VaR) and Expected Shortfall (ES) to explore the dynamic tail risk of the bitcoin market. To achieve more accurate measurements, we constructed a Markov Regime-switching (MS) model with time-varying transition probabilities driven by information from asset price bubbles. Experimental results using daily bitcoin data from 2013 to 2021 provide strong evidence for structural changes in bitcoin’s VaR and ES. Furthermore, the bubble index significantly impacts tail risk and enhances the model’s ability to estimate and predict VaR and ES. In the subsequent study, Chen et al. [8] combined random forest regression and LSTM to create a highly accurate model for next-day bitcoin price prediction,

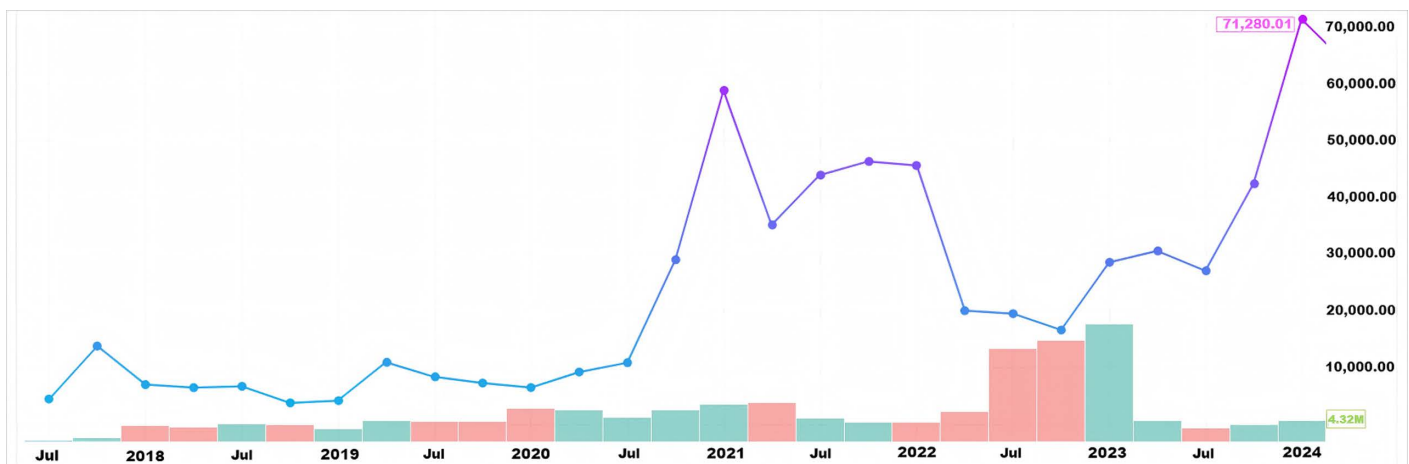


Fig 1. The trends in bitcoin prices from 2018 to 2024.

<https://doi.org/10.1371/journal.pone.0319008.g001>

detailing an impact analysis showing that the model with a single lagged explanatory variable had a negative impact on next-day bitcoin price prediction. The currency price prediction had the highest accuracy. Moreover, Ranjan et al. [9] adopted a variety of machine learning techniques to predict bitcoin prices. Experimental results showed that logistic regression could predict daily price predictions with 64.84% accuracy, while eXtreme Gradient Boosting (XGBoost) could predict 5-minute intervals with 59.4% accuracy. Following this, Koo et al. [10] introduced Concentrated Cluster Distribution (CCD) as a new data filtering method to improve bitcoin price prediction by resolving its extreme bimodality. This improved tail and overall prediction accuracy. Experimental results showed that with LSTM and SSA decomposition, the CCD-WES strategy surpassed the baseline by reducing the Root Mean Square Error (RMSE) by 11.5% and 22.5% in the overall and extreme cases, respectively.

Although the number of bitcoin value prediction methods is increasing and their accuracy is improving, studies on reliability remain scarce. We emphasize that predictions should not only be accurate but also reliable. Therefore, A method combining LSTM with conformal prediction is proposed in this article to enhance prediction reliability.

The design of LSTM, which is tailored for time series data processing and forecasting, offers unique advantages in predicting bitcoin's value. LSTM's gating mechanism allows it to capture long-term dependencies. It enables LSTM to "remember" key past data for future predictions. Moreover, LSTM's structure offers robustness against data outliers and noise. Most notably, it exhibits significant dynamic adaptability. As market conditions and behaviors evolve, so do stock conditions. LSTM continuously learns from new data, adapting to changes and ensuring predictions remain accurate and reliable. This guarantees LSTM's peak performance in predicting bitcoin values.

Conformal forecasting provides a unique approach to predicting bitcoin's value by quantifying uncertainty and enhancing decision-making confidence. Its primary advantage lies in providing confidence intervals for predictions, thereby effectively quantifying uncertainty. It offers not just a point estimate of bitcoin's expected price but also a range that could encompass the actual price. Moreover, confidence intervals allow bitcoin users, both individuals and businesses, to conduct more accurate risk assessments. For instance, a wider confidence interval suggests higher volatility, indicating a prudent holding strategy. Importantly, the conformal prediction framework applies to various models (such as LSTM, linear regression, random forests), offering a universal method for quantifying uncertainty.

The structure of this paper is presented as follows. The common bitcoin prediction model, LSTM, ACI predictor and conformal prediction are introduced in section 2: Related work. The evaluation metrics of the conformal prediction model are detailed in section 3: Evaluation criteria. The dataset used, and the processing and partitioning of high-dimensional features in the dataset using the Spearman correlation coefficient method are described in section 4: Dataset. The experimental results, including advantages and disadvantages, are discussed in section 5: Experiment analysis. The main findings are concluded in section 6: Conclusion.

2. Related work

2.1. Bitcoin value forecasting

2.1.1. Methods in bitcoin value forecasting. With the continuous development of artificial intelligence, bitcoin predictions have expanded beyond traditional statistics and mathematics. Currently, bitcoin prediction methods tend to combine both approaches. Consequently, machine learning methods like XGBoost and Support Vector Machines (SVM), as well as deep learning methods like long short-term memory networks and Convolutional Neural Network (CNN), demonstrate better performance. An overview of the research framework is shown in Fig 2.

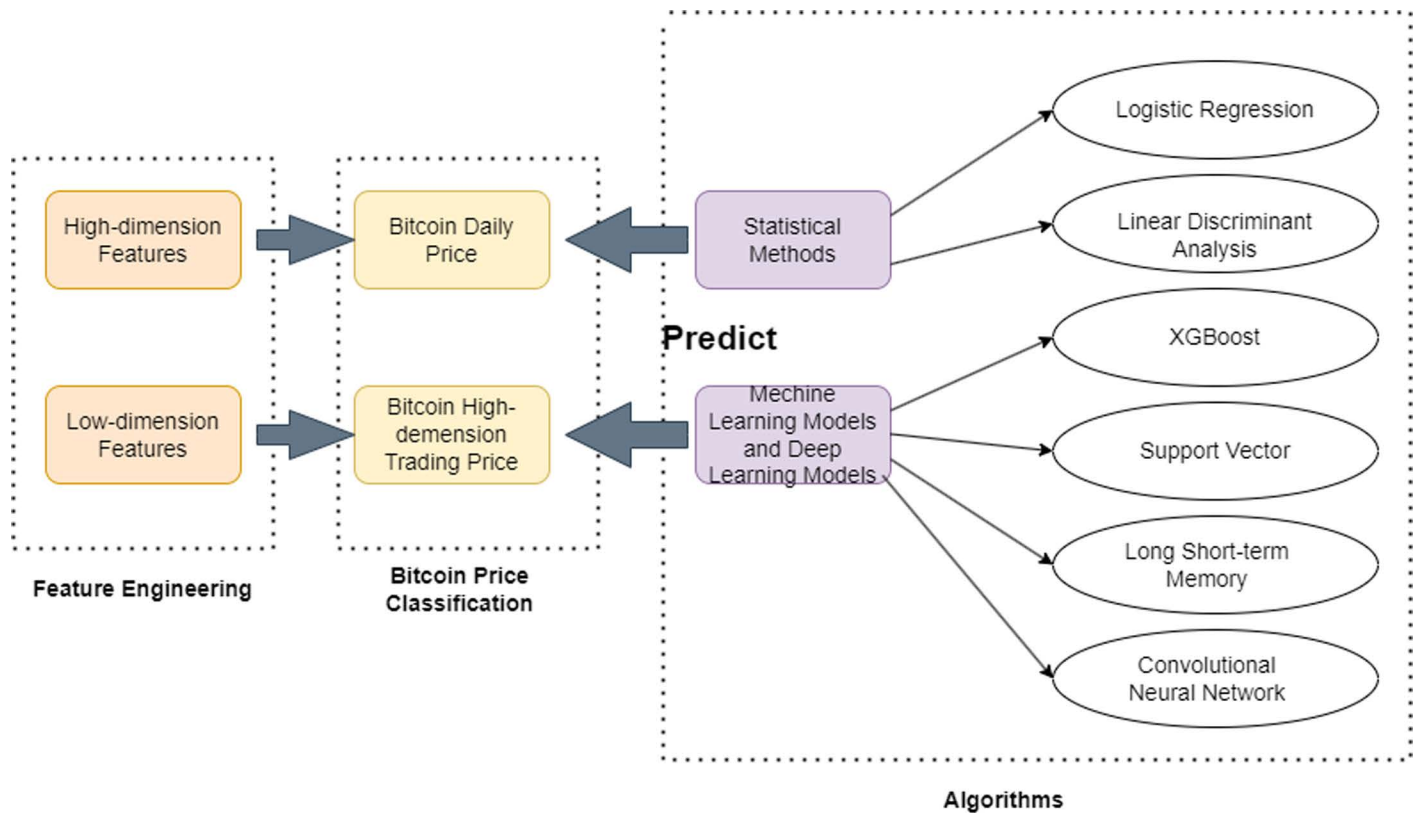


Fig 2. An overview of the framework used by machine learning methods for bitcoin price prediction.

<https://doi.org/10.1371/journal.pone.0319008.g002>

2.1.2. Classical bitcoin forecasting methods. Traditional methods for predicting bitcoin prices typically involve gathering data from social media, news reports, market comments, and other channels. Sentiment analysis seeks to comprehend how the market reacts to specific news events and their potential impact on price movements. A method for predicting cryptocurrency prices using news and historical price data is described by Vo et al. [11]. The findings indicate that sentiment analysis plays a key role in predicting cryptocurrency prices, given the interactive nature of financial activities. In the subsequent study, the use of sentiment analysis as a computational tool to predict the price of bitcoin and other cryptocurrencies over different time intervals is proposed by Wolk et al. [12]. Twitter and Google trends are used to predict short-term prices of major cryptocurrencies, highlighting how these social media platforms influence buying decisions. The findings indicate that people's psychological and behavioral attitudes significantly influence the prices of highly speculative cryptocurrencies.

2.1.3. Machine learning methods for bitcoin forecasting. The advent of machine learning has brought about innovative prediction methods for cryptocurrencies like bitcoin. bitcoin prices are predicted by McNally et al. [13] using Bayesian optimized Recurrent Neural Network (RNN) and LSTM networks. The results showed that the nonlinear deep learning method outperformed the less effective ARIMA prediction. The deep learning model is benchmarked on both GPU and CPU, and it trained 67.7% faster on GPU than on CPU. In the subsequent work, a framework for predicting daily bitcoin prices using advanced machine learning methods, taking into account specific exogenous and endogenous factors,

is examined by Dutta et al. [14]. RMSE is used as the evaluation metric. Experimental results showed that the Gated Recurrent Unit (GRU) model with recurrent dropout outperformed other well-known models. Subsequently, Hafid et al. [15] proposed a classification machine learning method to predict the market direction, whether it will be up or down. Key features including the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) are identified, and the data are fed into a machine learning model. Evaluation results showed that the proposed machine learning method could generate buy and sell signals with an accuracy of over 86%.

2.1.4. Deep learning methods for bitcoin forecasting. The emergence of deep learning has introduced a wider range of bitcoin prediction methods. Efficient deep learning-based prediction models, specifically utilizing LSTM and GRU, are developed by Awoke et al. [16] to resolve bitcoin price fluctuations with high accuracy. Their study compared these two time series deep learning techniques and demonstrated their effectiveness in predicting bitcoin prices. In the subsequent study, Jiang et al. [17] explored various deep learning networks and methods to improve Precision, such as min-max normalization, Adam optimizer, and windowed min-max normalization. It is found that MLPs are not suitable for predicting prices based on current trends due to lack of memory. LSTM models, especially when combined with past memory and GRU, provided the most accurate predictions.

2.1.5. Hybrid model for bitcoin forecasting. Theoretical and empirical studies indicate that the integration of various models, particularly from different methodologies, enhances predictive performance. Hybrid models are based on the rationale that a single model might not adequately capture the data-generating process or encompass all characteristics of the time series. Hybrid models reduce the risk of failure and enhance outcomes by merging multiple models to decrease dependence on any single, potentially inappropriate model.

Numerous studies have explored hybrid forecasting methods, frequently merging traditional and artificial intelligence techniques. A novel hybrid model is proposed by Hashish et al. [18] to address the challenge from both a descriptive and predictive perspective. Hashish employs hidden Markov models to describe the historical trends of cryptocurrencies and long short-term memory networks to predict future trends. The model proposed by Hashish has proven to be effective compared to traditional time series forecasting models such as ARIMA and traditional LSTM. In the subsequent study, Gao et al. [19] proposed a hybrid method that combines the advantages of non-stationary parametric models, such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH), with the nonlinear modeling capabilities of LSTM neural networks. The results show that their hybrid model achieves similar predictive performance in terms of Mean Square Error (MSE), Mean Absolute Error (MAE), and RMSE, but in terms of precision, accuracy, and F1-score with optimal hyperparameters.

2.2. LSTM

As shown in Fig 3, LSTM is a special type of RNN designed to process and predict long-term dependencies in time series data. LSTM addresses the problems of gradient vanishing and explosion in traditional RNNs by introducing three gates: input gate, forget gate, and output gate. The input gate controls the reception of new information, the forget gate determines how much old information to retain, and the output gate controls the transition from cell state to the final output. This gate structure allows LSTM to retain past information and use it when necessary. First, it decides what information to discard through the forget gate. Second, the input gate decides which new information to store in the cell state. Then, the cell state is updated to incorporate new information. Finally, the output gate determines the output based on the cell state and the current input.

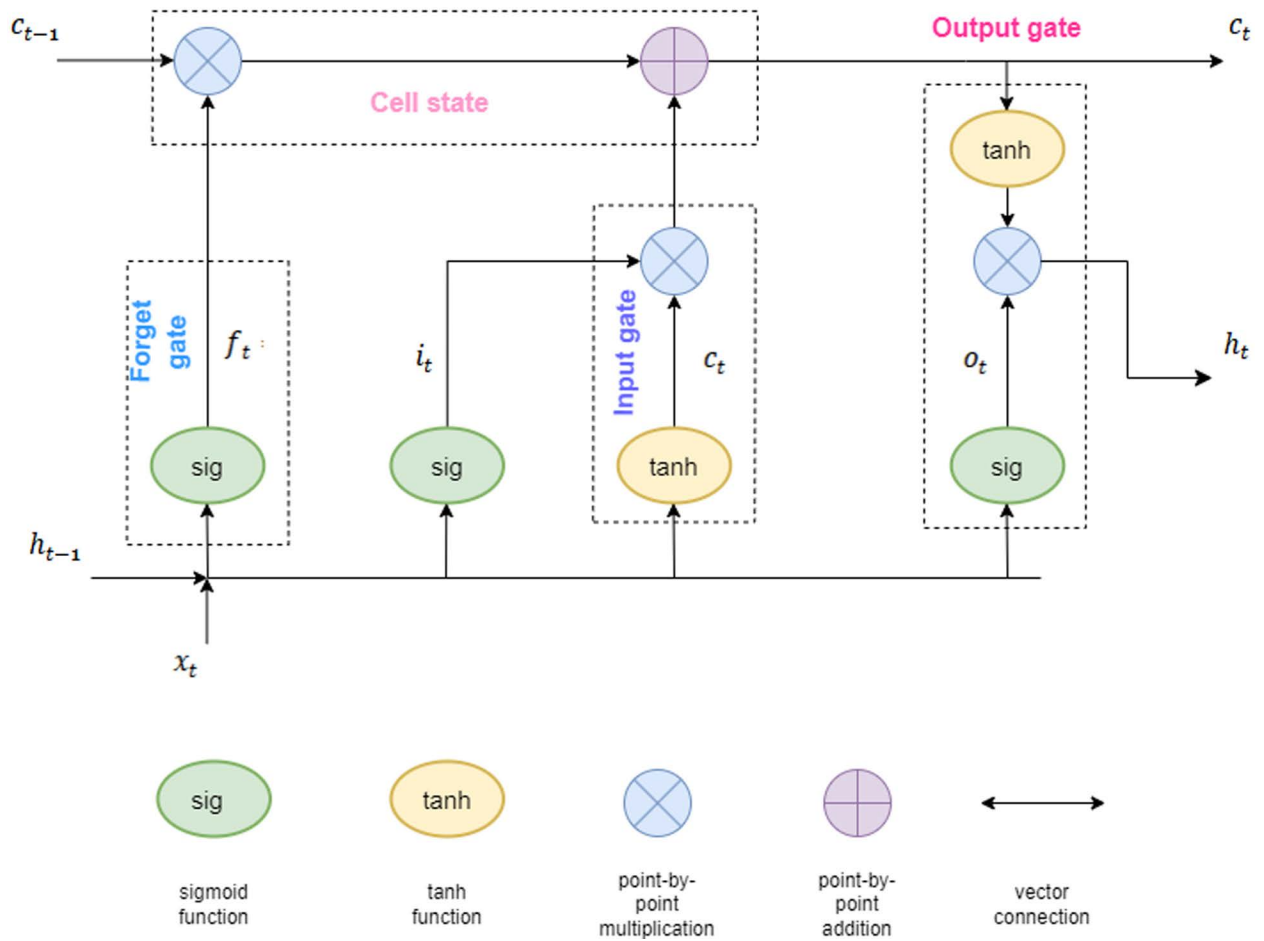


Fig 3. LSTM framework diagram.

<https://doi.org/10.1371/journal.pone.0319008.g003>

In the LSTM network, the equations for each time step operation are as follows, and Table 1 summarizes the notation used in this study:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{3}$$

$$c_t = (f \times c_{t-1}) + (i_t \times \sigma_c) \times (W_c x_t + U_c h_{t-1} + b_c), \tag{4}$$

$$h_t = o_t \times \tanh(c_t). \tag{5}$$

Table 1. A summary of the symbols used in the equations shown in LSTM.

Notation	Definition
f_t	The forget gate f_t determines how much of the previous cell state c_{t-1} is retained.
i_t	The input gate i_t decides how much new input information is stored in the cell state.
o_t	The output gate o_t determines how much of the current cell state c_t should be output.
c_t	The cell state c_t is the core of the LSTM, preserving crucial information over time.
h_t	The hidden state h_t contains the output information of the current time step.
W and U	W is the weight matrix for the input x_t , while U is the weight matrix for the previous time step's hidden state h_{t-1} .
b	Bias vector.
x_t	Input at time t .
h_{t-1}	Hidden state of the previous time step.
c_{t-1}	Cell state of the previous time step.
σ	The Sigmoid activation function, used to control the gates.
\tanh	The hyperbolic tangent activation function, used to process cell states and help regulate information flow.

<https://doi.org/10.1371/journal.pone.0319008.t001>

2.3. Conformal prediction

Conformal prediction [20] is a statistical framework that generates uncertainty intervals for the prediction of machine learning models. This study reviews several distribution-free prediction methods that provide guarantees under the assumption of exchangeability of the data and symmetry of the underlying algorithm.

Split conformal prediction: Split conformal prediction [21], also known as inductive conformal prediction, is a holdout method used to construct prediction intervals around a pre-trained model. Specifically, given a model $\mu: X \rightarrow \mathbb{R}$ fitted on an initial training data set, and n additional data points $(X_1, Y_1), \dots, (X_n, Y_n)$ (the holdout set), residuals are defined.

$$R_i = |Y_i - \hat{\mu}(X_i)|, i = 1, \dots, n, \tag{6}$$

and then compute the prediction interval at the new feature vector X_{n+1} as

$$\hat{C}_n(X_{n+1}) = \hat{\mu}(X_{n+1}) \pm (\text{the } [(1-\alpha)(n+1)]\text{-th smallest of } R_1, \dots, R_n). \tag{7}$$

Equivalently,

$$\hat{C}_n(X_{n+1}) = \hat{\mu}(X_{n+1}) \pm Q_{1-\alpha} \left(\sum_n^{i=1} \frac{1}{n+1} \cdot \delta_{R_i} + \frac{1}{n+1} \cdot \delta_{+\infty} \right). \tag{8}$$

where $Q_{\tau}(\cdot)$ denotes the τ -quantile of its argument, and δ_a denotes the point mass at a .

This method guarantees distribution-free predictive coverage at the target level $1 - \alpha$.

A drawback of the split conformal method is the loss of accuracy due to sample splitting, as the pretrained model μ must be independent of the holdout set. In practice, if only n labeled data points are available, $n/2$ points are used for training μ , and the procedure defined in equation (8) is run with a holdout set of size $n/2$ instead of n .

In this study, n will continue to denote the holdout set size for the split conformal method, to maintain consistent notation across different methods.

Full conformal prediction: An alternative to avoid the cost of data splitting is the full conformal method, also known as transductive conformal prediction.

$$\mathcal{A} : \cup_{n \geq 0} (\mathcal{X} \times \mathbb{R})^n \rightarrow \{\text{measurable functions } \hat{\mu} : \mathcal{X} \rightarrow \mathbb{R}\}. \tag{9}$$

Consider a regression algorithm: $n \geq 0 (\mathcal{X} \times \mathbb{R})^n \rightarrow \{\text{measurable functions } \mu : \mathcal{X} \rightarrow \mathbb{R}\}$, which maps a dataset containing pairs (X_i, Y_i) to a fitted regression function μ . The algorithm A must treat data points symmetrically, i.e.,

$$\mathcal{A}\left(\left(x_{\pi(1)}, y_{\pi(1)}\right), \dots, \left(x_{\pi(n)}, y_{\pi(n)}\right)\right) = \mathcal{A}\left(\left(x_1, y_1\right), \dots, \left(x_n, y_n\right)\right) \tag{10}$$

$\mathcal{A}\left(x_{\pi(1)}, y_{\pi(1)}\right), \dots, \left(x_{\pi(n)}, y_{\pi(n)}\right) = \mathcal{A}\left(x_1, y_1\right), \dots, \left(x_n, y_n\right)$ for all $n \geq 1$, all permutations π on $[n] := \{1, \dots, n\}$, and all $\{(x_i, y_i)\}_{i=1, \dots, n}$. For each $y \in \mathbb{R}$,

$$\hat{\mu}^y = \mathcal{A}\left(\left(X_1, Y_1\right), \dots, \left(X_n, Y_n\right), \left(X_{n+1}, y\right)\right). \tag{11}$$

let $\mu^y = \mathcal{A}\left(X_1, Y_1\right), \dots, \left(X_n, Y_n\right), \left(X_{n+1}, y\right)$ denote the trained model, fitted to the training data and a hypothesized test point (X_{n+1}, y) , and let

$$R_i^y = \begin{cases} \left|Y_i - \hat{\mu}^y(X_i)\right|, & i = 1, \dots, n, \\ \left|y - \hat{\mu}^y(X_{n+1})\right|, & i = n + 1. \end{cases} \tag{12}$$

The prediction set (which may or may not be an interval) for feature vector X_{n+1} is defined as

$$\hat{C}_n(X_{n+1}) = \left\{y \in \mathbb{R} : R_{n+1}^y \leq Q_{1-\alpha} \left(\sum_{i=1}^{n+1} \frac{1}{n+1} \cdot \delta_{R_i^y} \right)\right\}. \tag{13}$$

The full conformal method guarantees distribution-free predictive coverage at the target level $1 - \alpha$:

Theorem 1 (Full conformal prediction: If the data points $(X_1, Y_1), \dots, (X_n, Y_n), (X_{n+1}, Y_{n+1})$ are i.i.d. (or more generally, exchangeable), and the algorithm A treats the input data points symmetrically as in (10), then the full conformal prediction set defined in (13) satisfies.

$$\mathbb{P}\{Y_{n+1} \in \hat{C}_n(X_{n+1})\} \geq 1 - \alpha. \tag{14}$$

The same result holds true for split conformal.

In this study, we explore conformal prediction methods, specifically variants of split conformal prediction, also known as inductive conformal prediction. Next, we delineate the principal procedure of split conformal prediction for a general input x and output y .

Step 1. Split a dataset into two complementary subsets, the training fold D_{tr} and the calibration fold D_{cal} , with the calibration fold containing n samples.

Step 2. Use the training fold D_{tr} to train a deep learning model.

Step 3. Define a non-conformity score function $s(a, y)$ that quantifies how atypical a pair (a, y) .

Step 4. Calculate the quantile \hat{q} at the $\frac{(n+1)(1-\alpha)}{n}$ level for the non-conformity scores obtained from the calibration set D_{cal} .

Step 5. Use the quantile \hat{q} to generate the uncertainty intervals for predictions on a new instance, x_{test} :

$$\mathcal{C}(x_{\text{test}}) = \{y : s(x_{\text{test}}, y) \leq \hat{q}\} \tag{15}$$

The prediction set always satisfies a validity property for any score function and data distribution under the i.i.d condition. The coverage guarantee is formally provided next.

Theorem 2. Conformal coverage guarantee [22]. Suppose $(x_i, y_i)_{i=1, \dots, n}$ and $(x_{\text{test}}, y_{\text{test}})$ are i.i.d. and define \hat{q} as in step 4 above and $\mathcal{C}(X_{\text{test}})$ as in step 5 above. Then, the following statement holds.:

$$\mathbb{P}(y_{\text{test}} \in \mathcal{C}(x_{\text{test}})) \geq 1 - \alpha \tag{16}$$

Meanwhile, Theorem 1 also holds on the assumption of exchangeability [23]. The i.i.d. assumption in the above Theorem is stronger than the assumption of exchangeability that is practicable in the real world, Table 2 summarizes the notation used in this paper.

In summary, the bitcoin market often exhibits extreme price fluctuations, which can render traditional time series forecasting methods (such as ARIMA or conventional machine learning models) ineffective in predicting future prices, especially during sudden market events or sentiment shifts. Conformal forecasting addresses this challenge effectively. Conformal forecasting provides both a point forecast (such as a single price value) and a forecast interval (confidence interval). This interval is dynamically generated based on the distribution characteristics of both training and new data, reflecting the uncertainty of the forecast. In a highly volatile market like bitcoin, the forecast interval shows the range of possible price fluctuations, providing decision makers with more comprehensive information. Conformal forecasting does not rely on assumptions about the distribution of data. Even with high volatility and changing statistical properties over time, conformal forecasting adapts to market changes by dynamically adjusting the forecast interval and reducing the model’s sensitivity to extreme fluctuations.

Conformal prediction effectively handles non-stationarity. A key feature of conformal prediction is its independence from the assumption of data stationarity. Therefore, (1) it can be applied to non-stationary time series to learn and generate predictions directly from raw data. Despite the bitcoin market’s highly non-stationary characteristics, conformal forecasting provides meaningful forecast intervals to aid decision-making. (2) The conformal forecasting method avoids the complex step of stabilizing the data through a “non-parametric” method. It adjusts to the data at each moment, adapts to the market’s non-stationary characteristics, and responds to new trends.

2.4. ACI predictor

In this study, the ACI predictor is selected. The ACI predictor operates on the principles of conformal prediction theory in statistics. Conformal prediction, a powerful statistical tool,

Table 2. A summary of the symbols used in the equations shown in CP.

Notation	Definition
n	Size of the calibration set D_{cal} .
α	Confidence level parameter.
$\mathcal{C}(x_{\text{test}})$	Prediction set containing all candidate values y for the new instance x_{test} .
$s(x_{\text{test}}, y)$	Non-conformity score for the new instance x_{test} and candidate value y .
y_{test}	True value of the response variable for the new instance.
<i>i.i.d</i>	independent and identically distributed

<https://doi.org/10.1371/journal.pone.0319008.t002>

guarantees a certain coverage probability for predictions without assuming any specific data distribution.

Suppose we are given a fitted regression model for predicting the value of Y from X . Consider y as a candidate value for Y_t . To assess if \hat{y} is a reasonable estimate of Y_t , we define a conformity score $S(X, Y)$ to measure the extent to which y aligns with our model's predictions. For instance, if our model yields point predictions $\hat{\mu}(X_t)$, a conformity score could evaluate the distance between $\hat{\mu}(X_t)$ and y . An example of this is:

$$S(X_t, y) = |\hat{\mu}(X_t) - y|, \tag{17}$$

alternatively, let's assume our regression model produces estimates, $\hat{Q}(X; p)$, for the p^{th} quantile of the $Y | X$ distribution. We could then apply Conformal Quantile Regression (CQR), a method that evaluates the signed distance between y and the estimated upper and lower quantiles via a score.

$$S(X_t, y) = \max\{\hat{Q}(X_t; \alpha/2) - y, y - \hat{Q}(X_t; 1 - \alpha/2)\}. \tag{18}$$

Regardless of the chosen conformity score, the primary challenge is determining the threshold for $S(X_t, y)$ such that \hat{y} is considered a reasonable prediction for Y_t . Assume we have a calibration set $D_{cal} = \{(X_r, Y_r)\}_{r=1}^{t-1}$ that is different from the data that is used to fit the regression model. Using this calibration set, the conformity scores' fitted quantiles are defined as

$$\hat{q}(p) = \inf \left\{ s : \frac{1}{|D_{cal}|} \sum_{(x,y) \in D_{cal}} \mathbb{1}_{s(x,y) \leq s} \geq p \right\}. \tag{19}$$

A reasonable prediction for Y_t is y if $(X_t, y) \leq \hat{q}(1 - \alpha)$. The key observation is that if the data $D_{cal} \cup \{(X_t, Y_t)\}$ are exchangeable and ties are broken uniformly at random, the rank of $S(X_t, Y_t)$ among $\{S(X_n, Y_n)\}_{(X_n, Y_n) \in D_{cal}} \cup \{S(X_t, Y_t)\}$ will follow a uniform distribution. Therefore,

$$P(S(X_t, Y_t) \leq \hat{q}(1 - \alpha)) = \frac{|D_{cal}|(1 - \alpha) + 1}{|D_{cal}| + 1}, \tag{20}$$

thus, defining our prediction set as

$$\hat{C}_t = \{y : S(X_t, y) \leq \hat{q}(1 - \alpha)\}, \tag{21}$$

provides a marginal coverage guarantee.

$$P(Y_t \in \hat{C}_t) = P(S(X_t, Y_t) \leq \hat{q}(1 - \alpha)) = \frac{|D_{cal}|(1 - \alpha) + 1}{|D_{cal}| + 1}. \tag{22}$$

By introducing additional randomization this generic procedure can be altered slightly to produce C_t that satisfies the exact marginal coverage guarantee $P(Y_t \in \hat{C}_t) = 1 - \alpha$. Additionally, it is noteworthy that the described method is commonly known as split or inductive conformal inference. This refers to splitting the observed data into a training set for fitting the regression model and a separate calibration set, [Table 3](#) summarizes the notation used in this paper.

Table 3. A summary of the symbols used in the equations shown in ACI.

Notation	Definition
$S(X_t, y)$	Conformity score.
X_t	Predictor variable.
y	Candidate value for the response variable.
$\hat{\mu}(X_t)$	Model prediction value based on X_t (point prediction of the regression model).
$\hat{Q}(X_t; p)$	The p^{th} quantile of the distribution of Y given X_t .
α	Significance level parameter, usually set to 0.05.
$\hat{q}(p)$	Combined quantile.
inf	Infimum.
s	Candidate value in the range.
$ D_{cal} $	Size of the calibration set $ D_{cal} $.
(x, y)	Data pair in the calibration set.
$\hat{q}(1-\alpha)$	Quantile of $1-\alpha$.

<https://doi.org/10.1371/journal.pone.0319008.t003>

3. Evaluation criteria

This study employs the quantile loss function, RMSE, MAE and Mean Absolute Percentage Error (MAPE) as evaluation metrics. The quantile loss function is applied in quantile regression. Unlike traditional loss functions, the quantile loss function imposes varying penalties on predictions for low and high values, enabling the model to more effectively capture different aspects of the data distribution. The quantile loss function employs different τ values to mitigate overfitting and underfitting, thereby enabling effective quantile regression. y represents the true value of each observation. \hat{y} denotes the predicted value of the model corresponding to each observation. Table 4 provides a summary of the symbols used in this paper.

$$J_{\text{quant}} = \frac{1}{N} \sum_{i=1}^N \left[\tau |y_i - \hat{y}_i| \mathbb{1}_{y_i \geq \hat{y}_i} + (1-\tau) |y_i - \hat{y}_i| \mathbb{1}_{y_i < \hat{y}_i} \right], \tag{23}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \tag{24}$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \tag{25}$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%. \tag{26}$$

When $\hat{y}_i \geq y_i$ (the true value is greater than or equal to the predicted value), the loss function is $\tau \cdot (y_i - \hat{y}_i)$. This implies that the loss is weighted by the value of τ if the model underestimates the true value. Where, τ serves as a weight that amplifies the error when predicted values are low. Correspondingly, when $\hat{y}_i < y_i$ (the true value is smaller than the predicted value), the loss function is $(1-\tau) \cdot (\hat{y}_i - y_i)$, This implies that the loss is weighted by the

Table 4. A summary of the symbols used in the equations shown in quantile loss function.

Notation	Definition
J_{quant}	Quantile loss, which is used to evaluate the model's prediction error in quantile regression.
N	Total number of observations.
y_i	True value of the response variable for the i^{th} observation.
τ	Quantile level, a parameter that determines the weight of the loss function.
$\hat{y}_i \geq y_i$	The loss function is $\tau \cdot y_i - \hat{y}_i $, which means that when the model underestimates the true value, the loss is weighted by τ .
$\hat{y}_i < y_i$	The loss function is $(1 - \tau) \cdot y_i - \hat{y}_i $, which means that when the model overestimates the true value, the loss is weighted by $1 - \tau$.

<https://doi.org/10.1371/journal.pone.0319008.t004>

value of $(1 - \tau)$ if the model overestimates the true value. Where, $(1 - \tau)$ serves as a weight that amplifies the error when predicted values are high.

4. Dataset

In this study, we used bitcoin prices from 2018 to 2024 (Fig 1). Additionally, we utilized bitcoin blockchain information and various statistics [5]. We considered 29 features of the bitcoin blockchain among various available features (as shown in Table 5).

To identify the most useful features among those listed in Table 5, a correlation analysis with bitcoin price is performed on the remaining features. Specifically, we calculated Spearman's rank correlation coefficients and Spearman rank correlation coefficient matrix between bitcoin blockchain features, as shown in Table 6 and Fig 4. Spearman's rank correlation coefficient is a non-parametric measure used to evaluate the monotonic relationship between two variables. The Spearman correlation between two variables is similar to the Pearson correlation, representing a linear correlation between the ranked values of the two variables. A positive Spearman correlation coefficient between two variables X and Y indicates that when X increases (or decreases), Y also tends to increase (or decrease). Conversely, a negative coefficient indicates that as X increases (or decreases), Y tends to decrease (or increase). A Spearman correlation of zero indicates no correlation between X and Y .

In this study, we excluded features with correlation coefficients less than 0.75, such as cost-per-trans-pct, est-trans-vol, med-cfm-time, output-vol, and transfees, due to their weak relationship with bitcoin price. Features with correlation coefficients greater than 0.95, such as market capitalization, market price, and miner income, are also excluded. In fact, market cap and market price trends are almost identical to bitcoin's price. Note that market price is the average bitcoin USD market price across major bitcoin exchanges, and market cap is defined as the market price multiplied by the supply of bitcoin in circulation. The trends of some features, which are either highly correlated or not correlated with the bitcoin price, are shown in Fig 5. For example, the trend of trade-vol, shown in Fig 5b with a Spearman correlation of 0.8953, is very similar to that of the bitcoin price shown in Fig 5a. In contrast, the trend of est-trans-vol, shown in Fig 5c with a Spearman correlation of -0.0565 , is not related to the bitcoin price. In total, 18 features, including the bitcoin price, are used to develop prediction models and predict the next bitcoin price. Among the 18 features, the ranges of the values of difficulty, est-trans-vol-usd, hash-rate, my-wallets, trade-vol, and trans-fees-usd are very large. Therefore, the log values of these features are used in the experiment.

Table 5. Bitcoin blockchain features.

Feature	Description
avg-block-size	The 24 h average block size in MB.
blockchain-size	The total size of all block headers and transactions.
cost-per-trans	Miners revenue divided by the number of transactions.
cost-per-trans-pct	Miners revenue as percentage of the transaction volume.
difficulty	A relative measure of difficulty in finding a new block.
est-trans-vol	The estimated value of transactions on the bitcoin blockchain in BTC.
est-trans-vol-usd	The estimated USD value of transactions.
hash-rate	The estimated number of tera hashes per second the bitcoin network is performing.
market-cap	The total USD value of bitcoin supply in circulation.
market-price	The average USD market price across major bitcoin exchanges.
med-cfm-time	The median time for a transaction to be accepted into a mined block.
mempool-count	The number of transactions waiting to be confirmed.
mempool-growth	The rate of the memory pool (mempool) growth per second.
mempool-size	The aggregate size of transactions waiting to be confirmed.
miners-revenue	The total value of Coinbase block rewards and transaction fees paid to miners.
my-wallets	The total number of blockchain wallets created.
n-trans	The number of daily confirmed bitcoin transactions.
n-trans-excl-100	The total number of transactions per day excluding the chains longer than 100.
n-trans-excl-popular	The total number of transactions, excluding those involving any of the network's 100 most popular addresses.
n-trans-per-block	The average number of transactions per block.
n-trans-total	The total number of transactions.
n-unique-addr	The total number of unique addresses used on the bitcoin blockchain.
output-val	The total value of all transaction outputs per day.
total-bitcoins	The total number of bitcoins that have already been mined.
trade-vol	The total USD value of trading volume on major bitcoin exchanges.
trans-fees	The total BTC value of all transaction fees paid to miners.
trans-fees-usd	The total USD value of all transaction fees paid to miners.
trans-per-sec	The number of bitcoin transactions added to the mempool per second.
utxo-count	The number of unspent bitcoin transactions outputs.

<https://doi.org/10.1371/journal.pone.0319008.t005>

5. Experiment analysis

5.1. Experience and results

All experiments in this article are conducted on a single computer, Win10 64-bit operating system, Intel(R) Core (TM) i7-9750H CPU @2.60 GHz 2.59 GHz, NVIDIA GeForce RTX 2060 GPU and 16GB RAM HP laptop on Run the generated program. Data processing and enhancement via Python. The LSTM model and ARIMA model is implemented using Python3.9 and the Keras deep learning library, and the conformal prediction model is implemented using the Torchcp library. The parameter settings used in the experiments are shown in [Table 7](#).

Initially, the input sequence size m is determined through experimental analysis. [Table 8](#) illustrates the influence of m on the regression performance of the LSTM model in relation to MAPE. Optimal values are denoted by underlined numbers for a given window size, while those in bold signify optimal values for a specific prediction model. Notably, all prediction models achieve superior performance when $m = 5$, as lower MAPE values

Table 6. Spearman correlation coefficients between bitcoin prices and various features.

Bitcoin feature	Coefficient
avg-block-size	0.8479
blockchain-size	0.9017
cost-per-trans	0.7568
cost-per-trans-pct	-0.4734
difficulty	0.8967
est-trans-vol	-0.0565
est-trans-vol-usd	0.9397
hash-rate	0.9021
market-cap	0.9876
market-price	0.9915
med-cfm-time	0.0878
miners-revenue	0.9713
my-wallets	0.8998
n-trans	0.8162
n-trans-excl-100	0.8469
n-trans-excl-popular	0.8233
n-trans-per-block	0.8070
n-trans-total	0.9030
n-unique-addr	0.8655
output-vol	0.1428
total-bitcoins	0.9019
trade-vol	0.8953
trans-fees	0.4297
trans-fees-usd	0.9310
utxo-count	0.9013

<https://doi.org/10.1371/journal.pone.0319008.t006>

indicate better prediction accuracy. With a smaller input series size, prediction models tend to forecast prices closely aligned with the previous day, thereby minimizing MAPE. However, solely focusing on the changes indicated by the MAPE metric does not fully capture the model's performance. To provide a more comprehensive evaluation of the model's performance, [Figs 6](#) and [7](#) display the MAE and RMSE metrics for different batches. Consequently, selecting a smaller input sequence size is advisable. To provide a more intuitive presentation of the results, the data from 2023-2024 are partially visualized, the results are shown in [Fig 8](#).

However, a natural question arises. To test the generalization of CP, does directly embedding it into existing models also improve their reliability? Therefore, [Figs 9](#) and [10](#) display the performance of other models after embedding the CP module. Clearly, the performance of all models has improved to varying extents after embedding the CP module.

The results of the conformal prediction model are depicted in [Fig 11](#). Black dots illustrate the actual sample values in the test dataset, which the predictive model aims to forecast accurately. Each blue vertical line represents a prediction interval, signifying the model's potential prediction range for a sample. Prediction intervals typically capture model uncertainty arising from data noise, imperfections, or limitations in the training sample. The abscissa shows the test dataset sample index, linking each actual value and prediction interval with its corresponding test sample. On the ordinate, actual values (depicted by black dots) represent each sample's measured value in the test dataset. For the prediction

Spearman rank correlation coefficient matrix between Bitcoin blockchain features.

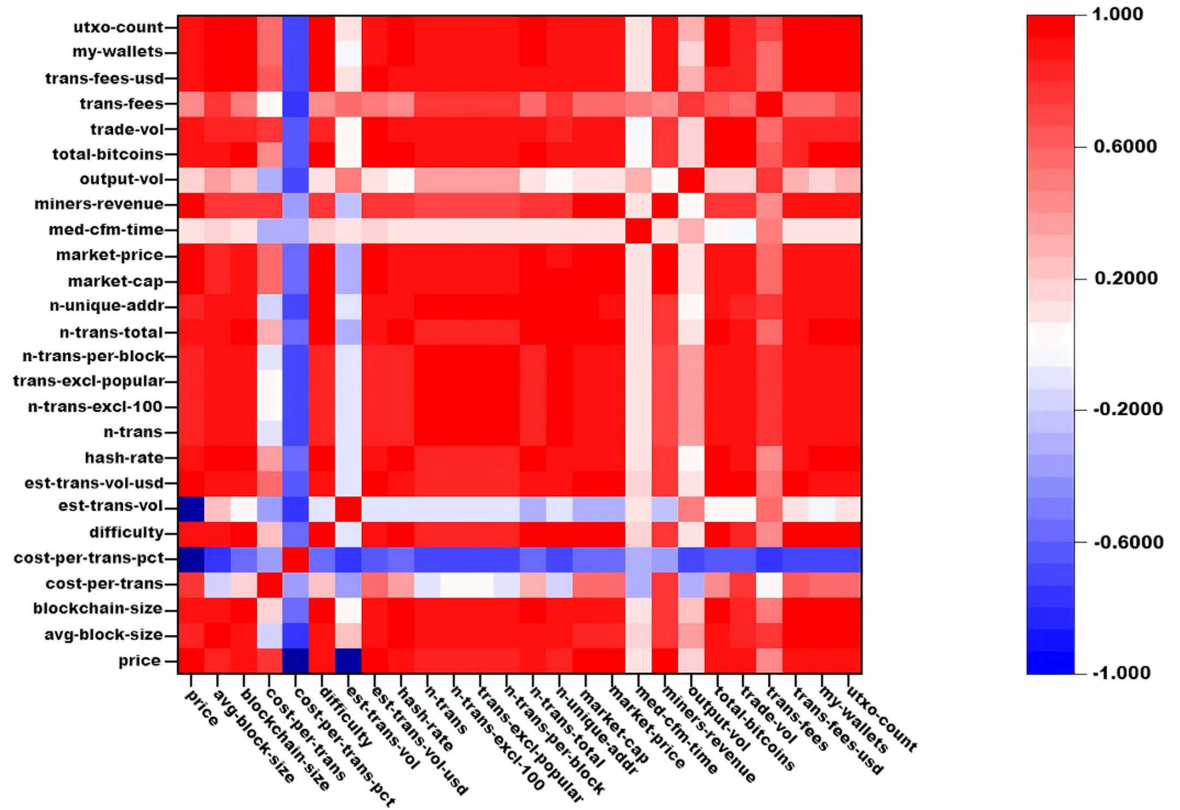


Fig 4. Spearman rank correlation coefficient matrix between bitcoin blockchain features.

<https://doi.org/10.1371/journal.pone.0319008.g004>

interval (blue vertical line), the ordinate's upper and lower ends indicate the upper and lower bounds of the prediction, respectively. As shown in Fig 11, approximately 95.2% of the actual values fall within the prediction interval, indicating the model's prediction interval offers a reasonably accurate range of uncertainty. This underscores our high confidence in the results of the LSTM model.

First, when the prediction interval is too wide, the associated uncertainty is also large, which may hinder effective decision-making. For high-volatility assets, an excessively wide prediction interval may lead to overly cautious risk management strategies and increased exposure to potential risks. For instance, if investors set stop-loss and take-profit points based on the width of the prediction interval, an overly wide interval may result in stop-loss and take-profit points being set too far apart, causing investors to lose control during sharp market fluctuations, potentially amplifying losses. Additionally, rapid market changes require immediate responses, and excessively wide prediction intervals may make decisions seem slow and overly conservative, disregarding the market's rapid adjustments.

When the prediction interval is too narrow, it indicates that the model's prediction of future prices is highly accurate, but this may not reflect the true uncertainty. Narrow prediction intervals may lead to overconfidence among decision makers and an underestimation of market risks. In investment decisions, excessive reliance on narrow predictions may prompt overly aggressive actions and the neglect of potential market fluctuations. Narrow predictions may sometimes result from the model's overfitting of historical data. Overfitting often



Fig 5. Changes in the values of selected features. (a) Bitcoin price. (b) Trade-vol. (c) Est-trans-vol. (d) Cost-per-trans.

<https://doi.org/10.1371/journal.pone.0319008.g005>

Table 7. The parameter settings used in the experiments.

Module	Parameter settings			
LSTM	Layers of LSTM	Hidden size of LSTM	Learning rate	Batch size
	2	64	0.001	16
CP	Confidence level	Calibration set size		
	95%	70% Training set 30% Calibration set		

<https://doi.org/10.1371/journal.pone.0319008.t007>

Table 8. The impact of sequence size m on regression (expressed as MAPE, %). Logarithmic values of the primary features, sequential partitioning, and normalization based on the first value are applied.

Size (m)	MAPE
5	3.68
10	4.01
20	4.51
50	6.94
100	28.09

<https://doi.org/10.1371/journal.pone.0319008.t008>

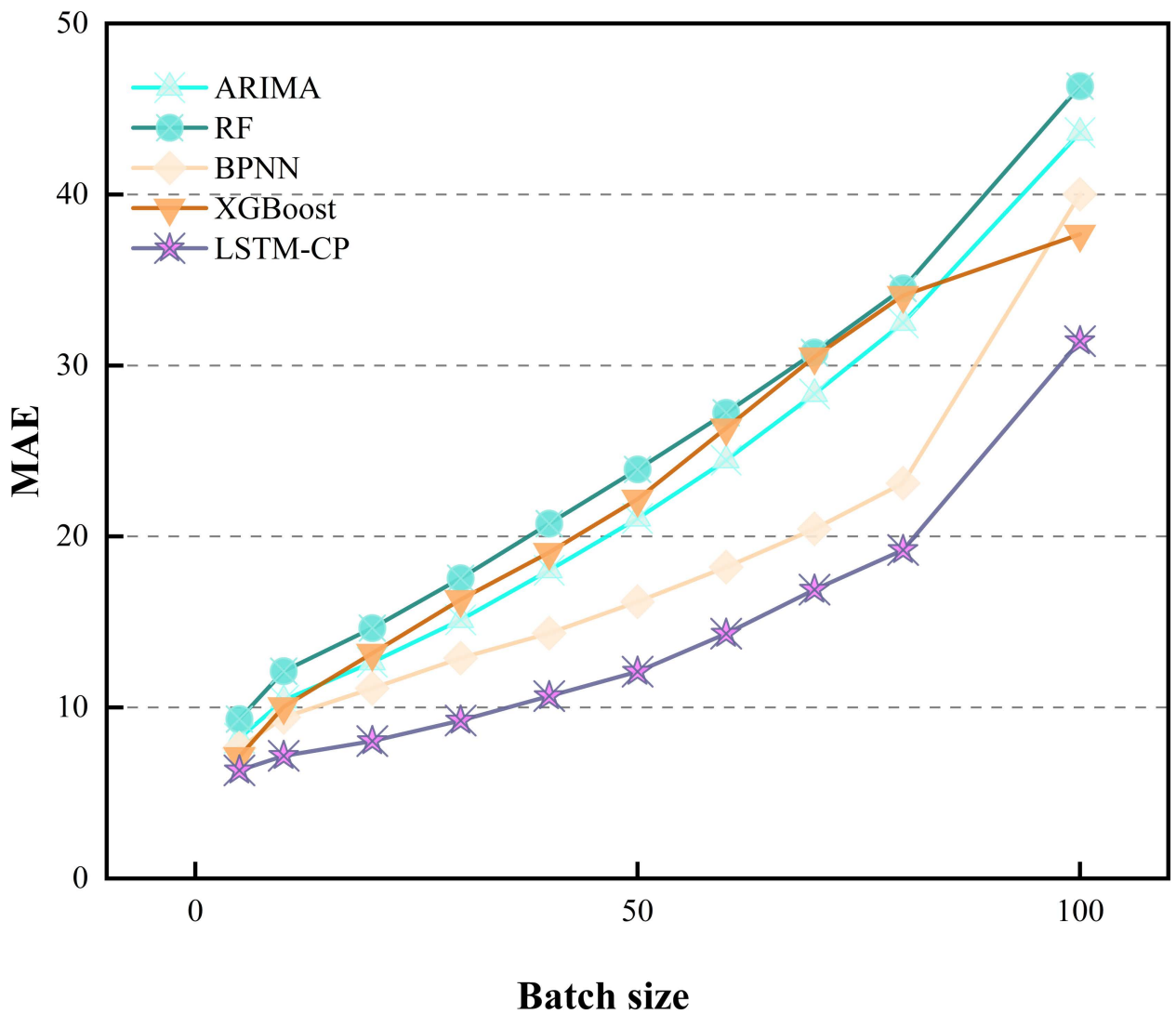


Fig 6. MAE metrics for different models at varying input sizes.

<https://doi.org/10.1371/journal.pone.0319008.g006>

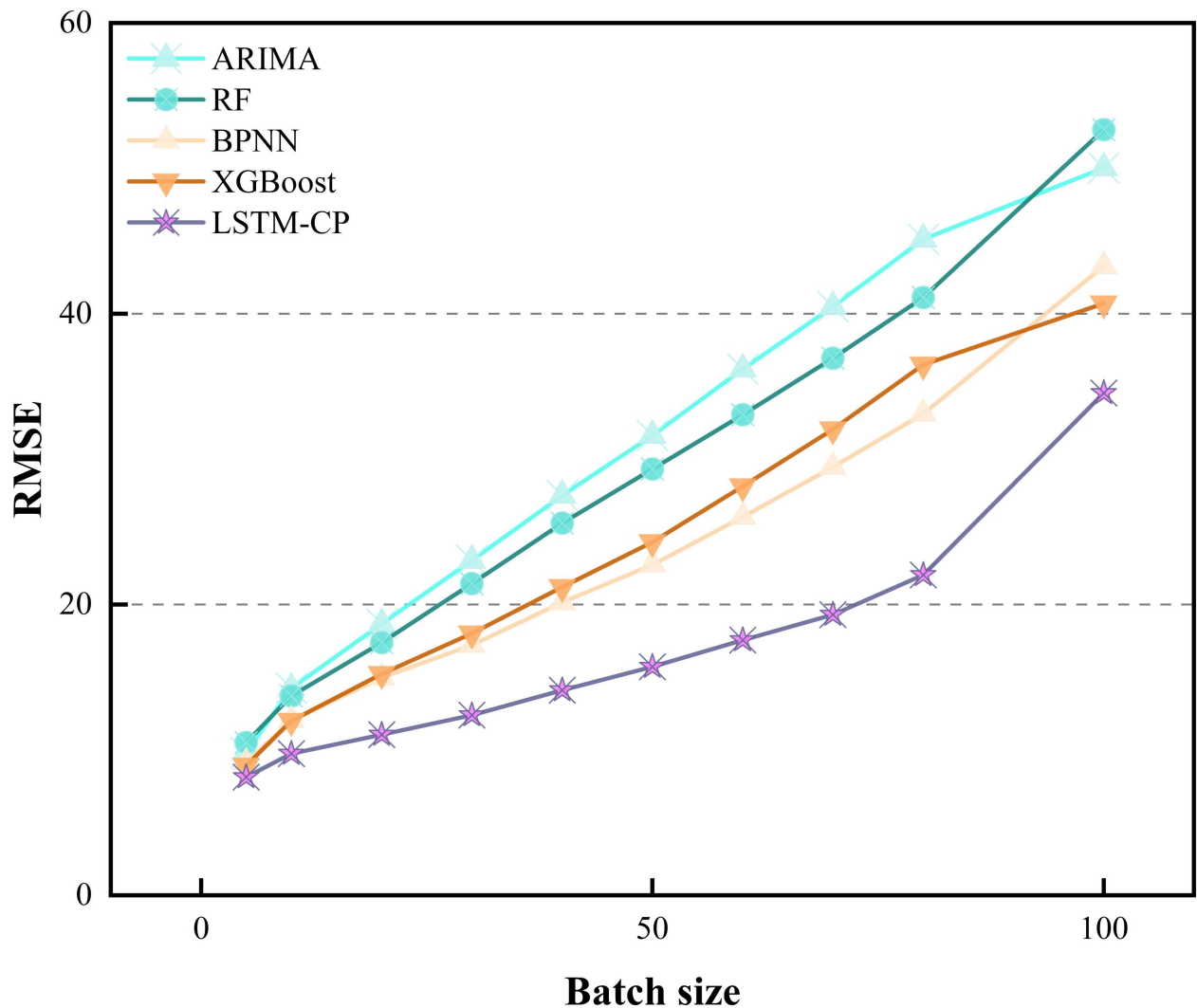


Fig 7. RMSE metrics for different models at varying input sizes.

<https://doi.org/10.1371/journal.pone.0319008.g007>

causes the model to become overly dependent on past market patterns, ignoring potential future changes.

In practice, while a wide range offers a higher confidence level, it may cause investors to miss quick market opportunities or increase risk exposure due to over-conservatism. In this case, there is a significant conflict between the width of the range and the urgency of the decision. Conversely, while a narrow range may improve decision-making accuracy, it may lead investors to overlook the true volatility and potential risks of the market. Particularly in highly volatile markets, a narrow range quickly becomes ineffective.

5.2. Discussion

A novel method that combines conformal prediction with LSTM models is proposed in this study, providing multiple improvements for time series forecasting. Specifically, LSTM models address the constraints of other predictive models by capturing the intricate and often non-linear dynamics of bitcoin market behavior, enabling real-time predictions in the

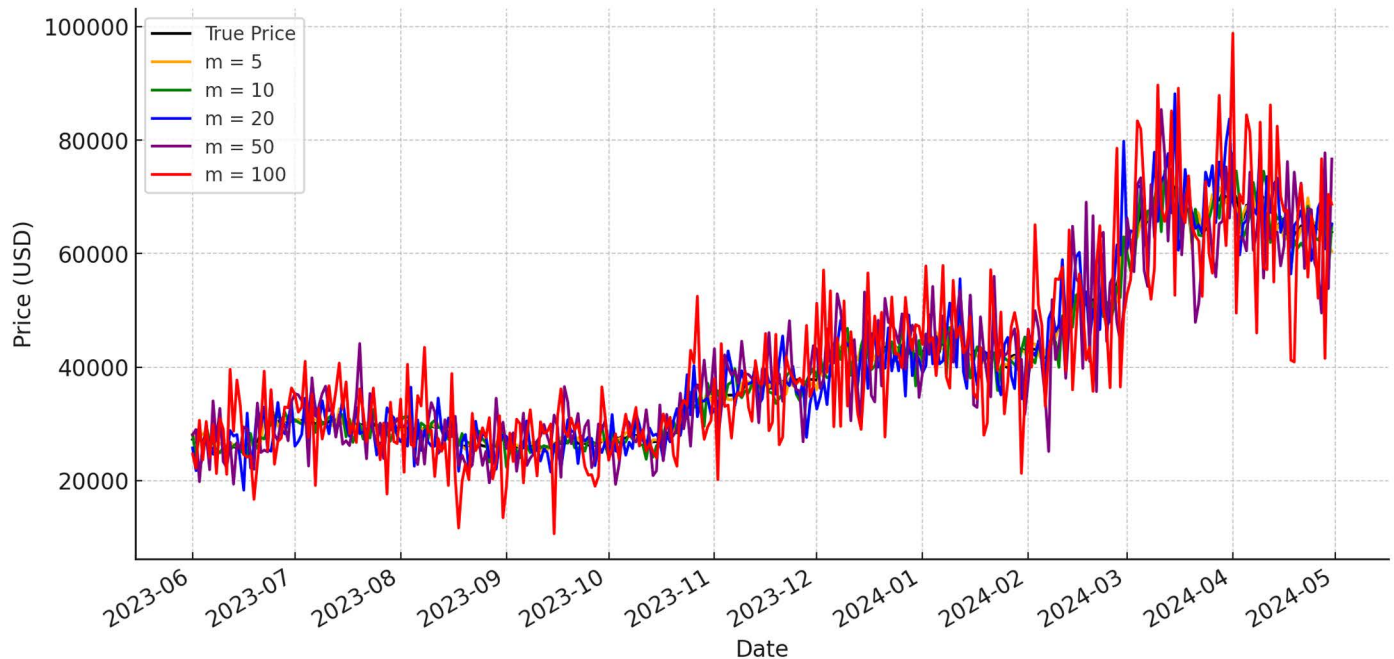


Fig 8. Comparison of predicted values across varying input sequence sizes.

<https://doi.org/10.1371/journal.pone.0319008.g008>

volatile bitcoin market environment. LSTM enhances the model's ability to generalize to unseen data. Moreover, incorporating confidence intervals through conformal prediction adds a probabilistic dimension that enhances interpretability by presenting a range of predicted values. This addition is particularly advantageous for decision-makers across various domains, especially in the intricate realm of bitcoin market forecasting, as it enables the assessment of prediction reliability for informed strategic adjustments. This advancement signifies a crucial stride toward enhancing the decision-making process through meticulous and informed approaches.

Despite its numerous advantages, this approach also faces inevitable limitations. The complexity of this approach demands substantial computational resources and prolonged training periods, posing implementation challenges, particularly when resources are limited or immediate results are desired. Additionally, the accuracy of confidence intervals generated by conformal predictions heavily relies on the representativeness of the calibration dataset. Inadequate reflection of future data distributions in the dataset can severely impact the reliability of confidence intervals, underscoring the significance of high-quality, representative data. Furthermore, noisy data or suboptimal predictions from the LSTM model may lead to overly wide confidence intervals, reducing the precision of confidence measures and emphasizing the need to balance interval width with information value. Currently, there exists a Denoising Diffusion Probabilistic Model (DDPM) [24], which is a class of generative models optimized through a two-step process: injecting noise into the training samples and subsequently removing it. This process consists of a forward phase (diffusion) and a reverse phase (denoising). By training the model to remove the noise introduced during the diffusion process, valid data samples closely aligned with the training data distribution are generated in the inference phase, making it highly suitable for modeling time series patterns and for downstream tasks. Integrating these advanced techniques into the existing LSTM-conformal

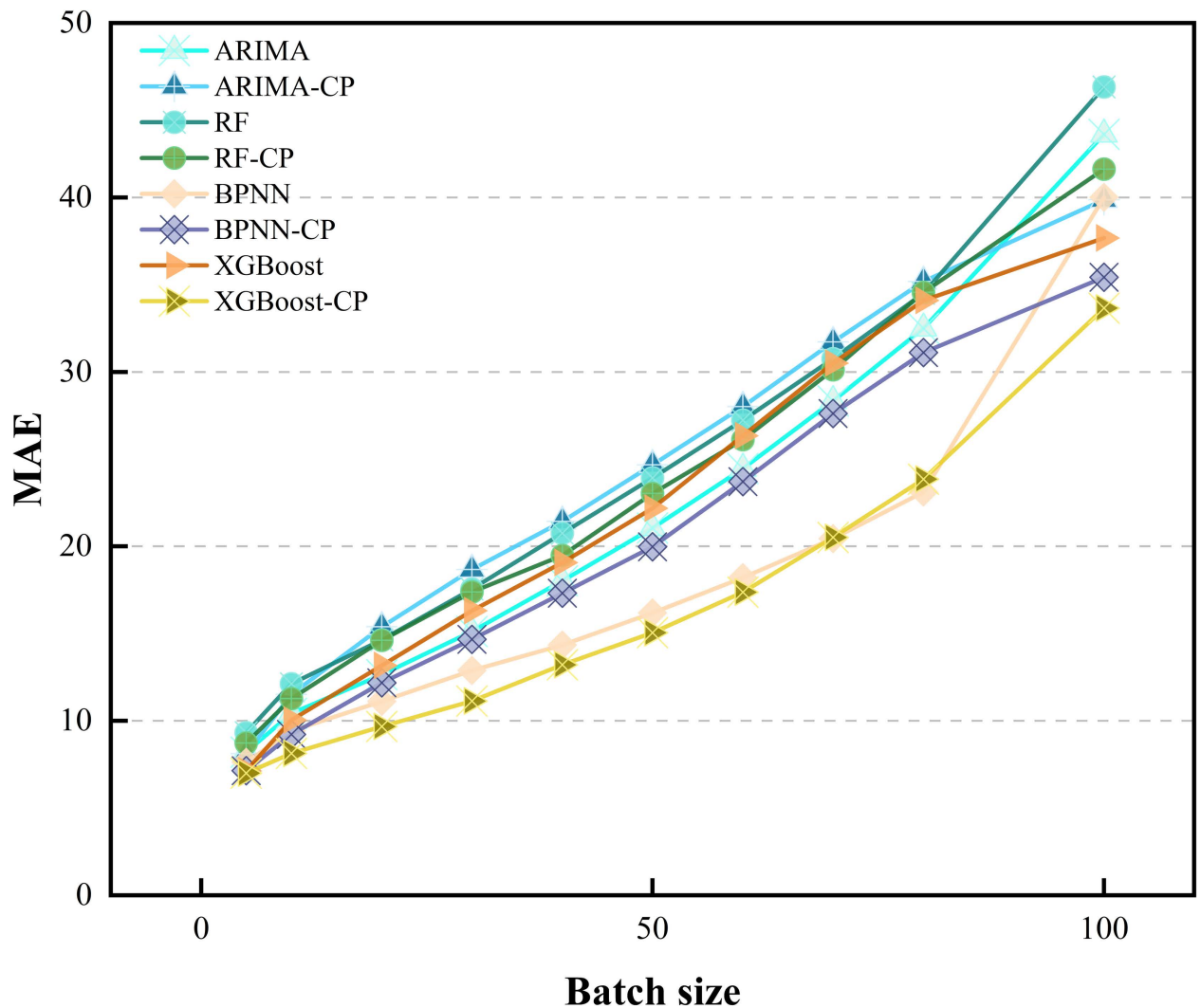


Fig 9. MAE after embedding the CP module into different models.

<https://doi.org/10.1371/journal.pone.0319008.g009>

prediction model is expected to significantly enhance prediction methods in economic scenarios.

In conclusion, while this novel approach is poised to notably enhance the accuracy and reliability of time series forecasting, it also presents challenges necessitating strategic considerations and mitigation efforts to fully leverage its benefits.

6. Conclusion

This study presents a novel approach in which LSTM models are integrated with conformal prediction techniques, tailored for bitcoin market prediction. This method enhances the credibility of forecasted values, addressing the primary challenge of accurately predicting market trends. The initial phase of our methodology involves selecting and preprocessing key features extracted from the dataset, including the average block size, total blockchain size, and transaction-related metrics such as average transaction size and transaction cost percentages.

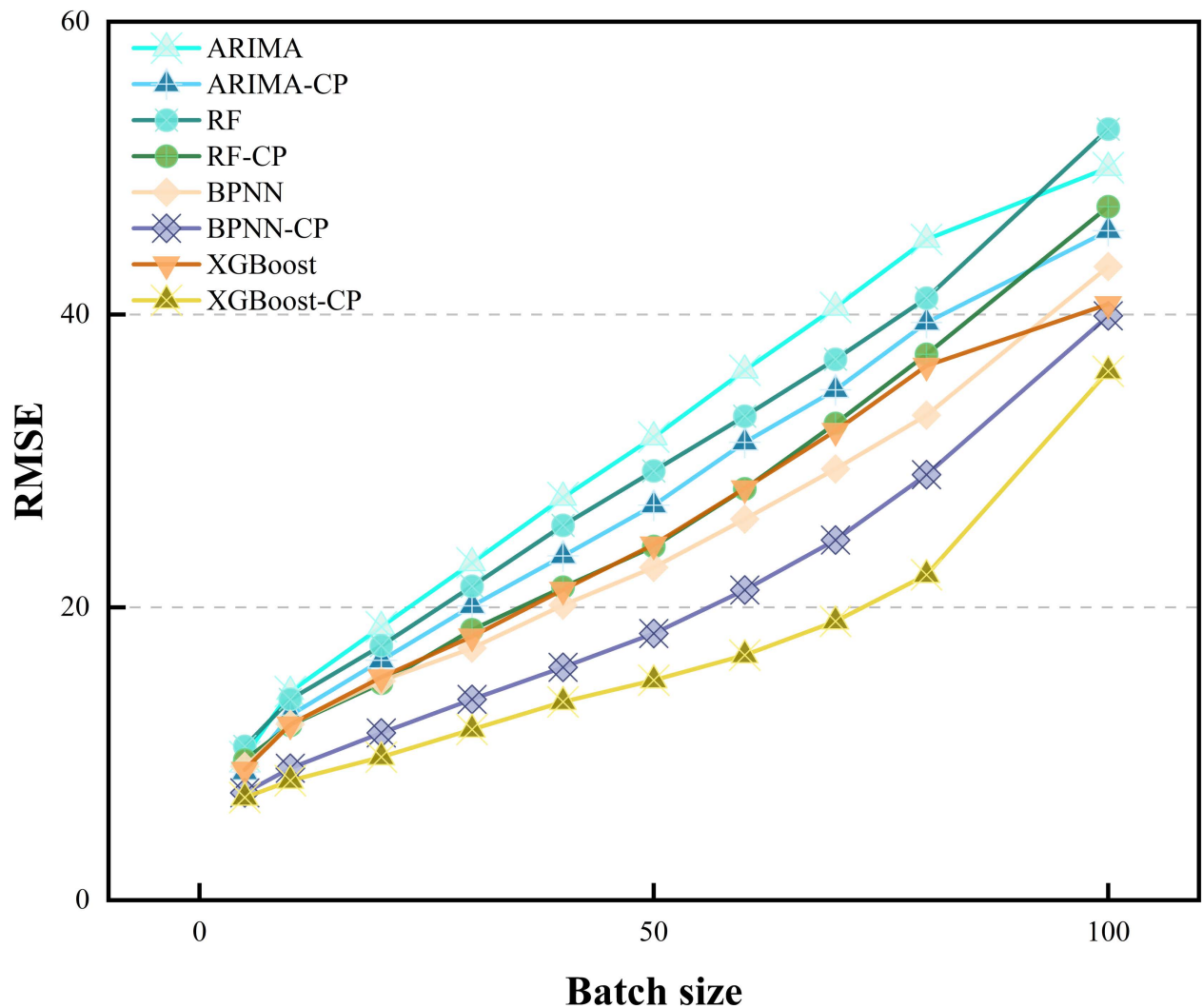


Fig 10. RMSE after embedding the CP module into different models.

<https://doi.org/10.1371/journal.pone.0319008.g010>

Features are selected based on Spearman correlation coefficients, excluding those with values below 0.75 or above 0.95.

The key aspect of the method is the intricate processing of preprocessed data using an LSTM network in sequential time steps. After generating predictions, backpropagation is employed to refine the final output using a loss function and optimizer, ensuring prediction accuracy. Subsequently, preliminary predictions from the LSTM model are incorporated into the conformal prediction model.

In future research, the challenge of managing excessively wide confidence intervals, often caused by significant dataset noise and dependency issues, will be addressed. The forecasting framework will be refined to improve the efficiency and credibility of market sales forecasts, contributing to the knowledge base and advancing predictive analytics in market trend forecasting. Advanced data preprocessing techniques will be explored, LSTM model parameters will be optimized, and other predictive models may be integrated to improve the robustness and accuracy of the methods.

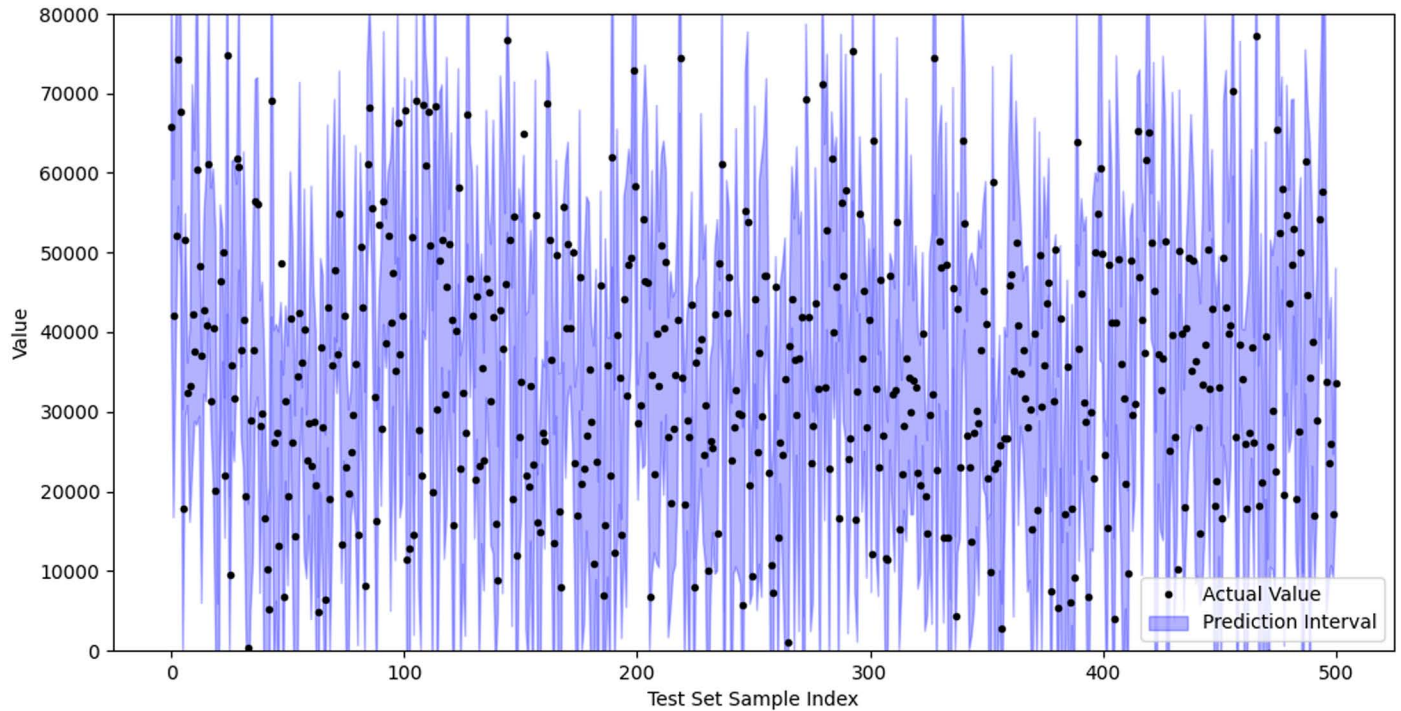


Fig 11. Prediction intervals and actual values.

<https://doi.org/10.1371/journal.pone.0319008.g011>

Author contributions

Conceptualization: Xiangyue Zhang.

Data curation: Wenjing Wang, Keqing Wang.

Formal analysis: Yuyun Kang.

Funding acquisition: Chao Li.

Investigation: Xiangyue Zhang.

Methodology: Xiangyue Zhang.

Project administration: Xiangyue Zhang, Yuyun Kang.

Resources: Wenjing Wang, Keqing Wang.

Software: Xiangyue Zhang.

Supervision: Yuyun Kang, Wenjing Wang.

Validation: Xiangyue Zhang, Wenjing Wang.

Visualization: Xiangyue Zhang.

Writing – original draft: Yuyun Kang.

Writing – review & editing: Xiangyue Zhang.

References

1. Sin E, Wang L. Bitcoin price prediction using ensembles of neural networks. 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD). IEEE; 2017. p. 1–10.
2. Struga K, Qirici O. Bitcoin price prediction with neural networks. RTA-CSIT; 2018. p. 15–23.

3. Azari A. Bitcoin price prediction: an ARIMA approach. arXiv:1904.05315 [Preprint]. 2019;1904(05315):1–10. <https://doi.org/10.48550/arXiv.1904.05315>
4. Poongodi M, Vijayakumar V, Chilamkurti N. Bitcoin price prediction using ARIMA model. *Int J Internet Technol Secur Trans*. 2020;10(4):396–406. <https://doi.org/10.1504/ijitst.2020.10028982>
5. Kang Y. Data for: Bitcoin. QDR Main Collection; 2025. <https://doi.org/10.5064/F6AE7MKW>
6. Liu M, Li G, Li J, Zhu X, Yao Y. Forecasting the price of bitcoin using deep learning. *Fin Res Lett*. 2021;40:101755. <https://doi.org/10.1016/j.frl.2020.101755>
7. Gao L, Ye W, Guo R. Jointly forecasting the value-at-risk and expected shortfall of bitcoin with a regime-switching CAViaR model. *Fin Res Lett*. 2022;48:102826. <https://doi.org/10.1016/j.frl.2022.102826>
8. Chen J. Analysis of bitcoin price prediction using machine learning. *J Risk Financial Manag*. 2023;16(1):51. <https://doi.org/10.3390/jrfm16010051>
9. Ranjan S, Kayal P, Saraf M. Bitcoin price prediction: a machine learning sample dimension approach. *Comput Econ*. 2023;61(4):1617–36. <https://doi.org/10.1007/s10614-022-10262-6>
10. Koo E, Kim G. Centralized decomposition approach in LSTM for bitcoin price prediction. *Expert Syst Appl*. 2024;237:121401. <https://doi.org/10.1016/j.eswa.2023.121401>
11. Vo A-D, Nguyen Q-P, Ock C-Y. Sentiment analysis of news for effective cryptocurrency price prediction. *Int J Knowl Eng*. 2019;5(2):47–52. <https://doi.org/10.18178/ijke.2019.5.2.116>
12. Wolk K. Advanced social media sentiment analysis for short-term cryptocurrency price prediction. *Expert Syst*. 2020;37(2):e12493. <https://doi.org/10.1111/exsy.12493>
13. McNally S, Roche J, Caton S. Predicting the price of bitcoin using machine learning. 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE; 2018. p. 135–41.
14. Dutta A, Kumar S, Basu M. A gated recurrent unit approach to bitcoin price prediction. *J Risk Financial Manag*. 2020;13(2):23. <https://doi.org/10.3390/jrfm13020023>
15. Hafid A, Senhaji Hafid A, Makrakis D. Bitcoin price prediction using machine learning and technical indicators. *International Symposium on Distributed Computing and Artificial Intelligence*. Cham: Springer Nature Switzerland; 2023. p. 123–32.
16. Awoke T, Rout M, Satapathy S, Mohanty L. Bitcoin price prediction and analysis using deep learning models. *Communication Software and Networks: Proceedings of INDIA 2019*. Singapore: Springer Singapore; 2020. p. 631–40.
17. Jiang X. Bitcoin price prediction based on deep learning methods. *J Math Finance*. 2019;10(01):132–9. <https://doi.org/10.4236/jmf.2020.101009>
18. Hashish IA, Forni F, Andreotti G, Facchinetti T, Darjani S. A hybrid model for bitcoin prices prediction using hidden Markov models and optimized LSTM networks. 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE; 2019. p. 1–8.
19. Gao Z, He Y, Kuruoglu EE. A hybrid model integrating LSTM and GARCH for bitcoin price prediction. 31st IEEE International Workshop on Machine Learning for Signal Processing (MLSP). IEEE; 2021. p. 1–6.
20. Balasubramanian V, Ho SS, Vovk V. Conformal prediction for reliable machine learning: theory, adaptations and applications. Newnes; 2014. p. 1–350.
21. Guan L. Localized conformal prediction: a generalized inference framework for conformal prediction. *Biometrika*. 2023;110(1):33–50. <https://doi.org/10.1093/biomet/asac040>
22. Papadopoulos H. Guaranteed coverage prediction intervals with Gaussian process regression. arXiv:2310.15641 [Preprint]. 2023;46(12):9072–83. <https://doi.org/10.1109/tpami.2024.3418214>
23. Lei J, G'Sell M, Rinaldo A, Tibshirani RJ, Wasserman L. Distribution-free predictive inference for regression. *Int J Prod Res*. 2018;113(523):1094–111. <https://doi.org/10.1080/01621459.2017.1307116>
24. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Adv Neural Inf Process Syst*. 2020;33:6840–51.