Computational Economics (2023) 62:1919–1945
https://doi.org/10.1007/s10614-022-10325-8

Modeling Bitcoin Prices using Signal Processing Methods, Bayesian Optimization, and Deep Neural Networks

Bhaskar Tripathi[1] · **Rakesh Kumar Sharma**[1]

Accepted: 20 September 2022 / Published online: 28 October 2022

Abstract

Bitcoin is a volatile financial asset that runs on a decentralized peer-to-peer Blockchain network. Investors need accurate price forecasts to minimize losses and maximize profits. Extreme volatility, speculative nature, and dependence on intrinsic and external factors make Bitcoin price forecast challenging. This research proposes a reliable forecasting framework by reducing the inherent noise in Bitcoin time series and by examining the predictive power of three distinct types of predictors, namely fundamental indicators, technical indicators, and univariate lagged prices. We begin with a three-step hybrid feature selection procedure to identify the variables with the highest predictive ability, then use Hampel and Savitzky–Golay filters to impute outliers and remove signal noise from the Bitcoin time series. Next, we use several deep neural networks tuned by Bayesian Optimization to forecast short-term prices for the next day, three days, five days, and seven days ahead intervals. We found that the Deep Artificial Neural Network model created using technical indicators as input data outperformed other benchmark models like Long Short Term Memory, Bi-directional LSTM (BiLSTM), and Convolutional Neural Network (CNN)-BiLSTM. The presented results record a high accuracy and outperform all existing models available in the past literature with an absolute percentage error as low as 0.28% for the next day forecast and 2.25% for the seventh day for the latest out of sample period ranging from Jan 1, 2021, to Nov 1, 2021. With contributions in feature selection, data-preprocessing, and hybridizing deep learning models, this work contributes to researchers and traders in fundamental and technical domains.

Keywords Time series forecasting · Deep learning · Bayesian optimization · Savitzky–Golay Filter · Outlier detection

✉ Rakesh Kumar Sharma
rakesh.kumar@thapar.edu

Bhaskar Tripathi
btripathi_phdp16@thapar.edu

[1] School of Humanities and Social Sciences, Thapar Institute of Engineering and Technology, Patiala 147004, India

JEL Classification C32 · G17 · C45 · C61 · E27

1 Introduction

Bitcoin is a peer-to-peer virtual currency that runs on Blockchain technology. It was introduced by Satoshi Nakamoto in October 2008 to solve the problem of double-spending without the need of any central authority (Nakamoto, 2008). Since 2013, Bitcoin has grown in popularity as a speculative asset. At present, Bitcoin is the most dominant among all other cryptocurrencies, which collectively reached a market cap of over 3 trillion US Dollars in the peak of 2021. However, throughout its history, Bitcoin has been an extremely volatile asset. According to https://coinmarketcap.com, accessed on November 6th, 2021, Bitcoin recently surpassed an all-time high of $64,000 before falling over 30% in less than 24 h. These irregular price fluctuations expose traders and investors to risks that may result in severe financial penalties. Simultaneously, it also poses a challenging task for researchers to devise new price prediction approaches to deal with such volatility and accurately predict the prices. Another problem is the changing nature of the market. The cryptocurrency market has evolved through several market cycles and phases of bull-bear runs, public attention, technological advancements in Blockchain, speculation, and adoption, among other natural processions. As market phases evolve, the pricing factors that influenced one stage of the market no longer apply to the subsequent phase, creating anxiety among investors looking to buy the dip or sell at a profit.

In past, various studies have forecasted Bitcoin prices using econometric, statistical, and machine learning models. The preliminary studies mainly focused on applying statistical and econometric methods to understand better the factors influencing Bitcoin's price development and fundamental characteristics (Cheung et al., 2015; Ji et al., 2018; Kristoufek, 2015; Szetela et al., 2016). More recently, the focus has shifted to machine learning models as they tend to perform well on non-linear data. Among machine learning models, Convolutional Neural Networks (CNN), recurrent neural networks (RNN) and Long Short-Term Memory (LSTM) models are the most effective in a range of financial forecasting studies. One important step in many methods is to reduce the signal noise. Example (Lahmiri & Bekiros, 2020) examined the LSTM and generalized regression neural networks (GRNN) performance in predicting the daily exchange rates of three virtual currencies: Bitcoin, Digital Cash, and Ripple. The strategy employed signal detrending techniques but examined just one data period and only used root mean square error (RMSE) as a metric, which is insufficient to determine the model's robustness. Massaoudi et al., (2020) used a hybrid deep learning model for short-term load forecasting using CNN-BiLSTM and Partial Least Square (PLS). The authors used the Savitzky–Golay filter to clean the variable signals and then fed the enhanced signal to the model. Mudassir et al., (2020) examined Support Vector Machines (SVM), LSTM, and Artificial Neural Networks (ANN), as well as stacked-ANN approaches, for forecasting short, medium, and long-term prices using a high dimension dataset. The best model had a mean absolute error (MAE) of 39.50 for the next day's forecast, and a root mean square error (RMSE) of 74.10. The forecast for the 7th-day horizon, on the other hand, was lower, with an MAE of 16.32 and an RMSE of 37.32, roughly half of what was indicated for the next-day forecast. These findings contradict the scientific consensus that the farther we forecast, the more the uncertainties and the magnitude of error variations should grow as prediction horizons increase. One probable explanation for this issue is that the time series in this work were shuffled before regression, introducing forward-peeking and overfitting. Hybridization of models can also enhance results. Mallqui and Fernandes (2019) used a recurrent neural network (RNN) based ensemble model to forecast the closing price of Bitcoin with an MAE of 14.32, RMSE of 25.47, and MAPE of 1.81% for their second interval. Adcock and Gradojevic (2019) demonstrated that ANNs can be superior to statistical and econometric models but did not consider the relative importance of varying inputs concerning each time interval.

Appropriate input data selection is equally critical for successful forecasting. It is based on the researchers' domain expertise and previous studies. In terms of input data selection, previous studies can be grouped into two categories: (1) using historical prices as inputs, and (2) using multivariate data as predictor factors. Certain works (Aggarwal et al., 2020; Faghih Mohammadi Jalali & Heidari, 2020; Gyamerah, 2020; Yao et al., 2018) have used univariate prediction models with lag values of closing price data. The univariate methods have their own advantages or limitations. Past studies have shown that in general, multivariate models perform better than univariate prediction models (Miller & Kim, 2021). However, as advised by Iwok and Okpe (2016), univariate models have their inherent strength in their simplicity, and it is always preferable to conduct a comparative analysis to determine which model works best for a given set of data.

Two primary strategies have been employed in multivariate forecasting. The first approach makes use of technical indicators as predictors. These predictors are constructed using price values for the Open, High, Low, and Close (OHLC) periods and trading volume. The second strategy uses fundamental predictors that form the intrinsic value of Bitcoin. Huang et al., (2019) used around 124 technical indicators to predict the Bitcoin daily returns of Bitcoin. These results indicated that owing to the highly speculative nature of Bitcoin; technical indicators were better predictors of returns than fundamental indicators. Cohen (2020) used Darvas box strategy, linear regression methods and technical indicators to forecast Bitcoin price shifts. On the other hand, some works (Chen et al., 2020; Zhu et al., 2017) have utilized fundamental predictors variables, including Blockchain information, Google search interest, twitter sentiments data, global currency exchange ratios, and macroeconomic variables to forecast the Bitcoin exchange prices. The evolution of Bitcoin's volatile price was studied by Wang and Hausken (2022), who examined data from 23 July 2010 to 21 June 2021. In order to forecast the future price of Bitcoin and the peaks of the bull market, the authors investigated five differential equation-based models using least squares and weighted least squares methods. The research used differential equations of different carrying capacities to analyze the pattern of Bitcoin's price increase. This work had a broader scope than prior research since it did not use predetermined short-term forecasting horizons but instead employed multiple-year forecasting horizons and five distinct differential equation-based growth models with dampened oscillations and lengthening cycles to identify many bear market minimums and bull market maximums. According to the research, the price of Bitcoin will stay volatile until 2100, and its market capitalization may eventually overtake traditional financial assets such as gold, stocks, bonds, and other financial assets. Very few works have performed a comparative analysis on the predictive ability of deep learning models using fundamental indicators, technical indicators, and univariate closing prices as input data.

Robust feature selection methodologies, data pre-processing, transformation, and strong algorithms are vital to accurate forecasting. Chen et al., (2020) used Boruta method for feature selection to predict the short-term stock market price trends. Deep learning methods have achieved high accuracy rates and often outperformed supervised learning and statistical methods. Lamothe-Fernández et al., (2020) proposed a deep recurrent convolutional neural network to forecast Bitcoin prices with 29 fundamental predictor variables. The model reported an accuracy of 94.42% accuracy on the out of sample data. Popular deep learning methods like LSTM, ANN, CNN and some hybrid models like CNN-LSTM are used by many recent studies to forecast Bitcoin prices (Chen et al., 2021; Huisu et al., 2018; Livieris et al., 2021; Mallqui & Fernandes, 2019; Mudassir et al., 2020; Wu et al., 2019). One common issue encountered by researchers while using deep learning models is optimizing the neural network hyperparameters. Bayesian Optimization (BO) has emerged as a viable solution in recent years to resolve this issue (Bakshy et al., 2018; Li et al., 2021; Ranjit et al., 2019). BO employs the restricted sample to create a posteriori probability distribution for the black box function to determine its best solution. In BO, a proxy model maps hyperparameters to model generalization accuracy.

The purpose of this work is to construct a hybrid framework capable of considerably improving Bitcoin price prediction and producing robust forecasts for a range of short-term prediction horizons. Keeping this in mind, we examine three distinct categories of input features, namely fundamental indicators, technical indicators, and lagged values of past Bitcoin prices, to identify the optimum Bitcoin price forecasting model for four distinct market stages. We use four deep learning models namely Deep Artificial Neural Network (DANN), LSTM, Bidirectional LSTM and a hybrid model CNN-BiLSTM to run our experiments. These models are optimized using Bayesian Optimization and regularization. Besides that, we address the issue of volatility and outliers in the Bitcoin time series and employ signal processing techniques to minimize noise interference.

The research's major contributions may be summarized as follows: first, a novel three-step feature selection procedure is given to identify variables with strong predictive potential. Second, we suggest a hybrid data smoothing technique for removing Bitcoin data outliers and intrinsic noise. We pre-process the data using Hampel and Savitzky–Golay filters. To our knowledge, this hybrid method has never been applied in the financial literature for data pre-processing. Finally, we use fundamental, technical, and historical price predictors with the pool of deep learning algorithms and optimize them with Bayesian Optimization to discover the model that produces the best multi-step ahead forecast. This kind of evaluation allows determining whether Bitcoin is a suitable hedging asset in the short period, resulting in more useful capital management.

2 Methodology and Data

2.1 Proposed Forecasting Framework

In this research, we propose a Hybrid forecasting framework that uses signal processing methods for data preprocessing and deep learning algorithms for predicting Bitcoin prices. Figure 1 describes the complete process used for solving the Bitcoin price prediction problem.

2.2 Dataset Description

The price of Bitcoin derives its value from multiple factors. We collected and prepared three different data groups to train our models: (1) fundamental indicators such as Blockchain information, Google search interest, tweet data, global currency exchange ratios, and macroeconomic variables. (2) Technical indicators created using movements of the past price and trading volume of Bitcoin, (3) lagged close price values of past days for univariate price modeling.

All data were collected from April 1, 2013, to November 6, 2021, consisting of 3142 daily observations. Since Bitcoin exchanges operate 24 h a day, seven days a week, and do not have standard daily Start and Close timings, we considered that the start of the day begins with the first trade after 00:00:00 Coordinated Universal Time (UTC) and ends at time 23:59:59 UTC for all Blockchain data. Tables 1 and 2 list the field information and description and previous financial forecasting sources for all data variables used as raw inputs. In Table 1, we gathered daily observations of Blockchain data classified by Group Ids 1–4, i.e., Blockchain wallet activity, Blockchain network activity, Blockchain mining information, and Blockchain market signals, from https://www.blockchain.com/charts#blockchain. The daily Google search trends index for the keyword "Bitcoin" was retrieved from https://trends.google.com (categorized as Group Id 4). The daily count of tweets with the hashtag "#Bitcoin" was acquired from https://bitinfocharts.com/comparison/tweets-btc.html#alltime (categorized as Group Id 4). Macro-economic indicators, and global currency ratios (categorized as Group Id 6–7) were extracted from https://www.investing.com. All data from above websites was retrieved using custom written python scripts.

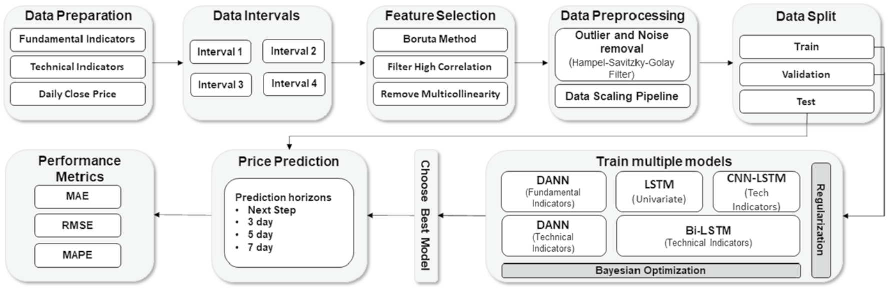

Fig. 1 Schematic architecture for the proposed forecasting framework

Table 1 List of Fundamental indicators used as independent variables. All data categories are comprised of daily observations

Group Id	Category	Description	References
1	Blockchain wallet activity	Newly registered wallets on blockchain.com, Total number of registered wallets	Aarhus et al. (2018), Chen et al. (2021), Jang and Lee (2018a), Nasir et al. (2019)
2	Blockchain network activity	Unique addresses used, confirmed transactions per day, Transaction rate per second, Output value per day, Mempool transaction count, Mempool size growth, Mempool size bytes, Unspent transaction outputs, Transactions excluding popular addresses, Estimated transaction value USD	Aarhus et al. (2018), Chen et al. (2021), Nasir et al. (2019)
3	Blockchain mining information	Total hash rate, Network difficulty, Miner's revenue USD, Total transaction fees in US$, Fees per transaction USD, Cost percent of transaction volume Cost of transaction	Aarhus et al. (2018), Chen et al. (2021), Jang and Lee (2018a), Nasir et al. (2019)
4	Blockchain market signals	Market value to realized value (MVRV), Network value to transactions (NVT)	Chen et al. (2021) and Nasir et al. (2019)
5	Social Media and Search Trends	Google Search Index with the keyword "Bitcoin," and Twitter tweets per day that include the hashtag "#Bitcoin"	Mallqui and Fernandes (2019), Politis et al. (2021)
6	Macroeconomic & Financial Indicators	S&P 500 Index, NASDAQ Index, Gold Price, WTI Crude Oil price, Dow Jones 100, US Fed Interest rates, US 10 Year Treasury rate, US Initial Claims (unemployment filing)	Chen et al. (2021) and Jang and Lee (2018a)
7	Global Currency/USD Ratios	CNY/USD, EUR/USD, GBP/USD, RUB/USD	Chen et al. (2021) and Jang and Lee (2018b)

In Table 2, we calculate all data using the Technical Analysis library (https://www.ta-lib.org) from the Bitcoin daily closing price. Additionally, we use lagged

Table 2 List of technical indicators used as independent variables for Bitcoin price prediction. All indicators are calculated from Bitcoin's daily closing price

#	Category	Description	Source(s)	Calculation(s)
1	Moving averages	1. *SMA*: 5, 10 period Simple Moving Average 2. *VAMA*: 9 period Volume Adjusted Moving Average 3. *TEMA*: 9 period Triple Exponential Moving Average 4. *EMA*: 9 period Exponential Moving Average 5. *DEMA*: 9 period Double Exponential Moving Average Exponential moving averages attempt to remove the inherent lag associated to Moving Averages by placing more weight on recent values	Yang Li et al. (2020), Mallqui and Fernandes (2019), Mudassir et al. (2020), Ntakaris et al. (2020) and, Politis et al. (2021)	TA-Lib
2	Price–volume indicators	*VWAP*: Volume Weighted Average Price is used to determine what the average price is based on both price and volume	Ntakaris et al. (2020) and Park et al. (2020)	TA-Lib
3	Momentum indicators	1. *MOM*: Market Momentum measured by continually taking price differences for 10 periods 2. *MACD*: Moving Average Convergence Divergence measures the relationship between two price moving averages of security 3. *PERCENT B*: Percent B Oscillator (%B) reflects the closing price as a percentage of the lower and upper Bollinger Bands 4. *Chaikin Oscillator*: It measures the accumulation/distribution line of the MACD 5. *ROC*: Rate of Change measures the percent change in price from one period to the next 6. *SO*: Stochastic Oscillator compares the closing price to the range of prices over a certain period of time (14 periods for this study)	Li et al. (2020), Mudassir et al. (2020), Ntakaris et al. (2020), and Politis et al. (2021)	TA-Lib
4	Oscillators	1. *TRIX*: Triple exponential average indicator is used to identify oversold and overbought markets 2. *Relative Strength Index (RSI)*: It measures the magnitude of recent price changes 3. *Williams %R*: Williams %R Oscillator shows the current closing price in relation to the high and low of the past N days	Mallqui and Fernandes (2019), Mudassir et al. (2020) and Sun et al. (2020)	TA-Lib

Source: Authors calculations

close prices from the previous 1, 2, and 3 days as our third raw input dataset. We collect and clean the data for these indicators and divide them into four separate intervals. We segment the data into four distinct intervals so that we can examine how the drivers of Bitcoin's price evolved overtime during each of these distinct market phases. We utilized the cubic spline interpolation method using polynomials of degree 3 to fill in any missing data for both Tables 1 and 2.

Three of these intervals are similar to prior state-of-the-art investigations (Adcock & Gradojevic, 2019; Mallqui & Fernandes, 2019; Mudassir et al., 2020) allowing us to assess the effectiveness of our proposed method. The fourth interval has not been investigated in the past literature and enables us to evaluate the most recent price movements. Table 3 describes the date ranges of all four data intervals considered for the experiments. We use all intervals to predict the next-day prices (time step $t+1$). Additionally, interval 4 is also used to predict the Bitcoin prices of three, five, and seven days ahead time horizons (time steps $t+3$, $t+5$, $t+7$). Unlike the first three intervals, Interval 4 contains the most recent market trends that provide a forward-looking assessment. Therefore, following previous research (Mudassir et al., 2020), we utilize interval 4 to construct a multi-step ahead forecast in addition to the next step forecast.

2.3 Feature Selection

Feature selection is the technique of selecting the most relevant, coherent, and non-redundant data attributes for model development. While feature selection is not mandatory for neural network-based models, a well-designed feature selection approach and appropriate data pre-processing techniques can considerably reduce computation time, improve the generalization ability of Deep Learning models and minimize overfitting. We use a three-step feature selection approach for the input datasets presented in Tables 1 and 2. In the First step we use Random Forest based Boruta method for initial feature selection then filter out the highly correlated variables and remove multicollinearity. The resultant subset of features is features with high predictive power and free from multicollinearity and high correlation.

We select important features using the Boruta technique (Kursa & Rudnicki, 2010) which improves the feature selection of Random Forest. All independent variables are duplicated and randomly shuffled in this procedure. These shuffled values are referred to as shadow features, and they are then combined with the original dataset. Following that, a Random Forest regressor is used to determine the variable's importance, followed by an iterative comparison of the original features' importance to their shadow copies. The key premise is that if a variable is chosen over n-iterations only after comparison with its shadow feature, it is not the outcome of randomness. On the other hand, if a shadow copy of a feature has more importance than the original feature then it can be deemed as noise. One of Boruta's limitations is that it neglects cross-correlations and does not account for multicollinearity because Random Forest determines its variable importance. Therefore, we manually eliminate strongly correlated variables and filter out the variables with a Variance Inflation Factor ≤ 10 as proposed by previous studies (Mudassir et al., 2020). The final feature subset produced by this three-stage procedure contains the high importance variables with lower correlation and multicollinearity. We repeat this process for each of the intervals specified in Table 3 to ensure that each feature selected is local to each interval and is representative of the actual Bitcoin prices for that interval. Tables 4 and 5 present the final list of all selected features.

2.4 Data Pre-processing and transformation

A model's forecasting performance can be enhanced by proper model design and data preprocessing, and the use of appropriate data, transformation, and feature engineering approaches. Data pre-processing is critical in improving the model's forecasting capabilities, as the model cannot be fitted directly to the raw data. According to Table 6, intervals 3 and 4 have an excess kurtosis greater than 3, indicating the presence of fat tails or extreme outlier occurrences in the data distribution. These outlier occurrences must be carefully treated.

The following steps are used to pre-process the data: (1) interpolating missing values, (2) eliminating outliers and reducing noise, and (3) transforming data to an appropriate scale. We use the cubic spline interpolation method to fill the missing observations, Hampel filter to detect and impute outliers, and Savitzky–Golay filter to minimize the noise in the Bitcoin time series. Following that, we apply a scaling pipeline to transform the data.

2.4.1 Hampel Filter for Removing Outliers

Outliers are anomalous observations whose values fall outside the expected range of data due to inherent random variability. Outlier detection and treatment is a vital step in enhancing data quality. We use the Hampel filter which is based on the Hampel identifier (Yao et al., 2019). The Hampel identifier is a reliable and efficient outlier detection technique that has been used in a variety of fields. The filter has a sliding window frame where each data observation is compared to the Median Absolute Deviation of the window (MAD). The central value in the local window is replaced with the median if the observation exceeds the MAD by 'n' times where 'n' is the threshold value. We choose a small window width of 15 and a MAD threshold value of 3 based on the nature of our data, the number of observations, and the volatility of Bitcoin prices. A small window width helps to minimize the data loss during imputation by suppressing the effect of largest strength values for that window. Bitcoin markets tend to generate extreme returns and losses. In such a scenario, eliminating outliers would imply that critical information is disregarded. As a result, we do not remove detected outliers but rather impute them with rolling medians of the sliding window.

Table 3 Four different intervals considered for experimentation

Intervals	Start date	End date
Interval 1	April 1, 2013	July 31, 2016
Interval 2	April 1, 2013	April 30, 2017
Interval 3	April 1, 2013	December 31, 2017
Interval 4	April 1, 2013	November 6, 2021

Table 4 The list of selected fundamental indicators from the three-stage feature selection process

Category	Feature	Interval 1	Interval 2	Interval 3	Interval 4
Search trends	Google Search Trend Index	✓	✓	✗	✓
Blockchain market signals	Network value to transactions	✗	✓	✗	✓
Blockchain mining information	Cost of transaction	✗	✓	✗	✓
	Cost percent of transaction volume	✓	✗	✗	✓
	Fees per transaction (US Dollar)	✓	✗	✓	✓
	Miners' revenue (US Dollar)	✓	✗	✗	✓
	Total transaction fees (US Dollar)	✗	✗	✓	✗
	Output value per day	✓	✓	✓	✓
	Unique addresses used	✗	✗	✓	✗
Macro-economic factors	US Initial claims close value	✗	✗	✗	✓

2.4.2 Savitzky–Golay Filter for Smoothing

Smoothing eliminates irregular oscillations from a signal and adjusts the data to minimize seasonal and cyclical components, resulting in a signal with less noise. We smoothen the data using Savitzky–Golay (SG) filter. The SG filter (Schafer, 2011) is a well-known method for reducing noise interference in signal processing. The primary benefit of the SG filter is that it retains the important characteristics of a signal, namely the area, phase, and amplitude of the signal spikes. The steps in the SG filtration method are as follows: (1) divide the signal into small time windows, (2) fit a polynomial of degree 'n' to the signal values inside each window for curve fitting, (3) use the fitted curve's midpoint as the new denoised data point, and (4) repeat the

Table 5 The list of technical indicators selected from the three-stage feature selection procedure. These indicators are calculated from Bitcoin's closing price

Category	Feature	Interval 1	Interval 2	Interval 3	Interval 4
Momentum indicator	CHAIKIN oscillator	✓	✓	✗	✗
Momentum indicator	Market momentum (MOM)	✓	✗	✓	✗
Oscillator	TRIX	✓	✓	✓	✗
Price volume indicator	VWAP	✓	✓	✗	✓
Momentum indicator	MACD	✓	✗	✗	✗
Oscillator	RSI	✗	✗	✗	✓
Oscillator	PERCENT B	✗	✗	✗	✓
Oscillator	Williams %R	✗	✗	✗	✓

process for all data points in a sliding window. The assumption is that by applying this strategy, a lower degree polynomial can be used to produce a better fit.

We aim to smooth the Bitcoin price data and denote the smoothened value at *k*th time interval as Ē. This can be expressed as:

$$\overline{\mathrm{E}}_k = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i E_{k+i}, \quad \frac{m+1}{2} \le k \le N - \frac{m-1}{2} \tag{1}$$

where m is the length of the window which must be an odd number, E_{k+i} is the noisy input signal at time $k+i$, C_i is the set of m convolution coefficients for ith smoothing and N is number of data points in smoothing window which must be equally spaced.

When the window length is increased, the cutoff frequency drops, resulting in a smoother signal and vice versa. A polynomial of a smaller degree, on the other hand, attenuates higher frequencies. The optimal window length should be determined by the type of data, and the extent of smoothing desired to avoid losing real signals. The degree (q) of the polynomial is chosen on the basis of its lowest error and best fit line. The criteria we employ in this study are consistent with prior work that focus on minimizing the Root Mean Square Error (RMSE) (Bi et al., 2020; Seo et al., 2018). We fixed the window size of m = 51 and after multiple trials of polynomial degrees q = {1,3,5,7,9}, we choose a polynomial degree of 5 for our investigation as it yielded the least RMSE against the original signal. Table 7 shows the RMSE values for different polynomial degrees. It is worth noting that the mode controls the extension applied to the padded signal. We use mode = 'nearest,' which indicates that the extension includes the input value closest to the current position.

2.4.3 Data Scaling and Transformation

After smoothing the data, we construct a pipeline in which we first apply zero mean and unit variance standard scaling. After that, we apply a robust scaler to remove the median and scale the data between the first and third quartiles, increasing the data's tolerance for outliers. After pre-processing the data, we divided each interval into independent subsets for Training, Validation, and Testing in the ratio of 70:15:15. In

Table 6 Descriptive statistics of Bitcoin's daily closing prices

Interval	Timestamp	Min	Count	Max	Mean	Std	Skewness	Kurtosis
Interval 1	April 1, 2013 to July 31, 2016	68.40	1218	1237.60	381.34	212.61	0.96	0.94
Interval 2	April 1, 2013 to April 30, 2017	68.40	1491	1351.90	467.54	282.77	0.90	0.27
Interval 3	April 1, 2013 to Dec 31, 2017	68.40	1736	19,345.50	1164.06	2342.75	4.60	24.37
Interval 4	April 1, 2013 to Nov 6, 2021	68.40	3142	65,979.10	8231.13	13,683.08	2.46	5.35

general, there is no optimal strategy for deciding the data split ratio. There is usually no best strategy for determining the optimal split ratio amongst Train, validation, and test subsets. However, the training set must have sufficient number of samples to avoid a high variance during model training. The first 70% of the time series is utilized for model training. The validation set (next 15% of the data) is used to evaluate the model's performance on the training data and reduce potential overfitting by adjusting the model's hyperparameters. For reliable test dataset results, the hyperparameters must be fine-tuned on training and validation datasets. The remaining data is used as a test set to assess the predictive performance of the trained model on unseen data.

2.5 Prediction Models

2.5.1 Deep Artificial Neural Network

Deep Neural Networks (DANNs) are Artificial Neural Networks with three or more hidden layers between the input and output layers. DNNs are universal approximators (Hornik et al., 1989) for complicated non-linear interactions and can estimate any continuous function. Figure 2 shows the architecture of an 'L' layered DANN.

Each input is assigned a weight denoted by W = (ω_1, ω_2, …, ω_{m}) and then passed to a summation function that computes the weighted sum of all inputs. Then, a bias term b_j is added to the weighted sum of network inputs to ensure that the activation function's input is not zero, even if the input to the network and connections are zero. Therefore, each node in the hidden and output layers has a bias associated with it. If x_j is the input to the node as the pre-activation at layer i can be expressed as:

$$x_j = x_1\omega_{1,j} + x_2\omega_{2,j} + \cdots + x_m\omega_{m,j} + b_j = \sum_{k=1}^{m} x_k\omega_{kj} + b_j \tag{2}$$

Table 7 Choice of an appropriate polynomial for Savitzky–Golay (SG) filter for noise reduction is based on lowest RMSE value

Polynomial degree (q)	RMSE
1	1334.43
3	992.61
5	**944.14**
7	951.08
9	991.61

The values in bold represent the lowest error for the specified degree of the polynomial

The table shows the RMSE value of SG Filters with signals of different polynomial degrees while keeping a fixed window width of 51 and mode = 'nearest'. Polynomial degree with lowest RMSE is chosen

Source: Authors calculations

where ω_{kj} denotes the connection strength between k and j, b_j is the bias. The layers between input and output layers are not exposed to any external source, so they are called hidden layers. The activation function, δ takes x_j as input and gives y_j as output. The output of the activation function that learns non-linear relations in the data is as follows:

$$y_j = \delta\left(x_j\right) \tag{3}$$

This output is passed on to the other nodes of the neural network. In this study, the Bitcoin price data is given as an input to the DANN model with three hidden layers. The network output O with (ω_1, ω_2, ω_3) and (b_1, b_2, b_3) as weights and biases of the network is calculated as follows:

$$\hat{y}_i = O\left(\omega^3\delta\left(\omega^2\delta\left(\omega^1 x + b_1\right) + b_2\right) + b_3\right) \tag{4}$$

The objective function we use for this study is Mean Square Error (MSE) which is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{k=1}^{n}\left(y_k - \hat{y}_k\right)^2 + r|l|^2 \tag{5}$$

where $\mathrm{y_k}$ is the kth observation of y and $\hat{\mathrm{y}}_\mathrm{k}$ is the predicted value, and n is the number of observations, r signifies the regularization used to avoid overfitting, and l denotes the neural network's hyperparameters. The problem of training a DANN model on an input dataset can be described as follows:

$$Minimize\, f\left(\vec{x}\right) = \overline{MSE} \tag{6}$$

where the objective is to minimize the training loss.

2.5.2 Long Short-Term Memory

Long Short Term Memory (LSTM) algorithm was introduced as and solution to effectively learn long-time lag problems by addressing the vanishing gradients (Hochreiter & Schmidhuber, 1997).

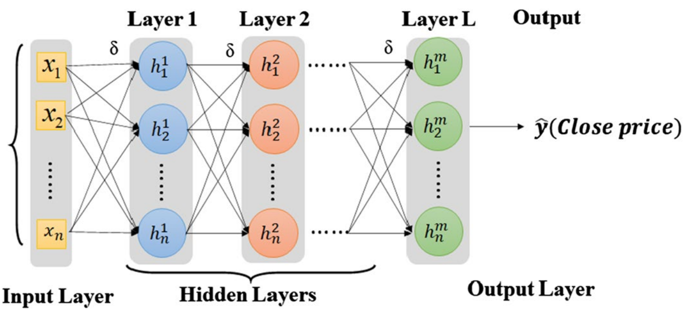

Fig. 2 Architecture of Deep Artificial Neural Network

The visual representation of an LSTM network memory cell unit is depicted in Fig. 3.

An LSTM network is similar to a Recurrent Neural Network (RNN), except that memory blocks substitute the hidden layer's summation units. The multiplicative gates of LSTM memory cells enable long-term storage and access of information, hence negating the issue of vanishing gradients. LSTM also maintains a memory cell c_t and a hidden state vector h_t. The LSTM can use the explicit gating mechanisms to read from, write to, or reset the cell at each time step. These gates referred to as input, output, and forget gates, are used to add or remove information in the network. A forward pass of an LSTM can be expressed by the following equations (Greff et al., 2017):

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{7}$$

$$i_t = \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{8}$$

$$o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{9}$$

$$c_t = f_t * c_{t-1} + i_t * \sigma_c\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{10}$$

$$h_t = o_t * \sigma_h\left(c_t\right) \tag{11}$$

where f_t is the forget gate that determines what information is retained from previous cell states and what is forgotten, σ_g is the sigmoid function, i_t is the activation vector of the input gate, o_t is the output gate's activation, c_t is the memory cell state vector at time t, σ_c, σ_h denote the hyperbolic tanh functions, h_t denotes the network's output

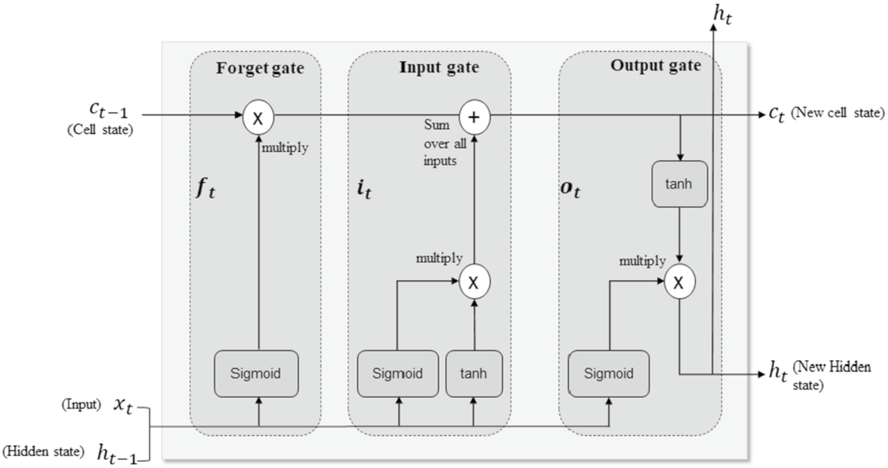

Fig. 3 Basic structure of an LSTM neural network memory cell

vector, W and U denote the weight and bias matrices for each gate. The symbol * denotes the elementwise product of the matrices.

2.5.3 Bi-Directional LSTM

Bidirectional LSTM (BiLSTM) is an improved variant of LSTM network where the input sequence is encoded in both forward and backward directions. In a Bi-LSTM network, following a forward pass, the input is reversed and fed into the LSTM network, enabling the network's temporal structure to evaluate a two-way association of the input data. Figure 4 illustrates a BiLSTM with forward and backward layers. Forward layer is responsible for receiving past sequence data. The backward layer accepts future data from the input sequence. Finally, the outputs of both hidden layers are integrated.

The output of the BiLSTM model's forward layer h_t^f and backward layer h_t^r is given as follows (Liu et al., 2020):

$$h_t = \theta h_t^f + \upsilon h_t^r \tag{12}$$

$$y_t = \sigma\left(h_t\right) \tag{13}$$

where y_t is the predicted value of tth sequence, where θ and υ are the factors controlling forward and backward LSTM, respectively. They adhere to the equivalence $\theta + \upsilon = 1$. σ is the softmax function.

2.5.4 CNN-BiLSTM

CNN-BiLSTM is a neural network architecture that combines Convolutional Neural Networks (CNN) with Bi-LSTM. The input is sent to a convolutional layer in this model, which identifies the internal structures in the input sequence. Following that, a pooling layer decreases the data's dimensionality. The pooling layer is connected to the Bi-LSTM and fully connected layers known as dense layers. The CNN-BiLSTM model employed for this study is based on (Ju et al., 2019; Massaoudi et al., 2020). The schematic architecture of a CNN-BiLSTM model is depicted in Fig. 5.

2.6 Hyperparameter Tuning

Following the identification of candidate models, the proper architecture for each model must be chosen. Manual network-tuning or conventional hyperparameter optimization approaches such as Grid-Search and Randomized-Search are computationally expensive and unfeasible in a multi-dimensional search space for real-world data.

We use Bayesian optimization technique (Snoek et al., 2012) to find the optimal configuration of hyper-parameters for all chosen models. Bayesian Optimization uses previously calculated hyperparameter values to guide the search for optimal parameters. We conduct the optimization experiments using Adaptive Experimentation library (AE) (Bakshy et al., 2018). AE is a well-known multi-objective optimization framework that has been used extensively within Facebook to conduct several

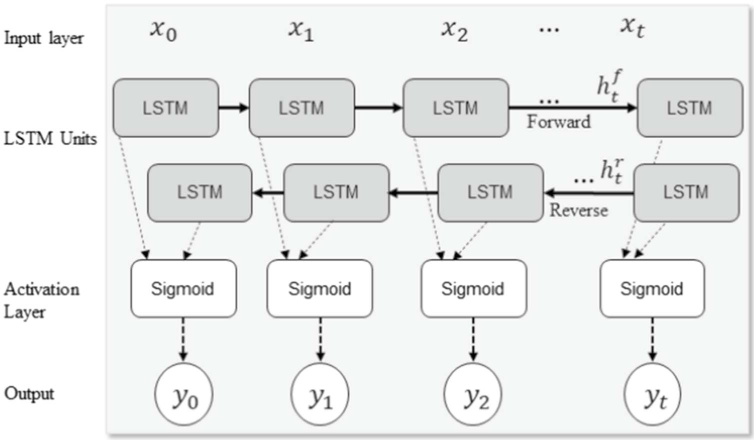

Fig. 4 Schematic diagram of a BiLSTM neural network

real-world online experiments. The primary benefit is its ability to cope successfully with black-box optimizations (those that lack mathematical expressions) and noisy functional evaluations. To identify the optimal network design, we optimize the following objectives concurrently: learning rate, dropout rate, number of hidden layers, neuron count per layer, batch size, activation function choice, and optimizer.

3 Experimental Settings

We conduct the experimentation using three types of inputs: (1) single feature input having close prices of lag 1,2, and 3, (2) input with fundamental indicators shown in Table 4, and (3) input with technical indicators listed in Table 5. We evaluate the following models to determine which is the best performer: DANN, LSTM, Bi-directional LSTM, and CNN-LSTM. These models are chosen after analyzing the performance metrics and conducting multiple trials for each input type and model combination. We use DANN with technical and fundamental indicators as input data, LSTM with single feature input of lagged closing prices, and Bi-LSTM and CNN-LSTM with technical indicators as input data for our final experiment. We follow the same training approach for both next-day and multi-step price forecasts. We used early stopping and regularization to prevent overfitting. As a result, the results are stochastic. Therefore, we repeat the trials for 15 runs per model and average the results.

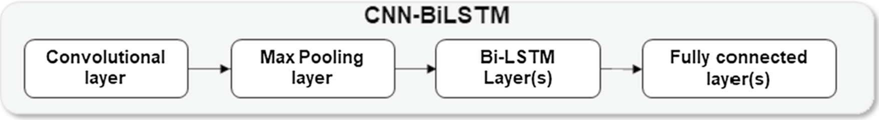

Fig. 5 Architecture of a CNN-BiLSTM neural network

Table 8 Bayesian Optimization parameters for search space exploration

Hyperparameter Name	Type	Bounds	Values	Description
Learning rate	Range	[0.00001, 0.01]	–	Learning rate to adjust the weights of the network
Dropout rate	Range	[0.001, 0.25]	–	Dropout regularization rate to ignore randomly selected neurons
Number of Hidden Layers	Range	[1, 5]	–	Number of hidden layers in the network
Neuron count per layer	Range	[400, 550]	–	Number of neurons to be used in a layer
Batch size	Choice	–	[8, 16, 32, 64, 80]	Batch Size or the count of samples to be used in one pass
Activation function	Choice	–	[linear, relu]	Activation function for the network
Optimizer	Choice	–	[rms, adam, adamx, sgd]	Optimizer for reducing the loss function

3.1 Parameter Settings

Table 8 shows the search space used in Bayesian Optimization to achieve the optimum model configuration. We initially started with larger bounds in the search space but narrowed it down after running several pre-assessment experiments for model optimizations.

Table 9 summarizes the hyperparameters that Bayesian Optimization suggests for each model. After running Bayesian Optimization and getting the best performing

Table 9 Network architecture suggested by Bayesian Optimization

Hyperparameter	DANN (fundamental indicators)	DANN (technical indicators)	BiLSTM (technical indicators)	LSTM (univariate)	CNN-BiLSTM (technical indicators)
Learning rate	0.0019	0.0019	0.0039	0.0038	0.0013
Dropout rate	0.0251	0.0019	0.2000	0.2000	0.0055
Number of Hidden Layers	3	3	5	4	5
Neuron units per layer	497	497	467	467	467
Batch size	8	8	80	80	80
Activation function	Linear	Linear	Linear	Linear	Linear
Optimizer	rms	adamx	rms	rms	adam

Source: Authors calculations

variant, we further improve the model performance by applying regularization manually. As noted in Eq. (5), regularization helps to minimize the objective function and prevents overfitting.

Table 12 in the appendix presents the model performance results of each model for each type of input. For our final comparative analysis of the results, we selected the model with the best performance among the fundamental predictors and univariate input types. In our pre-assessment experiments, models utilizing the technical indicators fared better than those utilizing the other two input types (fundamental, univariate). To present the most appropriate comparisons for all models, we rejected the LSTM model that delivered the worst performance for technical indicator inputs. Finally, we used the following input and model types for our final results:

1. Fundamental predictors: DANN
2. Technical indicators: DANN, BiLSTM, CNN-BiLSTM
3. Univariate input: LSTM

3.2 Performance Metrics

We use three metrics to assess and compare performance of all competing models:

1. Root mean square error (RMSE) $= \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y_k - \hat{y}_k\right)^2}$
2. Mean absolute error (MAE) $= \frac{1}{N}\sum_{k=1}^{N}\left|y_k - \hat{y}_k\right|$
3. Mean absolute percentage error (MAPE) $= \frac{1}{N}\sum_{k=1}^{N}\left|\frac{y_k-\hat{y}_k}{y_k}\right| \times 100$

where y_k is the kth observation of y and ŷ the predicted y value given the model.

All experiments were conducted on a Windows 10 machine equipped with an Intel Core i5, 8 CPUs running at 2.30 GHz, 20 GB RAM, and an NVIDIA GeForce GTX 1050 GPU. The deep learning methods were implemented using Python 3.7.3 64-bit, TensorFlow 2.6, and CUDNN10.1.105.

4 Results and Discussion

We run the experiments for each of the four periods separately for each model to get the forecast for the next day. Additionally, we perform a multi-step ahead forecast for the 3,5, and 7-day forecast horizons. The mean data for all performance measures are shown in Tables 10 and 11. Figure 6 shows the comparison of actual and predicted values for all competing prediction models for interval 4. As seen in Fig. 6 and Table 10, DANN with technical indicators is the best performing model for forecasting the next day closing prices for all four intervals. It implies that the model can learn local information about each market phase and remain relatively robust in the face of changing bitcoin market circumstances and variability in price. The fourth interval records relatively higher values for all three-performance metrics than the first three intervals. It can be attributed to the significant price volatility throughout 2020–2021.

Table 11 summarizes the forecast for the next 'N' days for interval 4, ranging from April 1, 2013, to November 6, 2021. DANN with technical indicators is the most promising model for 3 days and 5 days ahead forecasting horizons with the lowest MAE, RMSE, and MAPE values. Additionally, it reports the lowest RMSE of 192.33 and MAPE of 2.25% for the 7th day Bitcoin price forecast. CNN-BiLSTM, on the other hand, reports a marginally lower MAE value of 171.97 for the seventh day. It can also be observed from both Tables 10 and 11 that all models utilizing the technical indicators as input data outscore the DANN model with Fundamental indicators and LSTM model with lagged close prices. CNN-BiLSTM with technical indicators reports the next best scores overall for all time intervals and is fairly close to the results achieved by DANN with technical indicators. It is fair to infer that when technical indicators are used as input variables to the framework, the models yield better results. Other models outperform the univariate LSTM model, which uses only the lag prices as input data, suggesting that the model is inadequate to learn the chaotic nature of the Bitcoin time series.

Figure 7 shows the RMSE values for all time forecasting horizons for interval 4. The DANN model with technical indicators as inputs reports the lowest RMSE scores for all four intervals, followed by CNN-BiLSTM. When the model's fluctuation throughout prediction horizons is considered, CNN-BiLSTM exhibits much more volatility between next-day (RMSE 162.58) and seventh-day (RMSE 341.12) forecasts. As a result, the DANN model using technical indicators as input is the best prediction model when model robustness across market phases and performance metric variability are considered for all forecast horizons.

Furthermore, as shown in Table 9, Bayesian Optimization recommended a smaller network size with three hidden layers for the DANN model and five hidden layers for the CNN-BiLSTM model. In general, a smaller neural network size is preferable since it requires less computational time to train.

The results outperformed three previous benchmark studies with overlapping intervals (Adcock & Gradojevic, 2019; Mallqui & Fernandes, 2019; Mudassir et al., 2020). When compared to Mudassir et al., (2020), who improved the results of the first two studies and reported an MAE of 39.5, an RMSE of 52.51, and a MAPE of 1.44 percent for their best performing models, we observe that the DANN model with technical indicators improves each of the performance metrics by more than 40% for interval 3, their most recent reported period. Similarly, when compared to Chen et al., (2021), the best performing model that reported MAE of 156.2296, an RMSE of 233.883 and MAPE of 2.2602 for their most recent period, our best reported DANN model outperforms the MAE by 28%, RMSE by 52% and MAPE by over 100%.

Table 4 illustrates that no other attribute has equal importance for all intervals besides output value per day. For the most recent interval 4, blockchain data, Google search trends, and US initial claims are the most significant fundamental attributes. Interval 4 contains the most impactful boom phase of 2020–2021. Table 4 reveals that

Table 10 Mean results of 15 runs for next-day closing price prediction using for all four intervals

Intervals	Metrics	DANN (fundamental indicators)	DANN (technical indicators)	BiLSTM (technical indicators)	LSTM (univariate)	CNN-BiLSTM (technical indicators)
Interval 1	MAE	6.57	**0.58**	6.85	2.96	1.55
	RMSE	6.76	**0.80**	8.62	4.39	2.14
	MAPE	1.20	**0.10**	1.19	0.50	0.28
Interval 2	MAE	3.57	**1.38**	8.26	10.43	5.03
	RMSE	5.02	**1.61**	9.79	11.58	6.47
	MAPE	0.32	**0.14**	0.84	1.01	0.49
Interval 3	MAE	131.64	**18.95**	224.32	186.86	39.73
	RMSE	198.66	**52.51**	269.04	324.68	73.46
	MAPE	1.83	**0.18**	3.77	2.45	0.47
Interval 4	MAE	323.48	**121.99**	313.73	402.55	131.65
	RMSE	397.23	**153.93**	373.61	511.49	162.58
	MAPE	0.76	**0.28**	0.69	1.12	0.30

Source: Authors calculations

The values highlighted in bold signify the lowest prediction error

Table 11 Mean results of 15 runs for next 'N' days closing price prediction for Interval 4

Horizons	Metrics	DANN (fundamental indicators)	DANN (technical indicators)	BiLSTM (technical indicators)	LSTM (univariate)	CNN-BiLSTM (technical indicators)
3 days	MAE	343.38	**125.81**	329.27	413.94	147.65
	RMSE	412.76	**157.86**	385.66	526.69	258.12
	MAPE	2.96	**2.20**	2.14	1.99	0.60
5 days	MAE	350.79	**135.52**	340.70	428.55	151.65
	RMSE	426.13	**176.02**	395.57	551.49	321.12
	MAPE	4.96	**2.10**	3.06	3.12	3.46
7 days	MAE	361.79	181.72	344.06	472.55	**171.97**
	RMSE	456.81	**192.33**	403.66	551.49	341.12
	MAPE	4.13	**2.25**	3.37	4.12	4.72

Source: Authors calculations

The values highlighted in bold signify the lowest prediction error

majority of the significant features for interval 4 are classified within the blockchain information category. Network value to transactions, fees per transaction, miner's revenue, and output value per day reached record highs in 2020–2021, corresponding to interval 4. As increasing number of users joined Bitcoin's blockchain network after the 2017 surge, the network grew in size, and the influence of blockchain-related factors on the price also increased. Google search trends can be a good proxy for the public recognition and interest in Bitcoin over specific time intervals (more strongly in

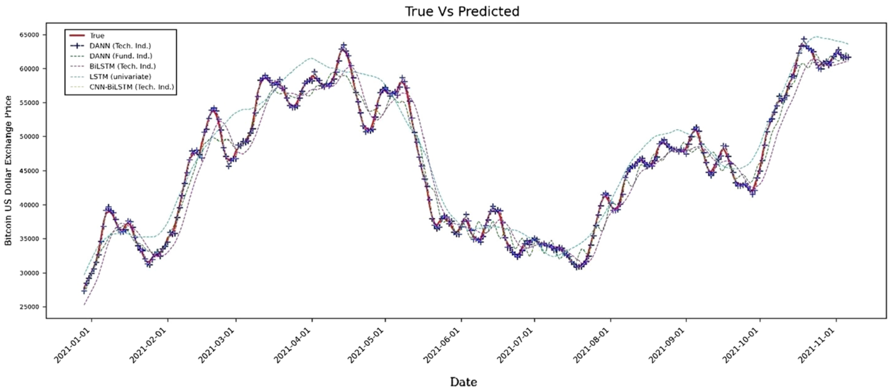

Fig. 6 Predictive comparison of True and Predicted values of Bitcoin closing prices for all competing models for interval 4

interval 4). Still, they may not consistently exhibit a strong correlation through all periods. Another significant feature in interval 4 is US initial jobless claims, which reflects the number of people who filed for unemployment insurance during the Covid19 pandemic. In the later parts of interval 4, strong Bitcoin surges and price drops can be linked to the unprecedented spike and subsequent correction in unemployment claims. Due to rising unemployment and stimulus funding during the pandemic and increased speculation, many institutional investors sought an alternative exposure to a market that promised a higher expected return.

Table 5 illustrates the significant features selected for different intervals. As seen, no single technical indicator is used for all periods. Due to the ever-changing nature of cryptocurrency markets, it makes sense that there cannot be a perfect technical indicator that works for all market phases. The feature importance criteria for intervals 1 to 4 vary according to market phases. Bitcoin halving event (Conway, 2021) is a major contributor to the change in market dynamics across all four intervals. Halving reduces the rate at which new coins are mined, reducing the quantity of fresh supply and increasing demand. Prior halving events were followed by substantial boom and bust cycles,

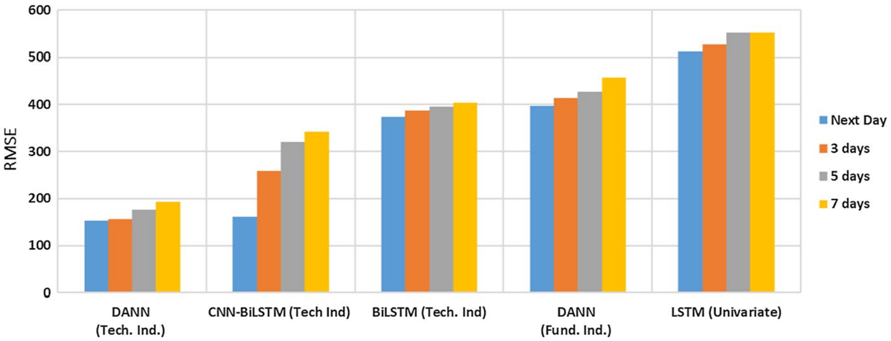

Fig. 7 Root Mean Square (RMSE) distribution for 1d, 3d, 5d, 7d step ahead forecasts for interval 4

resulting in overall higher prices than prior to the event. There have been three Bitcoin halving events in 2013, 2016, and 2020. The third halving occurred during Covid 19 in 2020, overlapping with interval 4. Furthermore, during interval 1 (2013–2016), the Bitcoin market was not severely affected by the additional factors of heightened regulation, large institutional holdings, systematic pump-and-dump networks, and a surge in competing cryptocurrencies. Nonetheless, as the market matured, these factors had a considerable influence on the price of Bitcoin throughout intervals 3 and 4. The technical indicators selected in interval 4 are mostly oscillators which help determine how fast the price of the underlying asset changes. The feature selection process responds to fluctuating market phases. It identifies oscillators that capture the speed of price movement, which varies dramatically from interval 1 to interval 4 due to the reasons mentioned above. Prior research (Mallqui & Fernandes, 2019) recommended using technical indicators such as Williams % R, MACD, and RSI to improve the accuracy of Bitcoin exchange price forecasting. The results of our feature selection process are consistent with their recommendation. Additionally, the feature selection procedure of each interval reveals that the Bitcoin price is influenced more by the most recent factors for a given interval than by the factors that worked in the preceding intervals. This finding is consistent with the findings of Wang and Hausken (2022), who observed that recent data has a greater impact on the price of Bitcoin than earlier data. Another important finding is that Bitcoin price keeps evolving for each interval as per the changing market dynamics. This corroborates the conclusions of Wang and Hausken (2022), who suggested that the Bitcoin price will continue to fluctuate until 2100 owing to dynamic market movements.

5 Conclusions

This paper proposes a hybrid deep neural network architecture to accurately forecast short-term Bitcoin exchange prices. The training approach utilizes univariate input of previous close prices and multi-variate feature inputs, namely financial and technical indicators. The predictive effect of each input strategy is systematically evaluated to determine the input-model combination that yields the best forecasting accuracy. Experiments are conducted using deep learning models with varying characteristics, strengths, and limitations, namely the DANN, LSTM, BiLSTM, and CNN-BiLSTM models.

In the first phase, we proposed a three-step feature selection approach to provide a strong feature subset with no multicollinearity among predictor variables. After that, signal processing methods are used for outlier treatment and data smoothing. We imputed outliers with the Hampel filter and used the Savitzky–Golay (SG) filter to remove noise from the series. Additionally, we examined the effect of window length and the polynomial degree of the SG filter to choose the most optimum signal-to-noise tradeoff for our input series. Finally, we employed Bayesian Optimization to find the optimal hyperparameter configuration for designing the architecture of neural networks.

We found that DANN outperformed its competitors when technical indicators were used as inputs. In addition, we discovered that DANN was the most reliable model with the lowest RMSE for multi-step ahead forecasting for the *N*th day forecasting horizon during the most significant out-of-sample period from January 1, 2021, to November 1, 2021, which recorded the highest volatility in Bitcoin's history. Moreover, the findings confirm a previous assertion (Huang et al., 2019) that technical indicators can be more accurate forecasters of Bitcoin prices than fundamentals. Furthermore, the experimental findings support the hypothesis that no technical indicator can have the same predictive ability throughout all market stages. It implies that the best forecasting strategy needs to use the predictor variables compatible with the market stage to achieve an optimal model performance.

The findings of this study have substantial implications for portfolio managers and algorithmic traders. Portfolio selection strategies seek an accurate price forecasting model that can avoid random market fluctuations and accurately forecast the price of each asset to maximize overall returns. The proposed forecasting method can be incorporated into a portfolio optimization system and utilized by portfolio managers to precisely forecast each asset's price, contributing to higher portfolio returns. Moreover, algorithmic trading systems can utilize the forecasting technique to generate trading strategy recommendations for sell, buy, and hold decisions.

Another implication of this study is the hybrid method of outlier and noise removal may be of interest to scholars conducting similar studies in the future. Additionally, the hybrid methodology contributes to the sparse literature on applying signal processing techniques for data pre-processing of financial time series. Further, the architecture could be applied in various financial and non-financial areas of research for feature selection, noise removal, and forecasting nonlinear data such as stock prices, electricity load, and weather. In the future, we intend to enhance the outlier detection and imputation procedures with more sophisticated methods. In place of Hampel filters, the approach for detecting outliers may employ more robust techniques, such as Autoencoder neural networks or Kalman filter. Effective treatments, such as Multiple Imputation (MI) and model-based imputation, can be used to enhance the outlier imputation. In addition, we intend to train the neural networks using the same framework for intraday algorithmic trading that uses high-frequency data and generates entry and exit rules for automatically placing buy and sell orders based on the forecast.

Appendix

See Appendix Table 12.

Table 12 Mean results of 15 runs for next-day closing price prediction using for all four intervals obtained in the experimental test harness for all models with all three types of inputs

Interval	Metric	DANN			BiLSTM			LSTM			CNN-BiLSTM		
		FI	TI	Uni	FI	TI	Uni	FI	TI	Uni	FI	TI	Uni
I 1	MAE	6.57	**0.58**	3.55	8.54	6.90	3.91	9.40	8.63	2.96	8.08	1.55	3.79
	RMSE	6.76	**0.80**	5.27	8.79	7.10	5.79	9.67	8.88	4.39	8.31	2.14	5.62
	MAPE	1.20	**0.10**	0.60	1.56	1.26	0.66	1.72	1.58	0.50	1.48	0.28	0.64
I 2	MAE	3.57	**1.38**	12.52	4.64	3.75	13.77	5.11	4.69	10.43	4.39	5.03	13.35
	RMSE	5.02	**1.61**	13.90	6.53	5.27	15.29	7.18	6.59	11.58	6.17	6.47	14.82
	MAPE	0.32	**0.14**	1.21	0.42	0.34	1.33	0.46	0.43	1.01	0.39	0.49	1.29
I 3	MAE	131.60	**18.95**	224.28	171.08	138.22	246.71	188.19	172.78	186.90	161.87	39.73	239.23
	RMSE	198.70	**52.51**	389.64	258.31	208.59	428.60	284.14	260.74	324.70	246.19	73.46	415.62
	MAPE	1.83	**0.18**	2.94	2.38	1.92	3.23	2.62	2.40	2.45	2.25	0.47	3.14
I 4	MAE	323.50	**122.00**	483.12	420.55	339.65	531.43	462.61	424.56	402.60	399.52	131.70	515.33
	RMSE	397.20	**153.90**	613.80	516.36	417.09	675.18	568.00	521.36	511.50	488.56	162.60	654.72
	MAPE	0.76	**0.28**	1.34	1.06	0.80	1.48	1.09	0.92	1.12	0.93	0.30	1.43

The values highlighted in bold signify the lowest prediction error

FI denotes fundamental indicators; TI represents technical indicators; Uni signifies univariate lagged input; I1, I2, I3, I4 represent Intervals 1, 2, 3 and 4

Funding The authors received no funding for the submitted work from any organization.

Declarations

Conflict of Interest The authors have no relevant financial or non-financial interests to disclose.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aarhus, H., Molnár, P., Erik, J., & Vries, D. (2018). What can explain the price , volatility and trading volume of Bitcoin ? *Finance Research Letters*. https://doi.org/10.1016/j.frl.2018.08.010

Adcock, R., & Gradojevic, N. (2019). Non-fundamental, non-parametric Bitcoin forecasting. *Physica a: Statistical Mechanics and Its Applications, 531*, 121727. https://doi.org/10.1016/j.physa.2019.121727

Aggarwal, D., Chandrasekaran, S., & Annamalai, B. (2020). A complete empirical ensemble mode decomposition and support vector machine-based approach to predict Bitcoin prices. *Journal of Behavioral and Experimental Finance, 27*, 100335. https://doi.org/10.1016/j.jbef.2020.100335

Bakshy, E., Dworkin, L., Karrer, B., Kashin, K., Letham, B., Murthy, A., & Facebook, S. S. (2018). AE: A domain-agnostic platform for adaptive experimentation. *Nips*, 8.

Bi, J., Lin, Y., Dong, Q., Yuan, H., & Zhou, M. C. (2020). An improved attention-based LSTM for multistep dissolved oxygen prediction in water environment. In *2020 IEEE international conference on networking, sensing and control, ICNSC 2020*. https://doi.org/10.1109/ICNSC48988.2020.9238097

Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting, 37*(1), 28–43. https://doi.org/10.1016/j.ijforecast.2020.02.008

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics, 365*, 112395. https://doi.org/10.1016/j.cam.2019.112395

Cheung, A., Wai, K., Roca, E., & Su, J. J. (2015). Crypto-currency bubbles: an application of the Phillips–Shi–Yu (2013) methodology on Mt. Gox bitcoin prices. *Applied Economics, 47*(23), 2348–2358. https://doi.org/10.1080/00036846.2015.1005827

Cohen, G. (2020). Forecasting bitcoin trends using algorithmic learning systems. *Entropy*. https://doi.org/10.3390/E22080838

Conway, L. (2021). Bitcoin halving: What you need to know. *investopedia.com*. Retrieved June 6, 2022, from https://www.investopedia.com/bitcoin-halving-4843769

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems, 28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

Gyamerah, S. A. (2020). On forecasting the intraday bitcoin price using ensemble of variational mode decomposition and generalized additive model. *Journal of King Saud University—Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2020.01.006

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huang, J. Z., Huang, W., & Ni, J. (2019). Predicting bitcoin returns using high-dimensional technical indicators. *Journal of Finance and Data Science, 5*(3), 140–155. https://doi.org/10.1016/j.jfds.2018.10.001

Huisu, J., Lee, J., Ko, H., & Lee, W. (2018). Predicting bitcoin prices by using rolling window LSTM model. https://doi.org/10.475/123_4

Iwok, I. A., & Okpe, A. S. (2016). A comparative study between univariate and multivariate linear stationary time series models. *American Journal of Mathematics and Statistics, 6*(5), 203–212. https://doi.org/10.5923/j.ajms.20160605.02

Faghih Mohammadi Jalali, M., & Heidari, H. (2020). Predicting changes in Bitcoin price using grey system theory. *Financial Innovation*. https://doi.org/10.1186/s40854-020-0174-9

Jang, H., & Lee, J. (2018a). An empirical study on modeling and prediction of bitcoin prices with Bayesian neural networks based on blockchain information. *IEEE Access, 6*, 5427–5437. https://doi.org/10.1109/ACCESS.2017.2779181

Ji, Q., Bouri, E., Keung, C., Lau, M., & Roubaud, D. (2018). International review of financial analysis dynamic connectedness and integration in cryptocurrency markets. *International Review of Financial Analysis*. https://doi.org/10.1016/j.irfa.2018.12.002

Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., & Rehman, M. U. (2019). A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting. *IEEE Access, 7*, 28309–28318. https://doi.org/10.1109/ACCESS.2019.2901920

Kristoufek, L. (2015). What are the main drivers of the bitcoin price? Evidence from wavelet coherence analysis. *PLoS ONE, 10*(4), 1–15. https://doi.org/10.1371/journal.pone.0123923

Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the boruta package. *Journal of Statistical Software, 36*(11), 1–13. https://doi.org/10.18637/jss.v036.i11

Lahmiri, S., & Bekiros, S. (2020). Intelligent forecasting with machine learning trading systems in chaotic intraday Bitcoin market. *Chaos, Solitons and Fractals*. https://doi.org/10.1016/j.chaos.2020.109641

Lamothe-Fernández, P., Alaminos, D., Lamothe-López, P., & Fernández-Gámez, M. A. (2020). Deep learning methods for modeling bitcoin price. *Mathematics, 8*(8), 1–13. https://doi.org/10.3390/MATH8081245

Li, Yaru, Zhang, Y., & Cai, Y. (2021). A new hyper-parameter optimization method for power load forecast based on recurrent neural networks. *Algorithms*. https://doi.org/10.3390/a14060163

Li, Y., Zheng, Z., & Dai, H. N. (2020). Enhancing bitcoin price fluctuation prediction using attentive LSTM and embedding network. *Applied Sciences (switzerland), 10*(14), 1–17. https://doi.org/10.3390/app10144872

Liu, Z., & xian, Zhang, D. gan, Luo, G. zhao, Lian, M., & Liu, B. (2020). A new method of emotional analysis based on CNN–BiLSTM hybrid neural network. *Cluster Computing, 23*(4), 2901–2913. https://doi.org/10.1007/s10586-020-03055-9

Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., & Pintelas, P. (2021). An advanced CNN-LSTM model for cryptocurrency forecasting. *Electronics (switzerland), 10*(3), 1–16. https://doi.org/10.3390/electronics10030287

Mallqui, D. C. A., & Fernandes, R. A. S. (2019). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing Journal, 75*, 596–606. https://doi.org/10.1016/j.asoc.2018.11.038

Massaoudi, M., Refaat, S. S., Abu-Rub, H., Chihi, I., & Oueslati, F. S. (2020). PLS-CNN-BiLSTM: An end-to-end algorithm-based Savitzky–Golay smoothing and evolution strategy for load forecasting. *Energies, 13*(20), 1–29. https://doi.org/10.3390/en13205464

Miller, D., & Kim, J.-M. (2021). Univariate and multivariate machine learning forecasting models on the price returns of cryptocurrencies. *Journal of Risk and Financial Management, 14*(10), 486. https://doi.org/10.3390/jrfm14100486

Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: A machine learning approach. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05129-6

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Www. Bitcoin. Org*. https://doi.org/10.1007/s10838-008-9062-0

Nasir, M. A., Huynh, T. L. D., Nguyen, S. P., & Duong, D. (2019). Forecasting cryptocurrency returns and volume using search engines. *Financial Innovation*. https://doi.org/10.1186/s40854-018-0119-8

Ntakaris, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2020). Mid-price prediction based on machine learning methods with technical and quantitative indicators. *PLoS ONE, 15*(6), 1–39. https://doi.org/10.1371/journal.pone.0234107

Park, C.-Y., Tian, S., & Zhao, B. (2020). Global Bitcoin markets and local regulations. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3590919

Politis, A., Doka, K., & Koziris, N. (2021). Ether price prediction using advanced deep learning models. In *IEEE International conference on blockchain and cryptocurrency, ICBC 2021* (pp. 0–2). https://doi.org/10.1109/ICBC51069.2021.9461061

Ranjit, M. P., Ganapathy, G., Sridhar, K., & Arumugham, V. (2019). Efficient deep learning hyperparameter tuning using cloud infrastructure: Intelligent distributed hyperparameter tuning with Bayesian optimization in the cloud. In *IEEE international conference on cloud computing, CLOUD*, *2019-July* (pp. 520–522). https://doi.org/10.1109/CLOUD.2019.00097

Schafer, R. W. (2011). What Is a Savitzky–Golay filter? [Lecture notes]. *IEEE Signal Processing Magazine, 28*(4), 111–117. https://doi.org/10.1109/MSP.2011.941097

Seo, J., Ma, H., & Saha, T. K. (2018). On Savitzky–Golay filtering for online condition monitoring of transformer on-load tap changer. *IEEE Transactions on Power Delivery, 33*(4), 1689–1698. https://doi.org/10.1109/TPWRD.2017.2749374

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems, 4*, 2951–2959.

Sun, S., Wang, S., & Wei, Y. (2020). A new ensemble deep learning approach for exchange rates forecasting and trading. *Advanced Engineering Informatics*. https://doi.org/10.1016/j.aei.2020.101160

Szetela, B., Mentel, G., & Gędek, S. (2016). Dependency Analysis between Bitcoin and Selected Global Currencies. *Analiza Zależności Pomiędzy Bitcoinem a Wybranymi Walutami., 16*, 133–144. https://doi.org/10.12775/DEM.2016.009

Wang, G., & Hausken, K. (2022). A Bitcoin price prediction model assuming oscillatory growth and lengthening cycles. *Cogent Economics & Finance*. https://doi.org/10.1080/23322039.2022.2087287

Wu, C. H., Lu, C. C., Ma, Y. F., & Lu, R. S. (2019). A new forecasting framework for bitcoin price with LSTM. In *IEEE international conference on data mining workshops, ICDMW*, *2018-Novem* (pp. 168–175). https://doi.org/10.1109/ICDMW.2018.00032

Yao, Y., Yi, J., Zhai, S., Lin, Y., Kim, T., Zhang, G., & Lee, L. Y. (2018). Predictive analysis of cryptocurrency price using deep learning. *International Journal of Engineering and Technology(UAE)*, *7*(3.27 Special Issue 27), 258–264. https://doi.org/10.14419/ijet.v7i1.5.9158

Yao, Z., Xie, J., Tian, Y., & Huang, Q. (2019). Using Hampel identifier to eliminate profile-isolated outliers in laser vision measurement. *Journal of Sensors, 2019*, 3823691. https://doi.org/10.1155/2019/3823691

Zhu, Y., Dickinson, D., & Li, J. (2017). Erratum to: Analysis on the influence factors of Bitcoin's price based on VEC model. *Financial Innovation, 3*(1), 7. https://doi.org/10.1186/s40854-017-0057-x

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.