Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05129-6

S.I. : DATA FUSION IN THE ERA OF DATA SCIENCE

Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach

Mohammed Mudassir[1] · Shada Bennbaia[1] · Devrim Unal[2] · Mohammad Hammoudeh[3]

Received: 15 April 2020 / Accepted: 16 June 2020

Abstract
Bitcoin is a decentralized cryptocurrency, which is a type of digital asset that provides the basis for peer-to-peer financial transactions based on blockchain technology. One of the main problems with decentralized cryptocurrencies is price volatility, which indicates the need for studying the underlying price model. Moreover, Bitcoin prices exhibit non-stationary behavior, where the statistical distribution of data changes over time. This paper demonstrates high-performance machine learning-based classification and regression models for predicting Bitcoin price movements and prices in short and medium terms. In previous works, machine learning-based classification has been studied for an only one-day time frame, while this work goes beyond that by using machine learning-based models for one, seven, thirty and ninety days. The developed models are feasible and have high performance, with the classification models scoring up to 65% accuracy for next-day forecast and scoring from 62 to 64% accuracy for seventh–ninetieth-day forecast. For daily price forecast, the error percentage is as low as 1.44%, while it varies from 2.88 to 4.10% for horizons of seven to ninety days. These results indicate that the presented models outperform the existing models in the literature.

Keywords Time-series forecasting · Deep learning · Machine learning · Blockchain

1 Introduction

Digital transformation of economies is the most serious disruption that is taking place now in all economies and financial systems. The economies and financial systems of the world are becoming digital at an unprecedentedly fast pace. According to a recent report, the size of digital economy in 2025 is estimated to be 25% (23 trillion USD), consisting of tangible and intangible digital assets [1]. The most recent technology for establishing and spending digital assets is the distributed ledger technology (DLT), and its most well-known application being the cryptocurrency named Bitcoin [2]. Following these developments, blockchain technology has found its place in the intersections of Fintech and next-generation networks [3].

An important issue about the non-tangible digital assets, and especially cryptocurrencies, is price volatility. The price of Bitcoin (BTC) for the period of April 1, 2013, to December 31, 2019, can be seen in Fig. 1. BTC prices have exhibited extreme volatility in this period. The price has increased 1900% in the year 2017, consecutively losing 72% of its value in 2018 [4]. Prior to 2013, the popular interest in BTC, its usage in virtual transactions and its prices have been low. That period is not considered in our models. Although the BTC prices exhibit extraordinary volatility, BTC as a digital asset is quite resilient as it can regain its value after significant drops, and even when the uncertainty is high in the market such as during the COVID-19 pandemic [5].

✉ Devrim Unal
dunal@qu.edu.qa

Mohammed Mudassir
mh1204033@student.qu.edu.qa

Shada Bennbaia
sb1209176@student.qu.edu.qa

Mohammad Hammoudeh
M.Hammoudeh@mmu.ac.uk

[1] Department of Mechanical and Industrial Engineering, Qatar University, Doha, Qatar

[2] KINDI Center for Computing Research, Qatar University, Doha, Qatar

[3] Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK

Published online: 04 July 2020

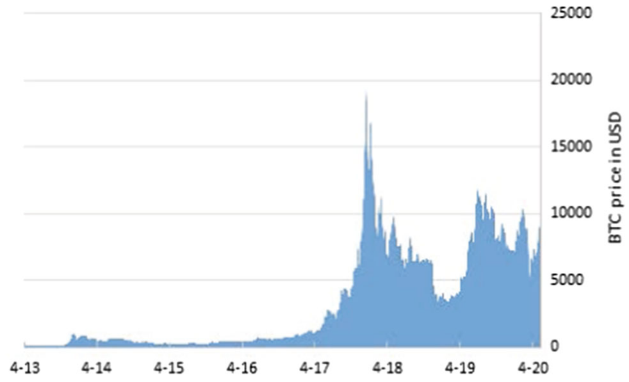


Fig. 1 Bitcoin (BTC) prices from April 2013 to April 2020

Despite its rapidly changing nature, the price of BTC has been an area where various researchers have presented efforts for price forecast. A number of studies have discussed whether BTC prices are predictable using technical indicators and demonstrated the existence of significant return predictability [6, 7]. Other recent studies such as [8, 9] and [10] have applied various machine learning-related methods for end-of-day price forecast and price increase/decrease forecasting. [9] reported maximum accuracy up to 63% for forecasting of increase or decrease of prices. [10] reported 98% success rate for daily price forecast. However, the time periods of these studies have been limited by data—up to April 1, 2017 [10] and up to March 5, 2018 [9]. We believe that a current study is needed considering the volume of the BTC price movements that occurred after these dates. Secondly, the cited works focus on end-of-day closing price forecast and price increase/decrease forecasting for the next day prices. In our study, we address mid-term price forecast and increase/decrease forecasting for horizons of forecast ranging from 7 day to 90 days, as well as daily closing price forecast, and price increase/decrease forecasting for the short term (end-of-day and next day). In addition, this is the first study that takes into consideration all the price indicators up to December 31, 2019, and provides highly accurate end-of-day, short-term (7 days) and mid-term (30 and 90 days) BTC price forecasts using machine learning.

Our performance results indicate that our results are better than the latest literature in daily closing price forecast and price increase/decrease forecasting. Additionally, we present high-performance neural-network-based models for medium term (7, 30 and 90 days) BTC price forecasts and price increase/decrease forecasting.

2 Related work

When Bitcoin began to get worldwide attention at end of 2013, it witnessed a significant fluctuation in its value and number of transactions [11]. A strand of literature has examined the predictability of BTC returns through various parameters such as social media attention [12, 13] and BTC-related historical technical indicators [14]. One group studied the period from September 4, 2014, to August 31, 2018, by capturing the number of times the term “Bitcoin” has been tweeted. The results showed that the number of tweets on Twitter can influence BTC trading volume for the following day [15]. Moreover, [16] studied the influence of users comments in online platforms on price fluctuations and number of transaction of cryptocurrencies and found that BTC is particularly correlated with the number of positive comments on social media. They reported an accuracy of 79% along with Granger causality test, which implies that user opinions are useful to predict the price fluctuations.

When it comes to time-series forecasts, there are three different types of model based approaches for time-series forecast according to [17]. The first approach, pure models, only uses the historical data on the variable to be predicted. Examples of pure time-series forecast models are Autoregressive Integrated Moving Average (ARIMA) [18] and Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) [19]. [20] presents an ARIMA-based time-series forecast for next-day BTC prices. However, we have not yet seen a study based on GARCH.

Pure time-series models are more appropriate for univariate and stationary time-series data. In this paper, we focus on machine learning with higher level features rather than the traditional models for the following reasons. First of all, BTC prices are highly volatile and non-stationary. We demonstrate that BTC prices are non-stationary in the next section. Secondly, there are a large number of features in the data and the proposed machine learning methodology handles autocorrelation, seasonality and trend effects, while the training process of pure time-series models require manual tuning to address these effects.

The second approach, explanatory models, uses a function of predictor variables to predict the target variable in a future time. Model-based time-series forecast approaches have the disadvantage of making a prior assumption about data distributions. For example, [20] and [21] are based on a log-transformation of the BTC prices. Similarly, [21] used daily BTC data from September 2011 up to August 2017 to conduct an empirical study on modeling and predicting the BTC price that compare the Bayesian neural network (BNN) with other linear and nonlinear benchmark models. They found that BNN performs well in predicting BTC log-transformed price, explaining the high volatility of the BTC price. However, the above-mentioned studies have used log-transformed prices for reporting performance metrics, which are misleading, as such values tend to be lower than performance metrics computed using real prices. We have analyzed this by calculating the performance metrics using log-normalized values and comparing against the non-log-normalized ones for our own results and found that although the log-normalized price forecast reports a much lower MAPE value, the actual error may be up to 10 times higher.

Since cryptocurrency prices are nonlinear and non-stationary, the assumptions on data distributions may have adverse effects on the forecast performance. Non-stationary time-series models exhibit evolving statistical distributions over time, which results in a changing dependency behavior between the input and output variables. Machine learning-based approaches utilize the inherent nonlinear and non-stationary aspects of the data. They can also take advantage of the explanatory features by taking into consideration the underlying factors affecting the predicted variable. There are several research studies on modeling and forecasting the price of BTC using machine learning,

[22] used Bayesian regression method that utilizes latent source model which was developed by [23] to predict the price variation using BTC historical data. [24] used machine learning and feature engineering to investigate how the BTC network features can influence the BTC price movements. They obtained classification accuracy of 55%. [9] used artificial neural network (ANN) to achieve a classification accuracy of 65%. Furthermore, [25] predicted the BTC price using Bayesian optimized recurrent neural network (RNN) and long short-term memory (LSTM). The classification accuracy they achieved was 52% using LSTM with RMSE of 8%. They also reported that in forecasting, the nonlinear deep learning models performed better than ARIMA. [10] employed ANN and SVM algorithms in regression models to predict the minimum, maximum and closing BTS prices and reported that SVM algorithm performed best with MAPE of 1.58%. One of the latest studies in predicting BTC daily prices is by [8], which used high-dimensional features with different machine learning algorithms such as SVM, LSTM and random forest. For next-day price forecast from July 2017 to January 2018, the highest accuracy of 65.3% was achieved by SVM.

3 Methodology

In this study, we are focusing on the time-series forecast of BTC prices using machine learning. A time-series is a set of data values with respect to successive moments in time. Time-series forecast is the forecast of future behavior by analyzing time-series data. The objective is to estimate the value of a target variable x in a future time point $\hat{x}[t+s] = f(x[t], x[t-1], \ldots, x[t-n]), s > 0$, where s is the horizon for forecast. We take into consideration, end-of-day, 7, 30 and 90 days as the horizon for forecast.

Figure 2 gives an overview of the main ideas used in this paper. The ML-based time-series forecast method starts with the construction of a dataset. This is followed by the training of ML models and forecasts based on these models for different horizons of forecast. Time-series forecast on cryptocurrency prices has underlying interdependencies that are hard to understand and model. For example, there are statistical factors such as variance and standard deviation that changes over time. Those interdependencies show up as technical indicators, which are explained in Sect. 3.1. In our study, open data sources have been utilized for gathering the BTC price technical indicators. In the data pre-processing step, data are gathered, cleaned and scaled/normalized. The collected BTC data are processed and divided into three intervals. Feature selection is used to identify relevant features. Furthermore, based on the third interval, the datasets of *n*th day forecast/forecast are created. We produce multiple datasets for different horizon of forecasts, for three different time periods, and exercise feature selection separately for each dataset. Feature selection is the most important step in ML for time-series forecast and explained in Fig. 5 and in Sect. 3.2.1. Feature selection is done to extract high ranking features from each of these datasets, using the random forest (RF) method and pruned based on variance inflation factor (VIF) and Pearson cross-correlation. The candidate features are explanatory features based on different statistical data about the operation of the blockchain itself, as well as technical market indicators. The datasets are split into training and validation sets. The ML classification and regression models are trained on the training split and validated on the holdout split.

The main difference of ML-based approaches from model-based methods for time series is the training phase. ML methods extract high-dimensional statistical trends and underlying features from the training data to allow it to predict the outcome in previously unseen cases. ML for time-series forecast can be used for classification and regression. We use the following ML models for classification and regression: artificial neural network (ANN), stacked artificial neural network (SANN), support vector machines (SVM) and long short-term memory (LSTM). The classification is applied as follows: If the BTC daily closing price $P_{BTC}[t+1] - P_{BTC}[t] \geq 0$, then $y[t] = +1$, and if $P_{BTC}[t+1] - P_{BTC}[t] < 0$, then $y[t] = 0$, where $y[t]$ is a target variable for categories of increasing and decreasing price. The regression models are used to predict BTC

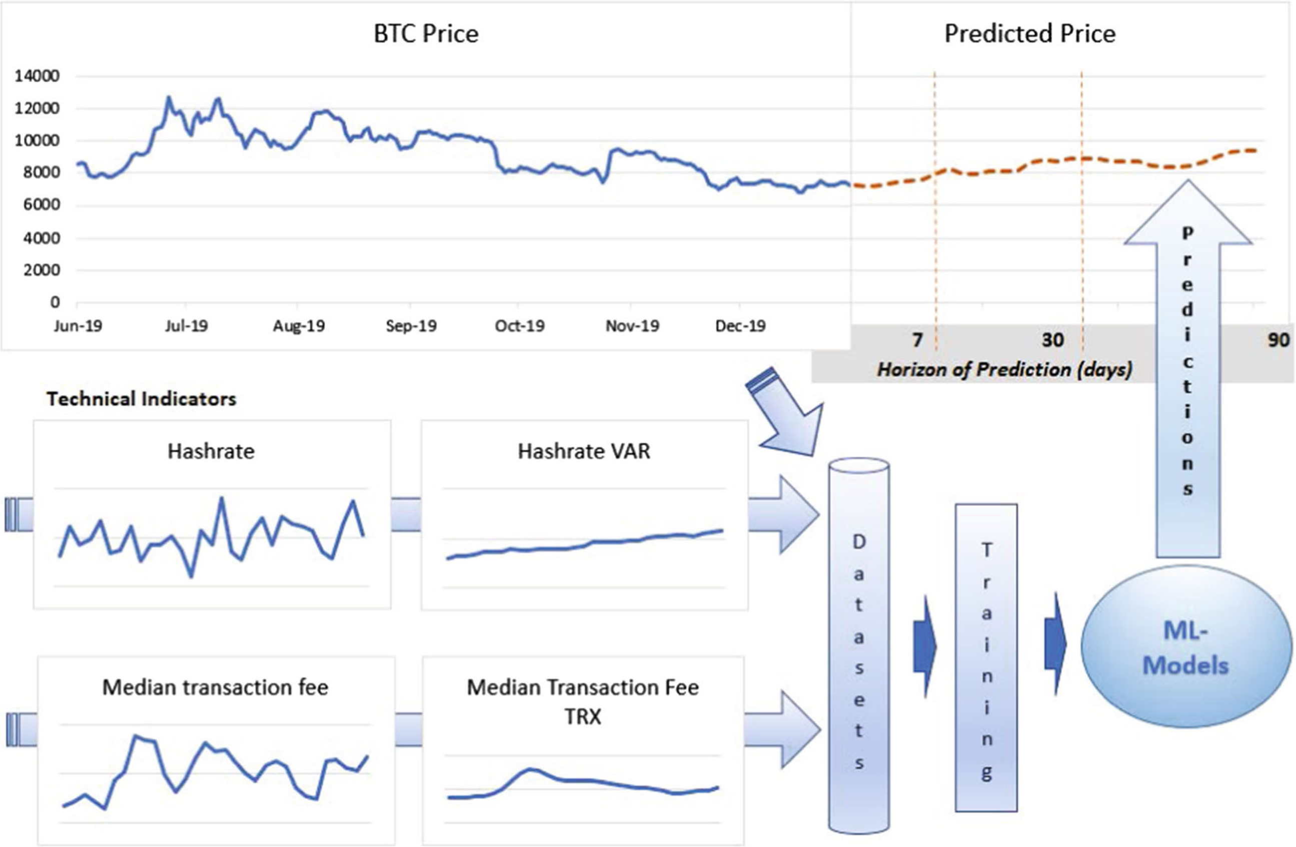


Fig. 2 ML-based time-series forecast using technical indicators

prices in a horizon of forecast for end-of-day, 7, 30 and 90 days. Figure 3 depicts the detailed steps of ML-based methodology used in this paper.

BTC prices prove to be a non-stationary time series, based on the unit root augmented Dickey-Fuller (ADF) testing, with ADF statistic of -1.6188 at 1% significance level ($\mathrm{ADF}_{\mathrm{critical}} = -3.433, p = 0.47$). For higher-order autoregressive processes given by (1), the ADF test checks for the non-stationarity by testing the null hypothesis, $H_0 : \delta = 0$, against the alternative, $H_1 : \delta < 0$. Failing to reject the null hypothesis ($p > 0.05$) indicates the time series has a unit root and a trend.

$$\Delta y_t = \alpha + \zeta t + \delta y_{t-1} + \sum_{i=1}^{K} \beta_i \Delta y_{t-1} + \epsilon_t \tag{1}$$

where Δ is the finite difference operator, the variable of interest is y_t, α is a constant, ζ is the coefficient of the deterministic trend, y_{t-1} is the lagged series, δ is the coefficient of the lagged series, β_i is the coefficient of the lags, and ϵ_t is the residual error.

Non-stationary time-series data exhibit varying statistical properties as shown in Fig. 4. The box and whisker plots of the 3 different linear segments of the BTC prices time series. The time series is divided in 3 segments each segment consisting of 822 days. Each of the segment has different means, standard deviations, maximum and minimum prices as shown in Table 1.

3.1 Data

BTC features and price data are available online and freely accessible. The data for this study were collected from https://bitinfocharts.com by a using web scraper written in Python 3.6. More than 700 features based on technical indicators were collected. From this large feature set, a smaller subset of features was selected through feature selection method. The technical indicators are: Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), Weighted Moving Average (WMA), Standard Deviation (STD), Variance (VAR), Triple Moving Exponential (TRIX) and Rate of Change (ROC). These technical indicators are calculated using different periods such as end-of-day, 7, 30 and 90 days. The end-of-day closing prices are considered as raw values. The raw features, upon which these technical indicators are based, are given in Table 2. The technical indicators show properties that are not readily found in the raw features: things like variances and standard deviations

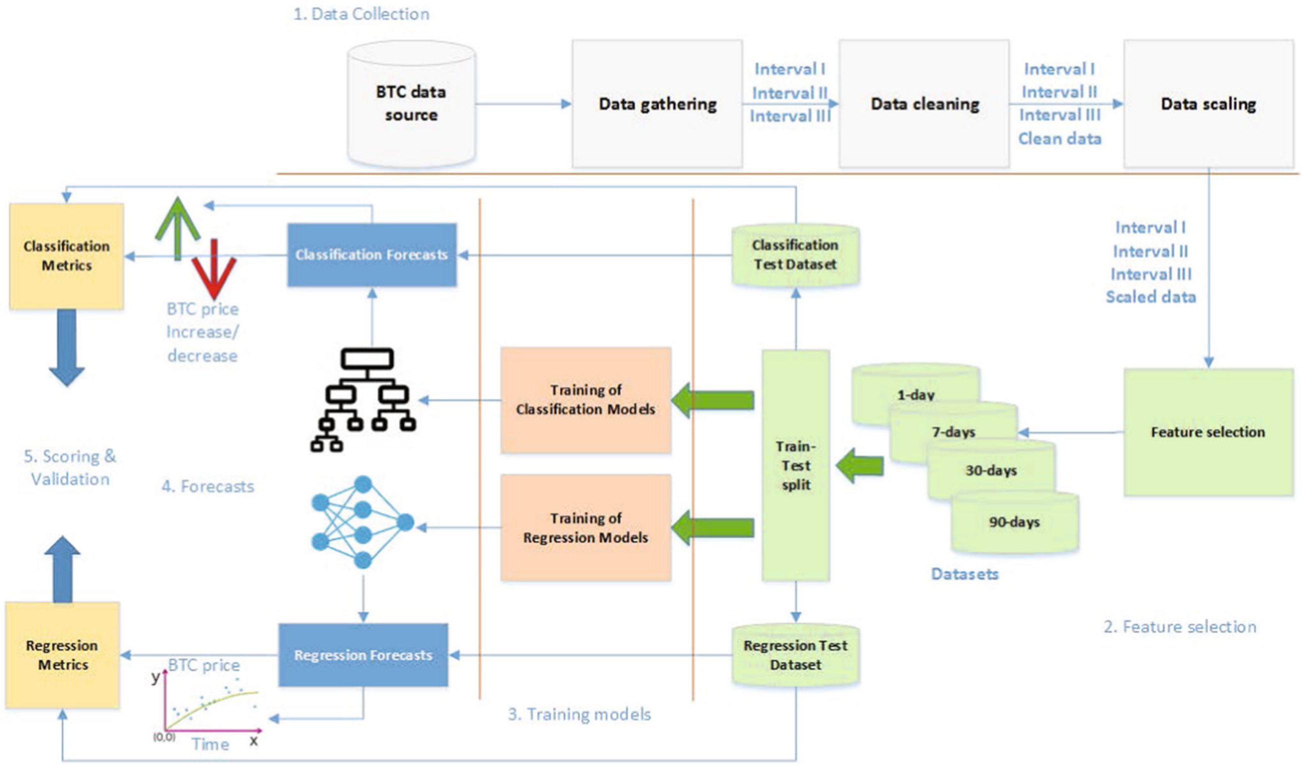


Fig. 3 Detailed steps of ML-based time-series forecast

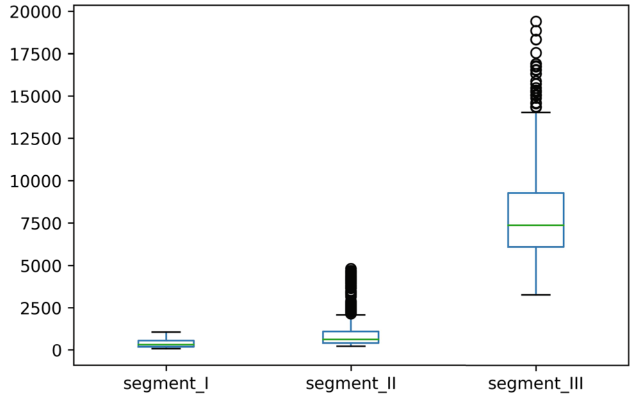


Fig. 4 Boxplots of the different segments of the BTC prices timeline

Table 1 Descriptive statistics show the varying statistical properties of the BTC prices in each segment of the time series

Segment	*n*	Mean	SD	Max	Min
I	822	365	227	1052	66
II	822	1031	1046	4808	214
III	822	7683	2870	19,401	3256

as a function of time. These technical indicators are calculated to show these properties in the BTC price time-series features. For example, they show how the BTC price is related to the standard deviation of the transactions or hashrate in 30-day periods rather than just the raw transactions and hashrates.

In this study, three data intervals were considered for comparing with the state-of-the art given by [10] and [9]. In the first interval, data between April 1, 2013, and July 19, 2016, were considered. The second interval consists of data from April 1, 2013, to April 1, 2017. The third and the largest interval contains data from April 1, 2013, to December 31, 2019. This interval has not been previously studied in the literature.

3.2 Pre-processing

In pre-processing, missing cases were imputed using linear interpolation method wherever possible. Otherwise, most frequently occurring value within the feature is used for imputation. For all the regression models, the dataset was shuffled and split into two sets: training set and validation set. 20% of the data were held for validation, and 80% of the data were used for training. Fivefold cross-validation was applied to the training set for training the stacked artificial neural network. For all the classification models, the dataset was linearly split into two sets: training and validation. The first 80% of the data were assigned to training set, and the last 20% were kept for validation.

Table 2 Raw features from which the technical indicators are created

Features	Description	Features	Description
Transactions	The number of sent and received Bitcoin payments	Median transaction fee	The median of transaction fees in Bitcoin
Block size	Transactional information cryptographically linked in the blockchain. The maximum block size is currently set at 1 megabyte	Average transaction fee	Each transaction can have an associated transaction fee determined by the sender. The transaction fee is received by the miners who verify the transaction. Transactions with higher fees incentivize the Bitcoin miners to process them sooner than transactions with lower fees
Sent from addresses	These are distinct Bitcoin addresses from which payments are made everyday	Block time	The time required to mine one block. Usually, it is around 10 min but can fluctuate depending on the hashrate of the network
Difficulty	The daily average mining difficulty. The difficulty is computed by the network after a specified number of blocks have been created so that the time it requires to mine a block remains around 10 min	Hashrate	The daily total computational capacity of the Bitcoin network. Hashrate indicates the speed of a computer in completing an operation
Average transaction value	The average value of the transactions in Bitcoin	Median transaction value	The median value of the transactions in Bitcoin
Mining profitability	The profitabilityi in USD/day for 1 terahash per second (THash/s)	Active addresses	The number of unique addresses participating in a transaction by either sending or receiving Bitcoins
Sent BTC	The total Bitcoins sent daily	Top 100 to total	The ratio of Bitcoins stored in the top 100 accounts to all the other accounts of Bitcoin
Fee-to-reward ratio	The ratio of the fee sent in a transaction to the reward for verifying that transaction by the other users		

For training ANN, stacked ANN and LSTM models, the features were scaled using the robust scaling followed by minmax scaling method. With the minmax scaling, the features are shifted between 0 and 1 while preserving the relative magnitude of the outliers. The robust scaling method uses the median and the interquartile range to scale the data. The parameters of scaling are fit using the training set and transformed to both training and validation sets. For training SVM, standard scaling was applied to the features as it improved the model performance compared to the other scaling methods based on our data.

For *n*th-day price forecast, the train–test split is the same except that the price column is shifted upward (or equivalently, backward in the time domain) based on the number of days required. For instance, for 7th-day forecasts, the price column is shifted upward by 7 days. This enables the regression models to learn the relation between the features and future prices. For classification models, the price is converted to categorical value by assigning a value of 1 if the price increases or remains the same compared to the previous day. It is assigned a 0 otherwise. For forecasting the *n*th-day price, the same technique is used. For instance, for 30th-day price forecasting, the 30th-day price is compared to today's price and the category 0 or 1 is assigned as appropriate. The effect of price outliers on the performance of the models was studied, and removing them resulted in improved performance.

Isolation Forest method [26] was used for this case. This is an unsupervised method for detecting outliers based on decision forest. It is built based on the assumptions that outliers tend to be few in number and have properties unlike the bulk of the data. For instance, a randomly occurring unusually large spike in BTC price data can be considered an outlier. Removing about 10% of the outliers increased model performance for most of the ML models. A few models performed well despite the outliers.

3.2.1 Feature selection

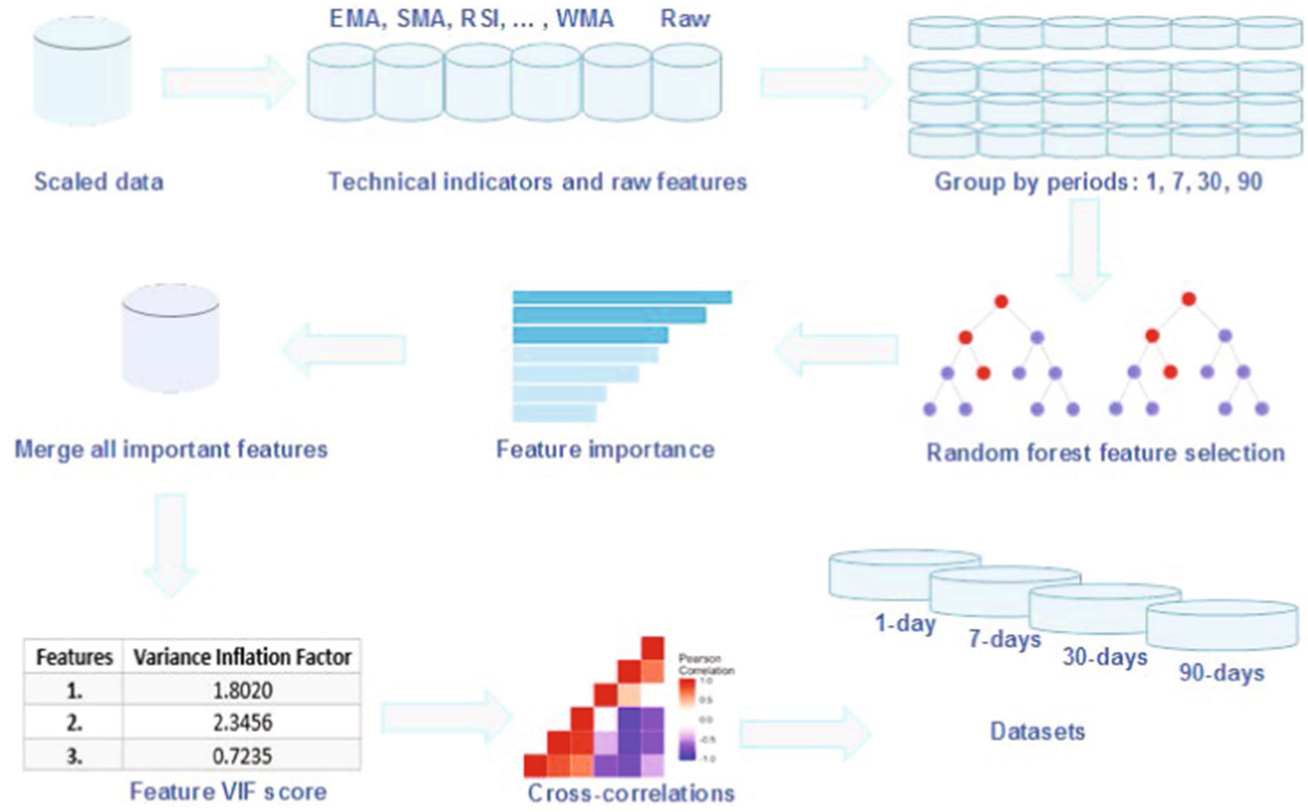
Feature selection, which is a crucial part of data pre-processing, is necessary to improve model performance. The features were extracted and pruned iteratively using a number of different approaches. Firstly, feature importance was determined using an ensemble method based on random decision forest. Secondly, the reduced feature set was checked for multi-collinearity and cross-correlations. Variance inflation factor (VIF) and Pearson correlation were used for these steps. The resulting subset of features were of relatively high importance with low cross-correlation values and no multi-collinearity. The feature selection is repeated for each of the three intervals. When forecasting or predicting for *n*th day, the feature selection process is reiterated to create a new subset of features that are better suited for the period of interest. For instance, the features that can forecast 7-day-ahead price movements fails to forecast 90-days-ahead prices reasonably. Furthermore, feature selection process is required for classification models in each interval after encoding them into categories such as increase, 1, or decrease, 0. Figure 5 shows the feature selection process.

Random forest is an ensemble machine learning method based on decision trees that can be applied to both regression and classification problems. Unlike a single decision tree, a random forest can use hundreds of trees to make forecasts giving better results. It does not require extensive training and is useful for relatively small datasets and for quick evaluation. The features that contributed to the forecast results are given importance scores, which can be inspected for tuning the results such as by keeping or removing those features. Since random forest does not consider multi-collinearity and cross-correlations, other methods need to be used to check for those issues. VIF is used for measuring the collinearity in a multiple regression model. It compares the difference between a model with multiple features and the same model with a lone feature. This indicates the variability that occurs in the model due to having a feature that correlates with another feature present in the model. While VIF $\leq$ 10 can be accepted, some suggest using VIF $\leq$ 5.

Table 3 shows some of the features that are common to several forecast periods based on the feature importance determined by random forest. To elaborate on the feature nomenclature, take the feature label *median_transaction_fee30trxUSD* for instance. This is the 30-day triple moving exponential smoothing of the median transaction fee of BTC given in terms of USD exchange rate at that time. An alternative to manual feature selection is dimensionality reduction by principal component analysis (PCA). In this way, all the features are transformed into a new set of components through matrix manipulations. These new features are linearly independent. A few datasets have been prepared using PCA that captures 95% of the variance in the original dataset.

3.3 Machine learning models

We modeled the Bitcoin prices using different machine learning regression and classification models based on



Features	Variance Inflation Factor
1.	1.8020
2.	2.3456
3.	0.7235

Fig. 5 Feature selection process

Table 3 Some features that are marked as important by random forest across different forecast horizons

Features	EOD	7	30	90
median_transaction_fee30trxUSD	•		•	•
median_transaction_fee7trxUSD	•			•
price90emaUSD	•			•
size90trx	•	•	•	•
transactions	•			•
price30wmaUSD	•		•	
price3wmaUSD	•	•	•	
price7wmaUSD	•	•		
median_transaction_fee7rocUSD	•	•		
difficulty30rsi	•	•	•	•
mining_profitability		•	•	•
price30smaUSD		•	•	
sentinusd90emaUSD		•	•	
transactionvalueUSD		•	•	•
top100cap		•		•
difficulty90mom			•	•
hashrate90var			•	•
price90wmaUSD			•	•
sentinusd90smaUSD			•	•
median_transaction_feeUSD			•	•

EOD refers to end-of-day, and 7, 30 and 90 refer to horizon of forecast

ANN, SANN, SVM and LSTM. These models are explained below.

3.3.1 Artificial neural network

Artificial neural network is a machine learning model that consists of an input layer, an output layer and one or more hidden layers. ANNs are universal function approximators [27] and widely used in machine learning for forecasts and classifications. The ANN model is trained on the training split with hyperparameter tuning for optimal performance. Satisfactory results were obtained using the configuration shown in Table 4 for Interval II. For this model and most other ANNs, the stochastic gradient-based optimizer Adam [28] was used as it performed better in comparison with other gradient-based optimizers on our dataset. Hidden layers, number of neurons per hidden layer, learning rate, epochs and batch sizes were tuned empirically to obtain optimum results. The loss function *logcosh* was used as it is less affected by sparsely distributed large forecast errors than the commonly used mean squared error. The rectified linear unit (ReLU) [29] was used as activation function as it is more robust to the vanishing gradient problem.

Table 4 Configuration of the ANN model used for Interval II

Hyperparameters	Settings
Optimizer	Adam
Hidden layers (neurons)	2 (128,128)
Learning rate	0.08
Epochs	5000
Batch size	64
Activation	ReLU
Loss function	logcosh

3.3.2 Stacked artificial neural network

Multiple ANNs can be used to create an ML model by a technique called stacking. The stacked ANN (SANN) consists of 5 individual ANN models that are used to train a larger ANN. The individual models were trained using the training split with fivefold cross-validation—each model trained on a separate fold. As ANNs are stochastic, each trained model has different weights enabling them to learn their respective fold well. The final ANN learns from these different models, thereby outperforming any individual model over the whole training set. Figure 6 shows the stacked architecture of the SANN regression model. In this figure, the train split is divided into fivefolds. A separate ANN is trained on each of these folds. The output of these ANNs is fed to the final ANN. The final ANN uses the test split to compare the outputs from the smaller ANNs and uses the best output as its input to make forecasts. Although it uses the test split in deciding which output to choose from the smaller input, it does not learn from the test split. The SANN model is different from the ANN model that is trained on the whole train split. The SANN model does not directly learn from the whole train split but rather trains on the outputs of the individual smaller ANNs.

3.3.3 Support vector machines

As a supervised machine learning algorithm, SVM is used for both classification and regression problems. SVM is based on the idea of separating the data points in the training split using hyperplanes such that the distance of separation is maximum. The support vectors are points closest to the hyperplane that are used for calculating its position. SVM kernels can be linear or nonlinear, which includes radial basis function (RBF), hyperbolic tangent and polynomial. For small datasets, SVM can yield forecasts with low error rates without requiring extensive training time. For computing the SVM, either of the objective functions based on L1 (2) or L2 SVM (3) has to

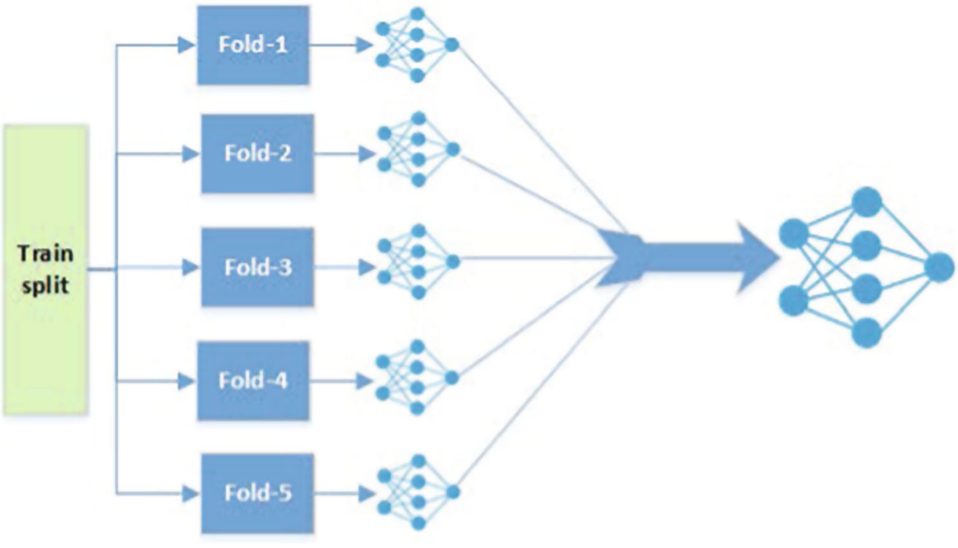


Fig. 6 Architecture of the densely connected SANN wherein separate ANN is trained on unique folds of the trained split. The outputs from the trained sub-models are used as the input for the final model, which uses the best input for making its own forecasts

be minimized subject to the condition given by (4). The Gaussian RBF kernel is given by (5).

$$\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\zeta_i \tag{2}$$

where the slack variable is ζ_i, the penalty is C, and $\mathbf{w}$ is the normal to the hyperplane.

$$\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\zeta_i \tag{3}$$

$$y_i(\mathbf{w}\cdot\phi(x_i) + b \geq 1 - \zeta_i, \quad \text{where } \zeta_i \geq 0 \tag{4}$$

where x_i and y_i are data points, and $\phi(x_i)$ is the data transformation. The offset of the hyperplane from the origin along the normal of the hyperplane, $\mathbf{w}$, is given by $\frac{b}{\|\mathbf{w}\|}$.

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)} \tag{5}$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|^2$ is the square of the Euclidean distance between the features $\mathbf{x}_i$ and $\mathbf{x}_j$, and σ is a free parameter.

3.3.4 Long short-term memory

Long-short term memory (LSTM) network is a type of recurrent neural network that can learn from both long- and short-term dependencies. This deep learning model is particularly useful for modeling and forecasting time-series data. Since the daily Bitcoin price and its features are time-series data, LSTM can be used for making price forecasts and forecasting rise or fall of BTC prices. An LSTM block is analogous to the neuron in the ANN. It has three gates represented by the sigmoid functions: forget (f), input (i) and output (o) gates. In the LSTM block, C_{t-1} is the memory or cell state from the previous block, h_{t-1} is the previous block output, X_t is the vector input, C_t is the cell state or memory of the present block, and h_t is the output of the current block. At the ⊗ junction, the Hadamard product is performed element wise, and likewise at the + junction the summation is done element wise. The LSTM gates and cell states equations are given by (6) to (11).

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{6}$$

where f_t is the activation vector of the forget gate, W and U are the weight matrices, b is the bias vector, and σ_g is the sigmoid function.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{7}$$

where i_t is the action vector of the input or update gate.

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{8}$$

where o_t is the activation vector of the output gate.

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \tag{9}$$

where the activation vector of the cell input is given by c_t and σ_h is the hyperbolic tangent function.

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \tag{10}$$

where c_t is the cell state or memory vector.

$$h_t = o_t \otimes \sigma_h(c_t) \tag{11}$$

where h_t is the output vector of the LSTM block or the hidden state vector.

4 Results

In this section, we present the results of the machine learning-based regression and classification.

4.1 Price forecasts by regression models

To evaluate the performance of regression models, the following metrics are used: mean absolute error (MAE) (12), root mean squared error (RMSE) (13) and mean absolute percentage error (MAPE) (14). A model with low MAE, MAPE and RMSE is desirable. In the context of BTC price, for example, an MAE of 5 means that the predicted price is ± USD 5 from the actual price. MAPE quantifies the error in terms of percentage. For example, a MAPE of 3% can mean either USD 3 or 30 depending on whether the actual price is USD 100 or 1000, respectively. RMSE indicates the spread of the forecast errors. A model that predicts occasionally erratic values will have a higher RMSE value, although it may have still have lower MAE or MAPE. Thus, the models should be evaluated with respect to all the three metrics.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{12}$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2} \tag{13}$$

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{14}$$

The results of the regression models for the three intervals are given in Table 5. In Interval I, from April 2013 to July 2016, the BTC prices did not experience much volatility as shown in Fig. 1. In this interval, all the models performed well with SANN reporting the lowest MAPE of 0.52%. The highest MAPE is reported by the ANN model with 1.88% and an MAE of 4.45. This outperforms [10] in the same interval where their highest performing model has MAPE of 1.91%, RMSE of 15.92 and MAE of 9.63.

Interval II, from April 2013 to April 2017, does have noticeably higher BTC prices toward the end; however, it is relatively stable like Interval I. SANN performed the best with lowest MAPE of 0.93%. Maximum MAPE of 1.98% was reported by LSTM with MAE of 6.55. LSTM reported the highest RMSE of 10.55. In comparison, MAPE of 1.81%, RMSE of 25.47 and MAE of 14.32 were reported by [10] for their best performing model in Interval II.

BTC prices experienced the highest volatility after April 2017, which is covered within Interval III (April 2013 to December 2019). In this interval, SVM reported the lowest MAPE of 1.44%, and ANN reported the highest MAPE of 3.78%. Nevertheless, ANN performed better than the SVM in terms of RMSE and MAE by scoring 74.10 and 39.50, respectively. While LSTM reported a lower MAPE than ANN, it had the highest RMSE and MAE. Stacked ANN reported a MAPE of 2.73, coming in second. It has MAE and RMSE comparable to those of the ANNs. Overall, all the four types of ML models showed robust performance in Intervals I and II, and satisfactory performance in Interval III, albeit with relatively higher errors.

Table 6 summarizes the forecast of the regression models for *n*th-day BTC price considering Interval III, from April 2013 to December 2019. The bar chart in Fig. 7 shows the performance of the ML models in terms of MAPE. SANN reports the lowest MAPE for *n*th-day price forecasts, except for SVM, which gives a lower error rate for end-of-day closing price forecasts. However, this should be evaluated considering the fluctuations of the models as shown in Figs. 8 and 9—where LSTM clearly outperforms all other models.

For 7th-day price forecast, ANN model reported the lowest RMSE of 31.78. Highest MAPE and MAE are reported by the LSTM model, and the highest RMSE is

Table 5 Performance of regression models in different intervals for predicting the daily closing price

Metrics	Intervals	ANN	SVM	SANN	LSTM
MAE	I	4.45	1.72	1.24	2.20
	II	4.61	5.23	4.13	6.55
	III	39.50	47.04	40.58	62.90
RMSE	I	6.13	2.37	1.58	3.01
	II	8.22	9.68	6.48	10.55
	III	74.10	122.34	87.62	135.76
MAPE	I	1.88	0.73	0.52	0.93
	II	1.27	1.23	0.93	1.98
	III	3.78	1.44	2.73	3.61

Table 6 Predicting BTC price by regression models for nth day using Interval III

Metrics	Horizon	ANN	SVM	SANN	LSTM
MAE	7	17.71	19.15	16.32	22.05
	30	90.12	87.43	77.12	116.37
	90	96.01	98.03	72.23	109.22
RMSE	7	31.78	37.32	36.33	36.12
	30	175.60	158.60	156.30	219.59
	90	210.09	203.11	140.00	217.84
MAPE	7	3.09	3.29	2.88	3.83
	30	4.83	5.50	3.45	5.96
	90	4.44	4.96	4.10	5.41

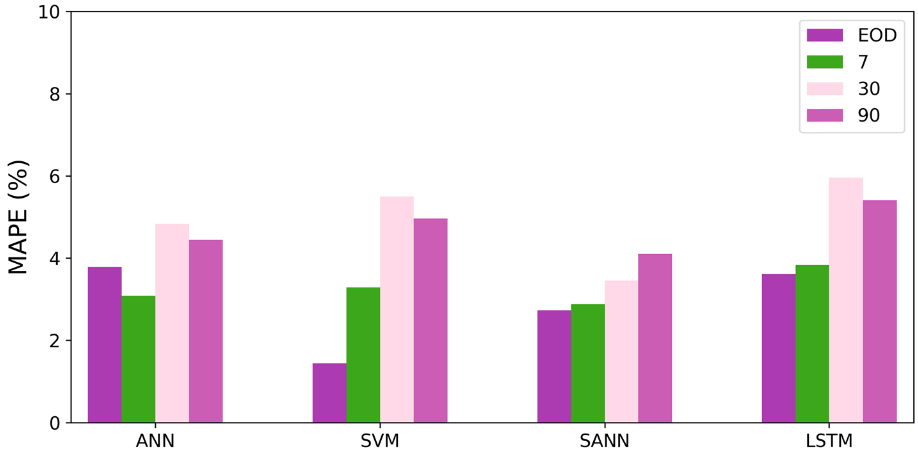


Fig. 7 MAPE of the regression models for *n*th-day forecasts in Interval III

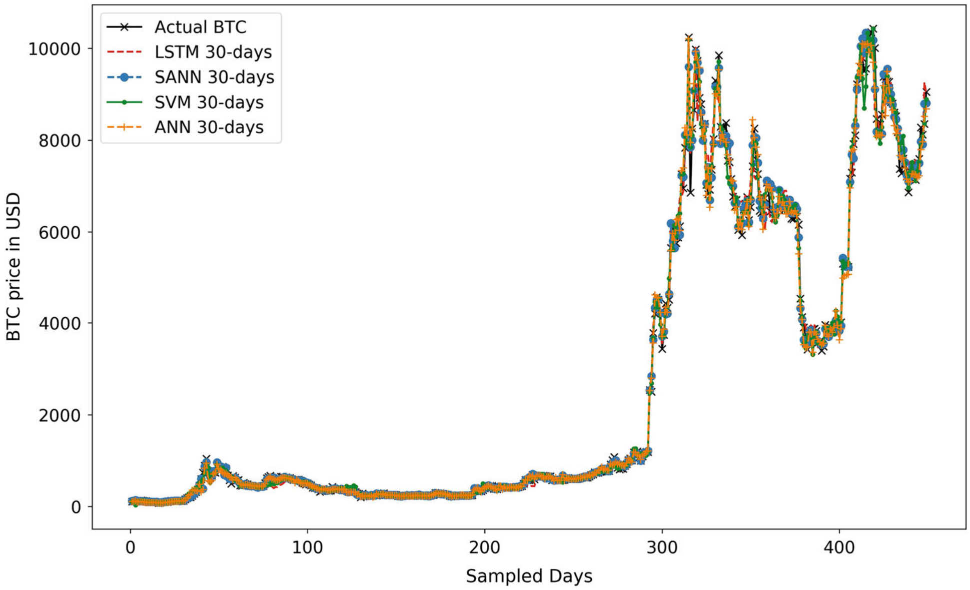


Fig. 8 Performance of LSTM, SANN, SVM and ANN regression models based on 30-days horizon of forecasting. The figure shows that all the models are quite close to each other and follow the BTC prices closely

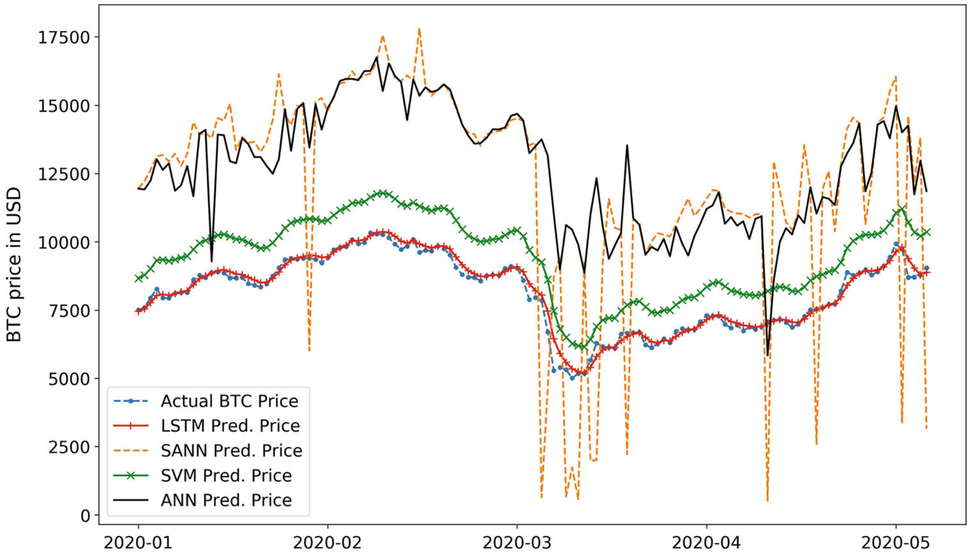


Fig. 9 The trained models based on Interval III are used to forecast the BTC prices for data after December 2019—which is excluded from the train and test sets. It shows LSTM performs the best, followed by SVM. ANN and SANN have similar patterns, although SANN has more fluctuations

reported by SVM. SANN has the lowest MAPE of 2.88% and the lowest MAE of 16.22.

The performance of the models dropped in 30th-day forecasts. In this forecast horizon, SANN reported the lowest error rate of 3.45% with lowest RMSE and MAE of 156.30 and 77.12, respectively. The highest errors were reported by LSTM with RMSE of 219.59, MAE of 116.37 and MAPE of 5.96.

Lastly, in 90th day forecast horizon, SANN performed the best with MAPE, RMSE and MAE of 4.10%, 140.00 and 72.23, respectively. LSTM reported the highest error rate with MAPE of 5.41%. Generally, the SANN model reported the lowest errors for this horizon of forecast, followed by ANN, SVM and LSTM, respectively. However, when considering the model fluctuation as shown in Figs. 8 and 9, LSTM performs the best, followed by SVM. ANN and SANN have similar patterns; however, SANN has high fluctuations. Consequently, even though SANN reported lower mean errors, it is the lowest performing model when considering the variability of its forecasts.

The regression models perform better than baseline price estimates calculated by moving averages and technical indicators. Table 7 shows the MAPE obtained by using moving averages against the MAPE of the ML models. In forecast of end-of-day closing price and short-term horizon of 7 days, the baseline estimate is competitive and comparable to some of our ML models. However, for medium-term horizon forecasts of 30 to 90 days, all developed ML models outperform the baseline.

4.2 Price increase/decrease forecast by classification models

The classification models require different performance metrics for evaluation. The metrics are accuracy (15), F-1 score (16), area under curve (AUC) and the receiver operating characteristic (ROC) curve. The ROC curve is plotted with recall (18) along the y-axis and specificity (19) along the x-axis. All these metrics are created based on true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) shown in the confusion matrix in Table 8. The accuracy is the most commonly reported classification metric and easily interpreted—a higher accuracy means a superior model. However, when the reported classes are imbalanced [30], such as dataset with more days of decreased price than increased ones, metrics such as F1-score may provide further insight. A higher F1-score indicates that the model performs both the precision (17) and the recall (18) well. AUC score indicates how good the model is in distinguishing between the true positives and the true negatives, AUC of 0.5 means no discrimination between classes, and thus, the closer the AUC to one, the better the classification performance [31].

Table 7 The baseline MAPE based on moving averages versus the regression models MAPE

Horizon	Baseline MAPE	Regression Models MAPE
EOD	2.32	1.44 (SVM)
7	3.287	2.88 (SANN)
30	8.36	3.45 (SANN)
90	14.13	4.10 (SANN)

Table 8 Confusion matrix

Actual price	Predicted price	
	Decrease	Increase
Decrease	TN	FP
Increase	FN	TP

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{15}$$

$$\text{F1} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

where precision and recall are given by (17) and (18), respectively.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{17}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{19}$$

Table 9 summarizes the results of classification models for the three intervals. These datasets are different from the regression datasets as they include different set and number

Table 9 Performance of classification models in different intervals

Metrics	Intervals	ANN	SVM	SANN	LSTM
Acc. (%)	I	57	55	60	54
	II	56	62	65	53
	III	53	56	60	54
F1-score	I	0.72	0.67	0.71	0.63
	II	0.69	0.74	0.75	0.58
	III	0.61	0.53	0.60	0.66
AUC	I	0.51	0.51	0.56	0.52
	II	0.50	0.55	0.59	0.53
	III	0.53	0.56	0.60	0.54

of features and are linearly split to make train–test sets, and the target variable is categorical. In Interval I, SANN performs the best in all the three metrics with accuracy of 60% and AUC of 0.56. ANN reported a higher accuracy compared to SVM and LSTM. In Interval II, SANN remains the best performing models with 65% accuracy and AUC of 0.59. F1-score is higher compared to Interval I. SVM comes in second with accuracy of 62%. LSTM reports the lowest accuracy of 53%. [10] reported their best accuracy of 59.45% with AUC of 0.58 using SVM for Interval II, and an accuracy of 62% in Interval I. In Interval III, SVM accuracy drops below 50%. SANN keeps its high classification performance with accuracy of 60% with both AUC and F1-score being 0.60. In comparison, [8] reported 65.3% accuracy with SVM; however, their interval of consideration was from mid-2017 to beginning of 2018 where the trend was generally increasing. In general, the classification accuracy can be improved. However, the extent of improvement remains a challenge since modeling the price increase and decrease by the selected technical features is not adequate. Macroeconomic factors and other unpredictable events affect the fluctuations of BTC price.

The results of the increase/decrease forecasting of *n*th-day BTC price are given in Table 10. Figure 10 shows the classification accuracies for the different types of ML models. For 7th-day forecasts, SVM performs the best with accuracy of 62% and AUC of 0.60. ANN performs the poorest with accuracy of 51%. LSTM performs better than ANN with 55% accuracy and AUC of 0.56.

For 30th day, SANN has the highest accuracy of 62% with AUC of 0.61. SVM, LSTM and ANN have similar accuracy, F1 and AUC scores.

Finally, for 90th-day forecasts, the LSTM model reports the highest accuracy of 64% with AUC of 0.66. The ANN model improves to 62% accuracy. SANN comes in third with 60% accuracy.

Table 10 Forecasting BTC price direction by classification models for *n*th day using Interval III

Metrics	Horizon	ANN	SVM	SANN	LSTM
Acc.(%)	7	51	62	53	55
	30	52	52	62	52
	90	62	54	60	64
F1-score	7	0.65	0.58	0.38	0.65
	30	0.68	0.68	0.70	0.68
	90	0.56	0.66	0.68	0.71
AUC	7	0.53	0.60	0.51	0.56
	30	0.50	0.50	0.61	0.50
	90	0.59	0.57	0.62	0.66

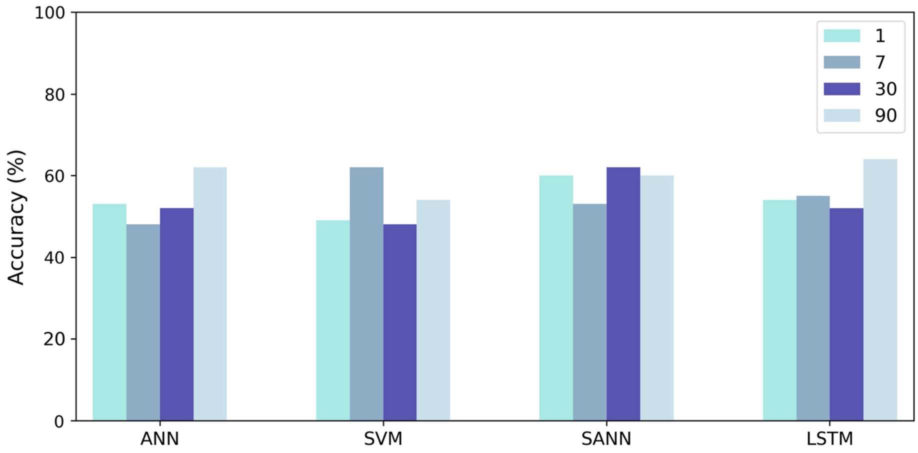
LSTM model performed best for forecasting 9th-day increase/decrease. SANN performed best for next-day forecasts across all intervals as well as 30th-day forecasts. SVM performed best for 7th-day forecasts. ANN had similar performance in Intervals I, II and III next-day forecast but improved in 90th-day forecast. Overall, LSTM model is the best performing one based on Figs. 8, 9 and the overall results from Table 10.

We applied principal component analysis (PCA) for dimensionality reduction for the purpose of measuring its effects. Based on Interval III, the components of PCA that capture 95% (PCA_{95}) of the variance in the original data were used for predicting BTC price and forecasting the increase/decrease. However, the performance of the regression models was subpar compared to the other models reported in Table 5. SVM resulted in MAPE of $\geq 30\%$, ANN in MAPE of $\geq 22\%$, SANN in MAPE $\geq 17\%$, and LSTM in MAPE $\geq 41\%$. In classification, LSTM and ANN models reported accuracy scores below 50%. However, SANN reported accuracy of 61%, and F1-score and AUC of 0.61. SVM reported 54% accuracy with F1-score of 0.57 and AUC of 0.54. Thus, while all the regression models did not perform well using PCA_{95}, the classification metrics showed that SANN and SVM are quite comparable to the models reported in Table 9.

5 Discussion

Modeling BTC price consists of two components: the rise and fall of the price and the actual price. Through this paper, we have shown that the latter can be done with very low error rates. However, the former is still an open challenge to all researchers. As noted in the literature, researchers have used internal and external factors to classify the increase/decrease of BTC price. BTC prices are stochastic, and no given sets of features can provide a complete forecast. Nevertheless, researchers have shown success to various degree in modeling BTC prices based on a different kinds of feature sets. In this paper, we have included features that are directly associated with the blockchain. For instance, if a lot of miners are interested in mining BTC, the hashrate and difficulty will be high. Likewise, if many people are using it for transactions, then the related features such as active addresses and number of transactions will be high. All these features can also be considered time-series features. The technical indicators are simple mathematical tools to convert these rapidly raw features into smoother time-series features that can used to make baseline estimates. Combining the technical

Fig. 10 Accuracy of the classification models for *n*th day forecast in Interval III



indicators for different time periods creates a large feature set that is suitable for machine learning.

The feature selection process has to be robust to find the most useful features. The selected features for the various intervals and forecast horizons are different and not unique to one particular interval or horizon period. The feature selection process presented is systematic and can be used to come up with good selections as evidence by the performance of the models. Alternatively, dimensionality reduction was experimented using the PCA method. The regression models based on PCA did not perform as well as the models trained on selected features. The reason our approach works well is that the techniques used in our feature selection process take care of the issues such as multi-collinearity and cross-correlations in addition to obtaining the feature importance. Although PCA has a similar effect of making new variables that are linearly independent, our inclusion of feature importance with random forest allows us to identify individual features with high importance, which was not possible to do with PCA.

The four ML models used are of different nature and different strengths are weaknesses. SVM is easy to train, but it is not truly stochastic. For a given dataset, it will always produce the same results with the same parameters. It is fast and can be used for small datasets. The reason why SVM performed better than ANN in some instances can be attributed to the size of the dataset. ANN performs generally performs with large datasets containing millions of data points. LSTM is designed to remember trends in the data such as past behavior. Its best performance in 90th-day forecast can be attributed to its design. SANN is a stacked model consisting of sub-models made from smaller ANN models. Considering the different performance metrics and fluctuations of the forecasts, LSTM performed the best overall, followed by SVM. SANN and ANN models can follow the BTC time series with more fluctuations. LSTM also performed the best in classification. This aligns with the results from [25].

6 Conclusion

In this paper, we address short-term to mid-term BTC price forecasts using ML models. It is the first study that takes into consideration all the price indicators up to December 31, 2019, and provides highly accurate end-of-day, short-term (7 days) and mid-term (30 and 90 days) BTC price forecasts using machine learning. Four types of ML models have been used: ANN, SANN, SVM and LSTM. The LSTM showed the best overall performance. All the developed models are satisfactory and have good performance, with the classification models scoring up to 65% accuracy for next-day forecast and scoring from 62% to 64% accuracy for seventh–ninetieth-day forecast. For daily price forecast, the MAPE is as low as 1.44%, while it varies from 2.88% to 4.10% for horizons of seven to ninety days. Performance evaluation results show an improvement over the latest literature in daily closing price forecast and price increase/decrease forecasting. The results are satisfactory and show potential for further applications in different areas such financial technology, blockchain and AI development.

Our results show that it is possible to forecast the actual BTC price with very low error rates, while it is much harder to forecast its rise and fall. The classification model performance scores presented are the best in the literature. Having said that, the classification models for Bitcoin need to be further studied. As further work, hourly BTC prices and technical indicators may be utilized as well as using ensemble models that combined different types of models for making forecast.

Further work which can be followed on the basis of this paper is investigating the use of artificial intelligence for modeling the price of cryptocurrencies as a basis for measuring the risk factor for the financial usage of blockchain technology. This model could also be useful in detecting fraudulent activities and anomalous behavior. When the actual behavior (price) changes significantly from the modeled behavior, this may indicate the effect of external factors such as major global events as well as fraudulent activities such as artificial pumps and dumps. While the price modeling and forecast is not the only tool to detect such external factors, one of the possible applications of such models is in the detection and prevention of fraudulent activities. Our future research will be focusing on such application areas. Using external data inputs related to global events and global financial risk, a combination of machine learning-based price models and anomaly detection methods may be utilized to assess and predict the stability of cryptocurrencies.

Funding This publication was made possible by NPRP Grant NPRP10-0125-170250 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Availability of data and material Data available at: https://github.com/heliphix/btc_data

References

1. Xu W, Cooper A (2017) Digital spillover: measuring the true impact of the digital economy. Huawei and Oxford Economics. https://www.huawei.com/minisite/gci/en/digital-spillover/files/gci-digital-spillover.pdf. Accessed 2 July 2020
2. Oliver J (2013) Mastering blockchain distributed ledger technology, and smart contracts explained-packt publishing, vol 53. Packt Publishing Ltd, https://doi.org/10.1017/CBO9781107415324.004, arXiv:1011.1669v3
3. Unal D, Hammoudeh M, Kiraz MS (2020) Policy specification and verification for blockchain and smart contracts in 5G networks. ICT Express 6(1):43–47. https://doi.org/10.1016/j.icte.2019.07.002
4. Morris DZ (2017) Bitcoin hits a new record high, but stops short of \$20,000 | Fortune. https://fortune.com/2017/12/17/bitcoin-record-high-short-of-20000/
5. Shukla S, Dave S (2020) Bitcoin beats coronavirus blues. https://economictimes.indiatimes.com/markets/stocks/news/bitcoin-beats-coronavirus-blues/articleshow/75049718.cms
6. Liu L (2019) Are Bitcon returns predictable?: Evidence from technical indicators. Physica A: Stat Mech Appl 533:121950. https://doi.org/10.1016/j.physa.2019.121950
7. Huang JZ, Huang W, Ni J (2018) Predicting Bitcoin returns using high-dimensional technical indicators. J Finance Data Sci 5(3):140–155. https://doi.org/10.1016/j.jfds.2018.10.001
8. Chen Z, Li C, Sun W (2020) Bitcoin price prediction using machine learning: an approach to sample dimension engineering. J Comput Appl Math 365:112395
9. Adcock R, Gradojevic N (2019) Non-fundamental, non-parametric Bitcoin forecasting. Physica A: Stat Mech Appl 531:121727. https://doi.org/10.1016/j.physa.2019.121727
10. Mallqui DC, Fernandes RA (2019) Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. Appl Soft Comput 75:596–606
11. Urquhart A (2018) What causes the attention of Bitcoin? Econ Lett 166:40–44
12. Philippas D, Rjiba H, Guesmi K, Goutte S (2019) Media attention and Bitcoin prices. Finance Res Lett 30:37–43
13. Abraham J, Higdon D, Nelson J, Ibarra J (2018) Cryptocurrency price prediction using tweet volumes and sentiment analysis. SMU Data Sci Rev 1(3):1
14. Huang JZ, Huang W, Ni J (2019) Predicting Bitcoin returns using high-dimensional technical indicators. J Finance Data Sci 5(3):140–155
15. Shen D, Urquhart A, Wang P (2019) Does twitter predict Bitcoin? Econ Lett 174:118–122
16. Kim YB, Kim JG, Kim W, Im JH, Kim TH, Kang SJ, Kim CH (2016) Predicting fluctuations in cryptocurrency transactions based on user comments and replies. PloS one 11(8):e0161197. https://doi.org/10.1371/journal.pone.0161197
17. Hyndman RJ, Athanasopoulos G (2018) Forecasting: principles and practice, 2nd edn. OTexts, Melbourne, Australia. https://otexts.com/fpp2/. Accessed 2 July 2020
18. Bakar NA, Rosbi S (2017) Autoregressive integrated moving average (ARIMA) model for fore-casting cryptocurrency exchange rate in high volatilityenvironment: A new insight of bitcoin transaction. Int J Adv Eng Res Sci 4(11):130–137. https://doi.org/10.22161/ijaers.4.11.20
19. Garcia RC, Contreras J, Van Akkeren M, Garcia JBC (2005) A GARCH forecasting model to predict day-ahead electricity prices. IEEE Trans Power Syst 20(2):867–874
20. Munim ZH, Shakil MH, Alon I (2019) Next-day Bitcoin price forecast. J Risk Financ Manag 12(2):103. https://doi.org/10.3390/jrfm12020103
21. Jang H, Lee J (2017) An empirical study on modeling and prediction of Bitcoin prices with bayesian neural networks based on blockchain information. IEEE Access 6:5427–5437
22. Shah D, Zhang K (2014) Bayesian regression and Bitcoin. In: 2014 52nd annual Allerton conference on communication, control, and computing (Allerton), IEEE, pp 409–414
23. Chen GH, Nikolov S, Shah D (2013) A latent source model for nonparametric time series classification. In: Advances in neural information processing systems, pp 1088–1096
24. Greaves A, Au B (2015) Using the Bitcoin transaction graph to predict the price of Bitcoin. https://doi.org/10.1109/CEC.2010.5586007
25. McNally S, Roche J, Caton S (2018) Predicting the price of Bitcoin using machine learning. In: Proceedings—26th Euromicro international conference on parallel, distributed, and network-based processing, PDP 2018 pp 339–343. https://doi.org/10.1109/PDP2018.2018.00060
26. Liu FT, Ting KM, Zhou ZH (2012) Isolation-based anomaly detection. ACM Trans Knowl Discov Data 6(1):1–39. https://doi.org/10.1145/2133360.2133363
27. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2(4):303–314. https://doi.org/10.1007/BF02551274
28. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
29. Hanin B, Sellke M (2017) Approximating continuous functions by ReLU nets of minimal width. arXiv:1710.11278
30. Patel H, Rajput DS, Reddy GT, Iwendi C, Bashir AK, Jo O (2020) A review on classification of imbalanced data for wireless sensor networks. Int J Distrib Sens Netw 16(4):1550147720916404. https://doi.org/10.1177/1550147720916404
31. Mandrekar JN (2010) Receiver operating characteristic curve in diagnostic test assessment. J Thorac Oncol 5(9):1315–1316

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