Journal of Industrial and Management Optimization
Vol. 20, No. 2, February 2024, pp. 570-588
doi:10.3934/jimo.2023091

BITCOIN PRICE PREDICTION USING LSTM, GRU AND HYBRID LSTM-GRU WITH BAYESIAN OPTIMIZATION, RANDOM SEARCH, AND GRID SEARCH FOR THE NEXT DAYS

I. Sibel Kervanci[1], M. Fatih Akay[2] and Eren Özceylan[*3]

[1]IT Department, Gaziantep University, Gaziantep, 27300, Turkey

[2]Computer Engineering Department, Çukurova University, Adana, 01250, Turkey

[3]Industrial Engineering Department, Gaziantep University, Gaziantep, 27300, Turkey

(Communicated by Gerhard-Wilhelm Weber)

Abstract. Bitcoin has high price fluctuations, which involve high risks and high return rates for investors. These high earnings have attracted the attention of investors. This paper proposes a new model for Bitcoin price prediction that effectively reduces prediction error. Hyperparameter optimization methods such as Bayesian optimization (BO), random search and grid search with Long Short-Term Memory (LSTM), Gated Repetitive Unit (GRU), and hybrid LSTM-GRU utilised. Models with BO achieved better results than others. To improve each model's results with BO; Gradient Incremental Regression Trees (GBRT), Gaussian Process (GP), Random Forest (RF) and Extra Trees (ET) were applied to optimizers and corresponding surrogate functions. Evaluating the effects of hyper-parameter values on the problem for each method contributes to the parameter selection process for similar prediction problems. To increase comparability in the literature, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE) and Mean Square Error (MSE) were used. There is a least one hyper-parameter combination, which produces a result close to the best value for each model when the results obtained from the experiments are interpreted. BO with hybrid LSTM-GRU outperformed all methods in this paper and the examined literature for the value of RMSE, MSE, and MAE.

1. Introduction.

Bitcoin is the first decentralized digital cryptocurrency system [42] and constitutes 40% of the approximately $797 billion crypto market (December 2022). Unlike legal money, there is no central authority to issue new money or verify money transfers in the bitcoin system. Both of these tasks are accomplished through the collaboration of miners in the bitcoin network. Due to its speculative nature, it is seen as an investment tool where high risks can be turned into profits for investors. As the popularity of Bitcoin increases, it has also become a popular topic in academic research, and many articles have been written about it. Granger and Wang's tests for severe causality in [33], as a result, they found a bidirectional

2020 *Mathematics Subject Classification.* Primary: 90C31, 68T07; Secondary: 60G25.

Key words and phrases. LSTM; GRU, grid search, random search, Bayesian optimization, Bitcoin, hybrid.

*Corresponding author: Eren Özceylan.

Granger-causal only under extreme situations with crude oil and Bitcoin. [34] found that little correlation between Bitcoin with gold and the U.S. dollar. So they figured out that Bitcoin investment is more like speculation due to the uncertainty of pricing dynamics. [6] concluded that Bitcoin is not isolated from other investment instruments, and to better understand the relationship between bitcoin and investment instruments, the bull and bear market periods should be examined separately . Regarding social media data, as mentioned in [56], different results come out because it consists of data collected from various sources and at different times. When causality tests are applied in a specific time interval, the causality found is valid for the entire time interval. However, in practice, investors generally prefer to keep different investment instruments in different periods [33]. 2022 has been a bad year for Bitcoin when investment instruments increased rapidly in the face of global inflation. The collapse of FTX, one of the crypto exchanges, mentions that the downward trend for cryptocurrencies can have a contagious effect as in traditional markets [10]. After the unexpected crash of the FTX exchange than BlockFi filed for bankruptcy on 28 November 2022; since then, there has been no sudden decrease. Small increases are seen in the market, where the thought that this is overcome. Low prices for Bitcoin, which has a cyclical cycle between extreme bullishness and extreme fall, can be perceived as a buying opportunity for crypto investors. For investors, it is a crucial question whether Bitcoin's bullish curve period has begun. Better predictions can be made through price prediction model selection, optimization, and incorporation of different data sources into the dataset to provide a reference for investors to avoid risks. This paper deals with model selection and optimization because of causal implications and periodic connections. From the comprehensive review article on Bitcoin price prediction, they concluded that machine learning (ML) algorithms play an important role and that the accuracy of the prediction largely depends on the input characteristics and the ML techniques used [28]. The review articles [26] and [28] LSTM and GRU are so popular estimation approaches, and for this reason, they were chosen for hyper-parameters optimization. The performance is increased by the hybrid models [3]. So hybrid models of GRU and LSTM were added to the paper. There are many methods and libraries for hyper-parameter optimization.The relationship between hyperparameters and their values that affect the performance of ML algorithms is unclear. To understand this relationship, it is necessary to adjust the hyperparameters and their values, and compare the performance of the results of different combinations of values. This paper applied grid search, random search, and BO methods to LSTM, GRU, and hybrid LSTM-GRU (GRU-LSTM) models for the Bitcoin price prediction problem. Grid and random search require manually entered information for hyper-parameter settings, while BO does not need manual input other than the first configuration. BO adjusts the following parameters, taking into account parameter values that were successful in the past, and these repeats until the desired results are obtained. In addition to the hyper-parameters of the models, the effects of the optimizer and surrogate function of the BO internal parameters on the models were investigated. BO's optimizers, and corresponding surrogate functions GBRT, GP, RF, and ET were applied. We explore the efficiency of BO surrogate functions on different learning methods with pipeline. Using BO will save time and provide opportunities for real-time application.

This paper aims to predict the Bitcoin price with minimum loss. The hourly data set was preferred to support this target. The expected performance could

not be achieved due to the gaps and delays in the data in the trials conducted in shorter periods than the hourly data. While making Bitcoin price prediction, in the first step, using the hourly data set with LSTM, GRU, hybrid LSTM-GRU, and hybrid GRU-LSTM models were run grid search, random search, BO, and BO optimization. The hyperparameters and the model with the best test result obtained from the first step were used in the second step. These hyper-parameters and models (LSTM, GRU, hybrid LSTM-GRU, and hybrid GRU-LSTM) determined in the second step were run and visualized independently from the first step with the daily data set. This is because a better result is obtained than the parameters obtained by optimizing the daily data set.
The contribution of the paper, Bitcoin price prediction problem literature, is below;

1. There are a few articles, that used more than one Optimization technique in the literature, this paper is aimed to provide a broader perspective by using Grid Search, Random Search, BO, and optimized BO.
2. To our knowledge for the Bitcoin price prediction problem, no such study has been published in the literature using an optimal model framework based on the optimal hyper-parameter set provided by the BO optimizer and corresponding surrogate functions RF, GP, ET, and GBRT.
3. Using machine learning algorithms like LSTM, GRU, hybrid LSTM-GRU (GRU-LSTM)
4. It allows comparing previous studies by listing the best results with four different score values.
5. Hyper-parameters were included and evaluated, which yielded the best results for each technique.
6. It contains the best results among the reviewed articles for the three parameters: $MAE = 0.002302$, $MSE = 0.000015$, *and* $RMSE = 0.003269$.
7. While it provides an hourly prediction opportunity for investors, it will also give an overview of the future price with its prediction in a time interval such as 30 days.

The organization of the paper; In Section 2, similar studies of Bitcoin price prediction problems, including hyper-parameter optimization and hybrid structure, are examined. Section 3, contains information about the dataset's acquisition and preparation. Section 4, summarises information about hyper-parameters and their values, coding techniques in python and gives information about error metrics. In Section 4.1 information about the min-max normalization technique. Section 5 Results of 5.1 The Grid search, 5.2 Random search, and 5.3 Bayesian optimization, brief information on techniques, and their results are presented with tables and figures. The grid search, random search, and BO results are evaluated together. The prediction and loss function results for the next 30 were assessed using the daily data set with the parameters of the best result. Section 6, Conclusion of the paper.

2. **Literature review.** There are many articles on Bitcoin price prediction problems comparing GRU and LSTM, of which GRU gives better results than LSTM [14], [50], [21], [4], [45] except [1]. However, few articles perform hyper-parameter optimization using BO, random search, and grid search algorithms to optimize hyper-parameters. Of these, [48], [46], [16] only for a grid search, [30] only for a random search, [8] both grid search and random search, and [11] both BO and grid search were used. [35], [13], [31], [41], [51] used only BO. Bitcoin and Etherium, Support Vector Regression (SVR), Polynomial regression, and K-Nearest Neighbors

(KNN) regression were modelled by adjusting hyper-parameters with grid search [48]. In [29], three hyper-parameters (timesteps, number of LSTM units, dropout) for hyper-parameter optimization were adjusted manually with LSTM. The results found for two different datasets are given graphically. In [11], the data in the dataset was considered to be scarce for neural networks. For this reason, Support Vector Machine (SVM), XGBoost (XGB), Random Forest Classifier (RFC), and Bernoulli Naive Bayes (BNB) were used to train the model. While deciding which hyper-parameters to choose for these models, BO and grid search were used, and after the parameters with the best results were determined, the model was tested with real-world data. They used ARIMA with grid search in reference [46], and the results of the experiment performed by determining the hyper-parameters gave the best values with MSE= 100753.86, MAE=192.76, and RMSE=317.417.All studies related to Bitcoin price prediction for hyperparameter optimizations are shown in Table 1. The dataset was scaled using Min Max Scaling, and KNN regression found the best MSE value '0.00021'. LSTM, GRU, Convolutional Neural Network (CNN), LSTM-CNN, CNN-LSTM, GRU-LSTM, and LSTM-GRU models were applied to the time series datasets, and the hybrid LSTM-GRU gave the best results on all three measurement metrics [59]. [38] used DCN regression and hybrid models for Bitcoin price prediction. The hybrid model includes classification and regression and is better than the DCN regression. [15] used grid search with support vector regression (SVR) and their results only have $MAPE\ metric$; $train - mape = 12.751$, $test - mape = 10.738$. [51] used LSTM, Bi-directional LSTM (BiLSTM), CNN-BiLSTM, and Deep Artificial Neural Network (DANN).Their best results DANN with $MAE = 0.58$, $RMSE = 0.80$, and $MAPE = 0.10$. Grid search and random search are used for hyper-parameter optimization with machine learning models LSTM, GRU, and RNN [8]. The results were evaluated with MAE and the best result was obtained using grid search with GRU, $training = 0.0043$ and $test = 0.0594$. Hyper-parameters in neural networks are activation function, optimizer, Lr, number of neurons in each layer, and network layers. Lr has a significant effect on loss because it determines how many gradients are applied to the weights to reduce the loss faster [57]. A good choice of Lr direction saves the time of training. The Hparam parameters method used in [27] is a tool of the tensorboard and tries all possibilities for the given parameters like grid search. For most of the Bitcoin price prediction problems, was used, the closing price, [49],[46], [21], [13], [3] and some multivariate datasets [9], [43]. In this paper, we used close prices and date-hour information. In general hourly [19] and daily [18], [30] datasets are used in forecasting problems, while more sensitive minute [50], [1] and second datasets are used in real-time forecasting.

Contributions of the paper are the number of optimization algorithms, using a hybrid model, the number of error metrics, and optimizing optimization algorithms, as a result of the literature. Also, evaluating the parameters obtained as a result of diversifying the optimizations with layers and models. Our motivation is to make future predictions with the parameters obtained as a result of optimization. For this purpose, the best-resulted parameters were used for price prediction for the next 30 days.

TABLE 1. The literatures about optimization for the Bitcoin price prediction problem are as follows

Year	Author	Optimizatin methods	DL methods
2018	[48]	Grid Search	Polynomial regression,SVR and KNN regression
2018	[35]	BO	RNN,LSTM
2019	[30]	Random Search	ARIMA, Random Forest,SVM,LSTM, and WaveNets
2019	[16]	Grid Search	LSTM
2021	[8]	Grid Search, Random Search	GRU,RNN and LSTM
2021	[11]	BO, Grid Search	SVM, XGB, RFC and BNB
2022	[13]	BO	XGBoost
2022	[31]	BO	SVR
2022	[41]	BO	LSTM
2022	[51]	BO	LSTM, BiLSTM, CNN-BiLSTM, DANN
2022	[29]	Manuel	LSTM
2022	[46]	Grid Search	ARIMA
2023	[27]	Hparam (Tensorboard tool)	LSTM

3. **Bitcoin dataset.** The datasets are downloaded from https://www.cryptodatadownload.com. First dataset consists of 38372 lines from 15 May 2018 06:00 to 30 September 2022 00:00. The second dataset consists of 2904 lines from 01 Jan 2015 to 13 Dec 2022 with the daily close prices. The second dataset was used to predict the next 30 days with the hyper-parameters and model that produced the best value found as a result of the optimization algorithms. When the data set is reduced to days, hours, minutes, and seconds, the date range of the data set used becomes shorter, respectively. However, the date range in this paper is quite long for an hourly data set. Transformed the price into a time series problem by shifting one hourly with the shift method. The dataset used in this paper is a time series. When the dataset was split into train and test sets, attention was paid to the $shuffle = False$ properties of the train-test-split functions. The experiments were performed in the three virtual machines in the Proxmox virtual environment with an Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz (2 Sockets) 24x2 Core 100 Gb Ram. Selected all three optimization methods $cross - validation = 3$, $train - size = 0.8$, and each method was run ten times in 100 epochs excluding Grid search.

Looking at the Bitcoin price-time graphs in Figure 1 for different start dates causes different results in deep learning algorithms, just like the different perceptions it creates in a person.

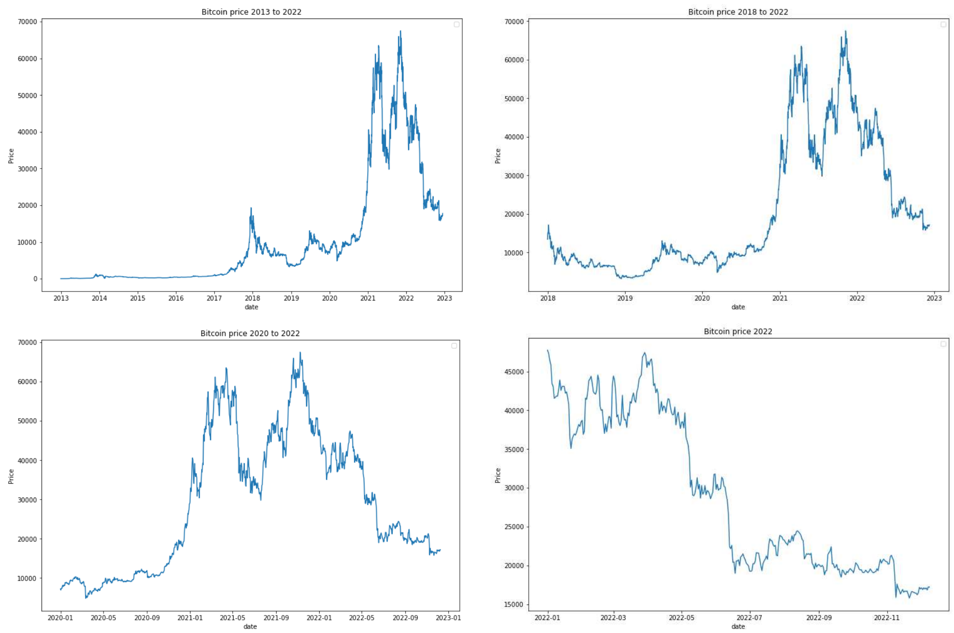

FIGURE 1. Bitcoin prices with different time intervals

4. **Methods.** Deep learning has promising results in many areas likes financial forecasts [14], speech recognition [2], object recognition [25], recommendation [60] and many more. It has also been observed that in the areas where deep learning is applied an enormous amount of the dataset, the number of model parameters, and the optimization of the parameters can significantly increase the accuracy of the predictions [32],[36]. In order to conclude from datasets, machine learning models' parameters' need to be adjusted according to the input values. Some factors affect machine learning algorithms' training process and topology, called parameters. They are separate as parameters and hyper-parameters. Parameters such as weight and bias are updated throughout the learning process, and the algorithm updates the parameters. The parameters mentioned in this paper are hyper-parameters. It is determined before the learning process and these hyper-parameter values affect the learning process. Complex algorithms such as machine learning algorithms and especially deep learning produce very different results with different examples of their hyper-parameters [23]. After experimenting with all variations to understand hyper-parameters' relevance, they showed that the configuration that effects performance depends on a few subsets of hyper-parameters. Given the parameterization of LSTM, GRU generalization performance is largely based on the ability to regularize models sufficiently; for this reason, hyper-parameter optimization is necessary [12]. In addition, the optimization technique used is also important. This is because it is aimed to find global optimum values instead of local optimum values. With this viewpoint, this paper used BO, random search, and grid search optimization methods with machine learning models LSTM, GRU, and hybrid LSTM-GRU. Keras regressor was used for grid search, random search, and BO. The sequential model was used to define layers for LSTM, GRU, and hybrid. In the paper used

Table 2. Hyper-parameters and their values

Activation	relu, tanh, softmax, sigmoid
Optimizer	adam, sgd, rmsprop, adamax
Learning-rate	0.1, 0.05, 0.001, 0.0001
Loss	mean-squared-error
Neuron	32, 64, 128
Dropout-rate	0.1, 0.2

hyper-parameters like an optimizer, activation function, neuron, learning rate (Lr), and dropout rate. The values given to these parameters are as follows in Table 2.

4.1. **Performance metrics.** First, for the performance evaluation to be fair [54], grid search, random search, and BO were all tested using the same parameters and values. While evaluating the combinations in the hyper-parameter probability sets, precise adjustments were made until the best value was obtained with the training dataset. The metric and loss can evaluate the quality of a regression model's prediction. The difference between the predicted and the training datasets is measured by loss functions to evaluate the model's performance. Optimization methods also aim to reduce the error in the loss values. On the other hand, the metrics evaluate the model's performance with a test dataset not seen during the training periods and are calculated from the difference between the estimated and the actual values. However, there is no mapping between models, and loss functions or metrics. The decision of which loss function to choose is made by evaluating the type of problem, model, and dataset [52]. The four metrics used in this paper are given below with their formulas.

$X = predicted\ value$, $Y = real\ value$, $m = number\ of\ value$

$$MAE = \frac{\sum_{j=1}^{m} |y_j - x_j|}{m} \tag{1}$$

$$MAPE = \frac{\sum_{j=1}^{m} (|y_j - x_j|/y_j)}{m} * 100 \tag{2}$$

$$MSE = \frac{\sum_{j=1}^{m} |y_j - x_j|^2}{m} \tag{3}$$

$$RMSE = \sqrt{\frac{\sum_{j=1}^{m} |y_j - x_j|^2}{m}} \tag{4}$$

4.2. **Min-Max normalization.** Setting the dataset at a certain scale increases performance and reduces the difference between different features, improving the model' consistency [15]. Data normalization is critical preprocess of the dataset, especially when dealing with big numerical datasets. In this paper, the Min-Max normalization technique was preferred. It fits the dataset into a predetermined range while preserving the relationship between the original dataset [40] [30].

$The\ Min - Max\ formulation\ is\ as\ follows\ Y_i\ values\ are\ in\ the\ range\ of$ $[0 - 1]$.

A *is the original dataset*, $[B_1, B_2]$ *predefined range*,

$$Y_{i(Min-Max-scaled)} = \frac{A_i - Min(A)}{(Max(A) - Min(A))} * (B_1 - B_2) + B_2 \tag{5}$$

5. Results.

5.1. Grid search.

Grid search is an optimization technique that helps determine the optimum values by comprehensively searching a specific model and selected parameter values. Grid search is simple in terms of implementation, but the experience is required as to which parameter values to choose. It is widely used in practice, but the number of parameters cannot be increased much [58] Because as the number of dimensions (the number of parameters) increases, the efficiency decreases and the processing time considerably increases. According to [15], the most straightforward search technique leads to one of the most accurate estimates, as each parameter combination is run independently of the other. For grid search, the GridSearchCV method is used with KerasRegressor. LSTM, GRU, LSTM-GRU, and GRU-LSTM applied with GridSearchCV and their results at the Table 3. As

Table 3. The best results of MSE with their parameters for grid search (test results)

Models	Neurons	Activation	Dropout	Optimizer	Lr	MSE
LSTM	32	relu	0.1	rmsprop	0.0001	0.143192
GRU	64	sigmoid	0.2	sgd	0.0001	0.242067
LSTM-GRU	128	tanh	0.1	adam	0.001	0.000023
GRU-LSTM	128	tanh	0.1	adam	0.001	**0.000020**

Table 4. Grid search with GRU, LSTM, and hybrid (test results)

Error Metrics	1-GRU	1-Layer LSTM	Hybrid GRU-LSTM	Hybrid LSTM-GRU
MAPE	0.939040	0.662516	**0.007375**	0.009460
MAE	0.458254	0.339161	**0.003268**	0.003820
MSE	0.242067	0.143192	**0.000020**	0.000023
RMSE	0.492003	0.378408	**0.004503**	0.004839

we can see in Tables 3, 4 and Figure 2 for Grid search, LSTM performed better than GRU. In general, hybrid structures gave better results than conventional structures. Hybrid GRU-LSTM gave the best results, but no significant difference exists between GRU-LSTM and LSTM-GRU. Grid search takes much time, as Table 2 has 384 combinations. So cannot compete with random and BO for epoch=100 and requires more than epoch 100. Even $epoch$ = 500, MSE = 0.001202 for 1-layer LSTM is quite far from the expected value. Therefore, only 1-layer GRU, 1-layer LSTM, hybrid LSTM-GRU, and hybrid GRU-LSTM were evaluated.

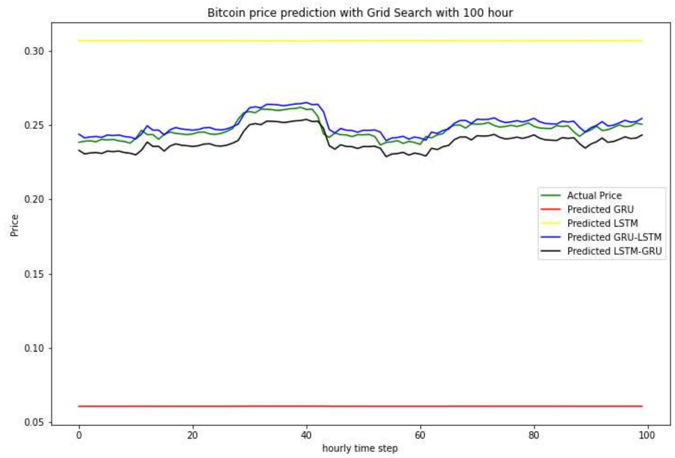

Figure 2. Grid search LSTM, GRU, and hybrid GRU-LSTM results test MSE 100 hours

Table 5. Random search with LSTM and GRU layers (test results)

Error Metrics	1-Layer GRU	1-Layer LSTM	2-Layers GRU	2-Layers LSTM	3-Layers GRU	3-Layers LSTM
MAPE	**0.012479**	0.016798	0.060828	0.043883	0.017961	0.088956
MAE	**0.005431**	0.007045	0.027493	0.017537	0.006493	0.039999
MSE	**0.000040**	0.000061	0.000809	0.000345	0.000081	0.001686
RMSE	**0.006327**	0.007791	0.028445	0.018577	0.009020	0.041065

5.2. **Random search.** Random search is a method that selects random combinations among hyper-parameter combinations, operating fewer resources and shorter execution times than grid search. Random search can reach the global optimum for a large number of values faster than grid search for a global optimization problem. The efficiency of random search is higher in high-dimensional search space than grid search, and the gap closes as the number of parameters decreases [5]. As we can see from Table 5, the best results for GRU and LSTM with 1, 2, and 3 layers worked better in 1 layer, and GRU outperformed LSTM. Table 5 shows a graphically represented in Figure 3 for test MSE values. For the test results in Table 5, the best parameters for each layer are shown in Table 6. The 'tanh' activation function produced the best result for each situation, and the number of neurons 128 is the best result in both LSTM and GRU. Considering Tables 3 and 6 together, $optimizer = adam$ generally fits the dataset. As a result of the proposal, Lr and nonlinearity 'tanh' activation function significantly effect performance [7], [57]. In Table 7, 1 layer LSTM and GRU, which produce the best results in Table 5, are compared with hybrid structures like GRU-LSTM and LSTM-GRU. Hybrid LSTM-GRU produced the best result. Table 7 shows a graphical representation of Figure 3 for test MSE values.

5.3. **Bayesian optimization.** The BO algorithm optimizes hyper-parameters with different combinations in each epoch. The $f(x)$ function is a cost function that requires much time to evaluate in terms of the number of parameter combinations.

TABLE 6. The results of MSE with their best parameters for a random search (test results)

Models	Neurons	Activation	Dropout	Optimizer	Lr	MSE
1-Layer GRU	128	tanh	0.2	adamax	0.05	**0.000040**
2-Layers GRU	32	tanh	0.2	rmsprop	0.001	0.000809
3-Layers GRU	128	tanh	0.2	adam	0.0001	0.000081
1-Layer LSTM	128	tanh	0.1	adam	0.05	0.000061
2-Layers LSTM	64	tanh	0.2	adam	0.001	0.000345
3-Layers LSTM	128	tanh	0.2	rmsprop	0.001	0.001686

TABLE 7. Random search with LSTM, GRU, and hybrids (test results)

Error Metrics	GRU	LSTM	Hybrid GRU-LSTM	Hybrid LSTM-GRU
MAPE	0.012479	0.016798	0.014593	**0.007283**
MAE	0.005431	0.007045	0.006527	**0.003362**
MSE	0.000040	0.000061	0.000056	**0.000021**
RMSE	0.006327	0.007791	0.007506	**0.004565**

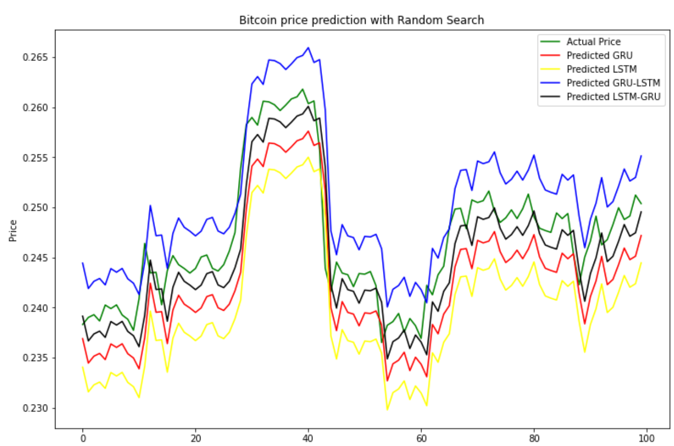

FIGURE 3. Random search LSTM, GRU, and hybrid results test MSE 100 hours

Let X be the set of all combinations of hyper-parameters. It look for the hyper-parameter combination $x*$ that produces the best MSE value among them for the minimum value of f is found in the fewest iterations.

$x*$ is a hyper-parameter set in the X sets, which gives the optimum value of the error rate evaluated with the $f(x)$ function. It reaches better hyper-parameters in less number of epochs by choosing the hyper-parameters that lead to better results in the next epoch, taking into account the results in the previous epoch [44]. It does this by using the two fundamental concepts in BO, the surrogate function and the acquisition function. Instead of the error rate as a function of hyper-parameters, we are trying to optimize the simpler surrogate function applied to black-box optimization. After retrieving the results for several hyper-parameter sets, an acquisition function is defined to select the next hyper-parameter set to use, which tells us there is a candidate with better results. To obtain the objective function $f(x)$ with a surrogate function, it uses an acquisition function that determines which of the elements of set X to try [20]. Utilize instability to stabilize exploration against instability. $Mean : -mean\ [f(x)] \rightarrow mean(x)\ and\ variance : var\ (f(x)) \rightarrow var(x, x^{'})$ $are\ used\ by\ acquisition\ function\ leads\ to\ areas\ where\ improvement\ is\ likely.$ $\varepsilon\ is\ a\ small\ value$; $Prospect\ of\ improvement = Prior\ point\ of\ x_{t-1}\ (f(x_t) < (best\ value\ of\ f(x)\ until\ now) - \varepsilon)\ Gauss\ process\ (GP)$

$$f(x) \sim GP(mean(x), var(x, x^{'})) \tag{6}$$

The proposal's experiment surrogate function is GP with BayesSearchCV class of skopt and the parameter is the base estimator. The different estimators available for BO are; ET, RF, and GBRT estimators in the minimize functions used as surrogate models. While GP uses a probabilistic regression model, RF and GBRT models use an ensemble learning strategy. Although they are similar, there are minor differences between RF and GBRT. The difference between GBRT and RF is that each tree in RF uses a parallel estimator, called bagging, whereas GBRT grows trees sequentially and each tree receives information from ancestral trees.[55]. B is a set of regression trees in the GBRT, s are the regression values from B tree, $|B|$ is number of regression trees,

$$mean(x) = \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} s(\mathbf{x}) \tag{7}$$

$$var(x)^2 = \frac{1}{|\mathcal{B}| - 1} \sum_{s \in \mathcal{B}} (s(\mathbf{x}) - mean(x))^2 \tag{8}$$

When we use RF as the surrogate function for BO, the estimation result produces a single result that falls within the training sample boundaries, rather than a probability distribution. To combine the estimates from the RF, we consider the estimated variances from B different trees as var_b and the mean as $mean_b$ [24].

$$mean(x) = \frac{1}{B} \sum_{B=1}^{B} mean(x)_b \tag{9}$$

$$var(x)^2 = \frac{1}{B} (\sum_{B=1}^{B} var(x)_b{}^2 + mean(x)_b{}^2) - mean(x)^2 \tag{10}$$

ET can be used instead of RF to reduce variance, although this can cause bias. This helps to choose the best one from the set of thresholds for the most distinctive point, providing more randomness with ET modelling for the splitting rule. With ET, we train a set of different decision trees with randomly selected features, while RF trains trees with bootstrap samples for each candidate partition based on a randomly selected subset. While RF uses the bagging procedure to iteratively generate sub-training sets, ET uses all the training examples to construct each tree with a varying number of parameters. Result of this, the number of ET leaves is

Table 8. BO-GP with LSTM and GRU layers (test results)

Error Metrics	1-Layer GRU	1-Layer LSTM	2-Layers GRU	2-Layers LSTM	3-Layers GRU	3-Layers LSTM
MAPE	**0.005526**	0.010038	0.009769	0.007523	0.022224	0.005693
MAE	**0.002562**	0.003921	0.004999	0.003875	0.011213	0.002752
MSE	**0.000015**	0.000025	0.000046	0.000029	0.000168	0.000018
RMSE	**0.003913**	0.005010	0.006785	0.005376	0.012973	0.004234

Table 9. The results of MSE with their best parameters for BO-GP

Models	Neurons	Activation	Dropout	Optimizer	Lr	MSE
1-Layer GRU	128	tanh	0.1	adamax	0.05	**0.000015**
2-Layers GRU	128	tanh	0.1	rmsprop	0.0001	0.000046
3-Layers GRU	128	tanh	0.2	adamax	0.001	0.000168
1-Layer LSTM	128	tanh	0.1	adam	0.05	0.000025
2-Layers LSTM	128	tanh	0.2	adam	0.001	0.000029
3-Layesr LSTM	128	tanh	0.2	adam	0.001	0.000018

larger than RF leaves [17].

$$mean(x) = \frac{1}{B} \sum_{B=1}^{B} mean(x)_b \tag{11}$$

The different estimators for BO are GP, ET, RF, and GBRT were used in the experiments and their effects on the results are shown in Table 12. According to [53], BO can find the optimal value even with a small number of samples. It is also quite efficient when compared to grid search and random search in terms of uptime. While minimizing the $f(x)$ function, a random restart should be used in the code so that it can be tested on parameters other than the hyper-parameters around the local minimum. As a result of the tests, as we can see in Tables 8 and 9, in terms of the number of layers. Both LSTM and GRU give the best results in a layer, and GRU outperformed LSTM. Our experiments, BO-GP with the hybrid LSTM-GRU produced the best test MSE, RMSE, MAE, and MAPE results. Based on [37], the Bayesian optimized CNN-LSTM model produced the best results compared to other models. Hybrid CNN-LSTM structure gave better results than convolutional neural networks. In addition, with the application of BO, the decrease in the number of unnecessary objective function evaluations has increased the efficiency. The same result is valid for the LSTM-GRU we optimized with BO, and there is a very small difference in the MAPE value, while the algorithm produces the best result in the literature in three of the four metrics in both distant and near-time predictions.

In Figure 4, the last 100 predicted data are visualized. Thus, it is aimed at better understanding the results. In Figure 4, the GRU and hybrid LSTM-GRU values overlapped because the results of MSE are the same value, as seen in Table 10. As can be seen at Table 9,the best performing parameters of GRU and LSTM are $activation\ function = tanh$, $optimizer = adam$, $dropout = 0.1$, $neuron = 128$ and $learning\ rate = 0.001$ (epoch=100) for BO. The activation function tanh gives good results with optimizers like adamax, rmsprop, and adam. Can be seen at the Table 10 hybrid LSTM-GRU is better all other combinations. Table 11, Hybrid

TABLE 10. BO-GP with LSTM, GRU, and hybrids

Error Metrics	1-GRU	1-Layer LSTM	Hybrid GRU-LSTM	Hybrid LSTM-GRU
MAPE	0.005526	0.010038	0.006927	**0.005497**
MAE	0.002562	0.003921	0.003083	**0.002302**
MSE	**0.000015**	0.000025	0.000019	**0.000015**
RMSE	0.003913	0.005010	0.004306	**0.003269**

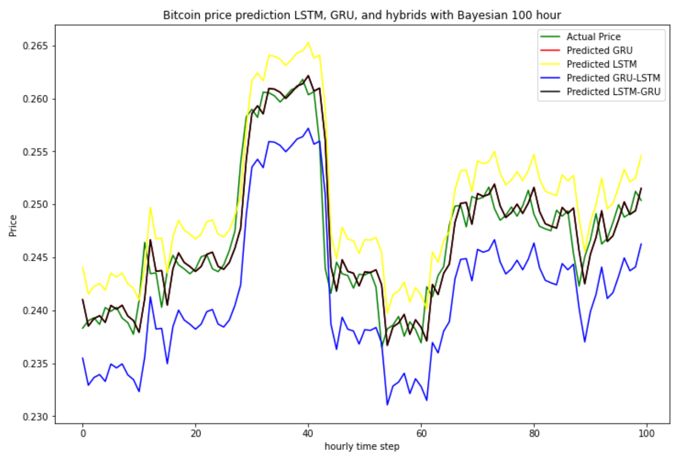

FIGURE 4. BO-GP LSTM, GRU, and hybrid results test MSE 100 hours

TABLE 11. The results of MSE with their best parameters for BO-GP with hybrids

Models	Neurons	Activation	Dropout	Optimizer	Lr	MSE
GRU-LSTM	128	tanh	0.1	adamax	0.05	0.000019
LSTM-GRU	128	tanh	0.1	sgd	0.1	**0.000015**

LSTM-GRU and Hybrid GRU-LSTM best performing parameters are $activation\ function = tanh$,$optimizers = sgd$ and $adamax$, $dropout = 0.1$,$neuron = 128$ and $learning\ rates = 0.05$ and 0.1 (epoch=100).

BO loop for an optimizer represents the steps of a BO. To use this we used a loop mechanism. The various optimisers are provided by the library of Skopt, which uses this class under the base-estimator parameter. A tree-based regression model is improved by sequentially evaluating the next best point using the forest-minimize function. Thus, it reaches the result with the least possible evaluation. Forest-minimize is suitable for ET and RF surrogate models, and experiments have been made in both and the results are given in Table 12.

They used GBRT, BO-GBRT, SVR, RF,Back Propagation (BP) and Multilayer perceptron (MLP) for short-term wind energy forecasting, and combined an efficient optimization framework based on BO-GBRT, providing superior forecast accuracy, computational cost and stability[22]. As seen in Table 12, at least one MAE and

TABLE 12. BO results with different surrogate functions

Estimator	Error Metrics	GRU	LSTM	GRU-LSTM	LSTM-GRU
RF	MAPE	0.011081	**0.007351**	0.020605	0.025902
	MAE	0.004323	**0.003121**	0.009656	0.010337
	MSE	0.000029	**0.000019**	0.000148	0.000125
	RMSE	0.005370	**0.004322**	0.012160	0.011163
GP	MAPE	0.005526	0.010038	0.006927	**0.005497**
	MAE	0.002562	0.003921	0.003083	**0.002302**
	MSE	0.000015	0.000025	0.000019	**0.000015**
	RMSE	0.003913	0.005010	0.004306	**0.003269**
ET	MAPE	0.010253	**0.007933**	0.019982	0.010650
	MAE	0.004329	**0.003491**	0.007872	0.004574
	MSE	0.000028	**0.000022**	0.000079	0.000031
	RMSE	0.005277	**0.004665**	0.008905	0.005584
GBRT	MAPE	0.007655	**0.008401**	0.012354	0.043622
	MAE	0.003686	**0.003657**	0.005489	0.020411
	MSE	0.000025	**0.000023**	0.000042	0.000485
	RMSE	0.004997	**0.004747**	0.006497	0.022014

MAPE for each model take a value of one thousandth. This shows that, when evaluated with the results in Table 13, they are quite acceptable values. In addition, after determining the parameters, gain computational cost and stability thanks to the estimations made with the same parameters for a certain period of time. It is

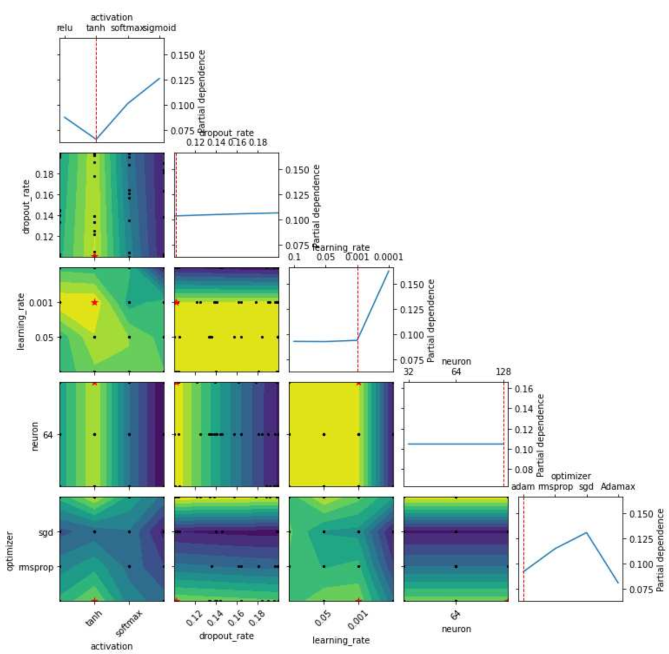

FIGURE 5. Sensitivity analysis of the hyperparameters, LSTM-GRU optimized with BO-GP

very useful to have sensitivity analysis when there are many predictive variables to

Table 13. The results of the literature review comparison of the articles MSE, RMSE, MAE and MAPE values with our best results

References	Method	MSE	RMSE	MAE	MAPE
[1]	4 layers LSTM	-	32.980000	-	-
[13]	XGboost with BO	3273.300000	-	-	-
[41]	LSTM and BO	18155370	4260.912100	-	-
[49]	2 layers LSTM with 10 Fold Cross validation	-	-	0.004300	-
[43]	MLP (learnable window size)	-	-	-	0.310048
[39]	Random Forest	-	0.014100	0.007200	-
[47]	3-layer GRU	63307.312000	251.609400	164.488200	0.003100
Our best result	BO-GP hybrid LSTM-GRU	0.000015	0.003269	0.002302	0.005497

help interpret the models produced by the "black box" estimation method of partial dependency functions. For many hyper-parameters and their values in Figure 5, it would be quite helpful to have relevance to reduce combinations of variables to consider. Since the distribution for the number of neurons shifts towards 128, it is concluded that the number of neurons should be increased, while values less than 0.05 are preferred for Lr instead of values greater than 0.05. Thanks to these interpretations, gain computational cost and stability, it is ensured that the correct parameter values are preferred from unrelated regions to the relevant regions.

When Table 13 is investigated, it is seen that we obtained the best result in the examined literature, our results for MSE, RMSE, and MAE are better except for MAPE. The forecast graph for the next 30 days (14 December 2023 to 12 January 2023 daily) using the best hyper-parameters found with the BO-GP LSTM-GRU predicts that the fluctuating chart of bitcoin will continue in the coming days, Figure 6 predicted value ranges are \$16371<close price<\$21292.

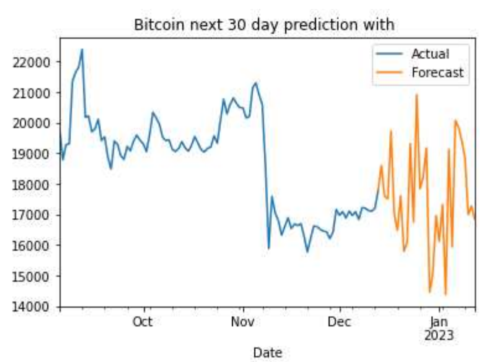

Figure 6. BO LSTM-GRU within next 30 days prediction.

As seen in Figure 7, the loss function increases with the epochs. This is because the value estimated at each step is added to the forecast for the next day, respectively, and the losses grow.

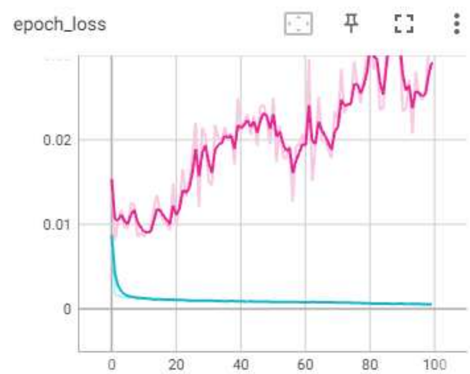

FIGURE 7. Loss of BO LSTM-GRU within next 30 days (training=blue, forecast=pink).

6. **Conclusion.** The points reached according to the results obtained are as follows. In general, dataset size increases when the larger date range is used, and the accuracy increases accordingly. However, due to Bitcoin's extreme volatility, the dataset's variability may also increase. For this reason, it cannot be said that a large dataset for Bitcoin always improves performance. The most appropriate activation function for the proposed time series dataset is 'tanh'. $The f(x) = 2sigmoid(2x)^{-1}$ function is continuous in the range of $[-1, 1]$, differentiable, and its gradient is steeper than the sigmoid function. In addition, the changes are not unidirectional and fit with the normalized data. The parameters affecting Bitcoin are generally unique, although some dependencies have been found in some periods. In general, it is seen that the parameters affecting Bitcoin have temporary effects. We believe in the necessity of utilizing deep learning and optimization methods when predicting the future price of Bitcoin. In this direction, we have obtained very low loss results in our study. Optimizing the surrogate function contributed to the improvement of reducing the loss. Experiments with BO using different surrogate functions such as GP, ET, RF and GBRT showed good interpolation ability in the operating range. But also, the application of the Gaussian process is quite flexible to fit the data and update the posterior distribution. Therefore, GP is considered very suitable for BO, and comparing the results of Table 12 shows that GP achieves quite good results within the four models. It contains the best results among the reviewed articles for BO-GP LSTM-GRU: $MAE = 0.002302$, $MSE = 0.000015$, $and\ RMSE = 0.003269$. Sensitivity analysis is recommended to be used because it provides gain calculation cost and stability, and helps to choose the right parameter values from unrelated regions to relevant regions.

We hope that the paper will give an idea to the investors, of the next 30 days' prediction which obtained an upward with the fluctuating curve.

REFERENCES

[1] A. Aggarwal, I. Gupta, N. Garg and A. Goel, Deep learning approach to determine the impact of socio economic factors on bitcoin price prediction, *Twelfth International Conference on Contemporary Computing (IC3)*, Noida, India, (2019), 1-5.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, et al., Deep speech 2 : End-to-end speech recognition in english and mandarin, *Proceedings of Machine Learning Research (PMLR)*, New York, USA, (2016), 173-182.

[3] S. Aras, Stacking hybrid GARCH models for forecasting Bitcoin volatility, *Expert Systems with Applications*, **174** (2021), 114747.

[4] T. M. Awoke, M. Rout and S. Satapathy, Bitcoin price prediction and analysis using deep learning models, *Communication Software and Networks*, INDIA, (2019), 631-640.

[5] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, *Journal of Machine Learning Research*, **13** (2012), 281-305.

[6] E. Bouri, M. Das, R. Gupta and D. Roubaud, Spillovers between Bitcoin and other assets during bear and bull markets, *Applied Economics*, **50** (2018), 5935-5949.

[7] T. Breuel, Benchmarking of LSTM networks, preprint, 2015, `arXiv:1508.02774`.

[8] N. Buslim, I. L. Rahmetullah, B. A. Setyawan and A. Alamsyah, Comparing bitcoin's prediction model using GRU, RNN, and LSTM by hyperparameter optimization grid search and random search, *9th International Conference on Cyber and IT Service Management (CITSM)*,Bengkulu, Endonezya: IEEE, (2021), 1-6.

[9] S. Cavalli and M. Amoretti, CNN-based multivariate data analysis for bitcoin trend prediction, *Applied Soft Computing*, **101** (2021), 107065.

[10] T. Conlon, S. Corbet and Y. Hu, The collapse of FTX: The end of cryptocurrency's age of innocence, *Available at SSRN*, (2022).

[11] N. Crone, E. Brophy and T. Ward, Exploration of algorithmic trading strategies for the Bitcoin market, preprint, 2021, `arXiv:2110.14936`.

[12] Y. N. Dauphin, H. D. Vries, J. Chung and Y. Bengio, RMSProp and equilibrated adaptive learning rates for non-convex optimization, *Advances in Neural Information Processing Systems (NIPS)*, **28**, (2015).

[13] J. Drahokoupil, Application of the XGBoost algorithm and Bayesian optimization for the Bitcoin price, *Prague University of Economics and Business*, **4** (2022).

[14] A. Dutta, S. Kumar and M. Basu, A gated recurrent unit approach to bitcoin price prediction, *Journal of Risk and Financial Management*, **13** (2020).

[15] I. Fadil, M. A. Helmiawan and Y. Sofiyan, Optimization parameters support vector regression using grid search method, *9th International Conference on Cyber and IT Service Management (CITSM)*,Bengkulu, Indonesia, IEEE, (2021), 1-5.

[16] L. Felizardo, R. Oliveira, E. Del-Moral-Hernandez and F. Cozman, Time series classification of cryptocurrency price trend based on a recurrent LSTM neural network, *Journal of Information Processing Systems*, **15** (2019), 694-706.

[17] P. Geurts, D. Ernst and L. Wehenkel, Extremely randomized trees, *Machine Learning*, **63** (2006), 3-42.

[18] Y. Gong and R. Huser, Asymmetric tail dependence modeling, with application to cryptocurrency market data, *The Annals of Applied Statistics*, **16** (2022), 1822-1847.

[19] N. Gradojevic, D. Kukolj, R. Adcock and V. Djakovic, Forecasting Bitcoin with technical analysis: A not so random forest?, *International Journal of Forecasting*, **1** (2023), 1-17.

[20] R. Grosse, *Bayesian Hyperparameter Optimization*, Toronto University, 2017, Available from: `https://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/slides/lec21.pdf`.

[21] M. J. Hamayel and A. Y. Owda, A novel cryptocurrency price prediction model using GRU, LSTM and bi-LSTM machine learning algorithms, *AI*, **2** (2021), 477-496.

[22] H. Huang, R. Jia, X. Shi, J. Liang and J. Dang, Feature selection and hyper parameters optimization for short-term wind power forecast, *Applied Intelligence*, **51** (2021), 6752-6770.

[23] F. Hutter, H. Hoos and K. Leyton-Brown, An efficient approach for assessing hyperparameter importance, *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, **32** (2014), 754-762.

[24] V. H. Jeroen and J. Vanschoren, Hyperboost: Hyperparameter optimization by Gradient Boosting surrogate models, preprint, 2021, `arXiv:2101.02289`.

[25] S. Kartal, Comparison of semantic segmentation algorithms for the estimation of botanical composition of clover-grass pastures from RGB images, *Ecological Informatics*, **66** (2021), 101467.

[26] I. S. Kervanci and M. F. Akay, Review on Bitcoin price prediction using machine learning and statistical methods, *Sakarya University Journal of Computer and Information Sciences*, **3** (2020), 272-282.

[27] I. S. Kervanci and M. F. Akay, LSTM Hyperparameters optimization with Hparam parameters for Bitcoin Price Prediction, *Sakarya University Journal of Computer and Information Sciences*, **6** (2023), 1-9.

[28] A. M. Khedr, A. Ifra, P. V. P. Raj, M. El-Bannany, M. S Alhashmi and M. Sreedharan, Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. Intelligent Systems in Accounting, *Finance and Management*, **28** (2021), 3-34.

[29] J.-H. Kim and H. Sung, Understanding bitcoin price prediction trends under various hyperparameter configurations, *Computers*, **11** (2022), 167.

[30] D. H. Kwon, J. B. Kim, J. S. Heo, C. M. Kim and Y. H. Han, Comparative study of Bitcoin price prediction using waveNets, recurrent neural networks and other machine learning methods, *6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, (2019), 1-6.

[31] S. Lahmiri, S. Bekiros and F. Bezzina, Complexity analysis and forecasting of variations in cryptocurrency trading volume with support vector regression tuned by Bayesian optimization under different kernels: An empirical comparison from a large dataset, *Expert Systems with Applications*, **209** (2022), 118349.

[32] Q. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow and A. Ng, On optimization methods for deep learning, *ICML'11 Proceedings of the 28th International Conference on International Conference on Machine Learning*, Bellevue, Washington, USA, (2011), 265-272.

[33] D. Li, Y. Hong, L. Wang, P. Xu and Z. Pan, Extreme risk transmission among bitcoin and crude oil markets, *Resources Policy*, **77** (2022), 102761.

[34] X. Liu and B. Li, Safe-haven or speculation? Research on price and risk dynamics of Bitcoin, *Applied Economics Letters*, **77** (2022), 1-7.

[35] S. McNally, J. Roche and S. Caton, Predicting the price of Bitcoin using machine learning, *26th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, Cambridge, United Kingdom, IEEE, (2018), 339-343.

[36] G. Melis, C. Dyer and P. Blunsom, On the state of the art of evaluation in neural language models, preprint, 2018, `arXiv:0707.0078`.

[37] W. Min, L. Xikun and Z. Yi-di, Gun life prediction model based on bayesian optimization CNN-LSTM, *Integrated Ferroelectrics*, **228** (2022), 107-116.

[38] P. K. Nagula and C. Alexakis, A new hybrid machine learning model for predicting the bitcoin (BTC-USD) price, *Journal of Behavioral and Experimental Finance*, **36** (2022), 100741.

[39] G. R. Patra and M. N. Mohanty, Price prediction of cryptocurrency using a multi-layer gated recurrent unit network with multi features, *Computational Economics*, (2022), 1-20.

[40] S. K. Patro and K. K. Sahu, Normalization: A preprocessing stage, preprint, 2015, `arXiv:1503.06462`.

[41] E. S. Pour, H. Jafari, A. Lashgari, E. Rabiee and A. Ahmadisharaf, Cryptocurrency price prediction with neural networks of LSTM and Bayesian optimization, *EJBMR*, **7** (2022), 20-27.

[42] M. Rahouti, K. Xiong and N. Ghani, Bitcoin concepts, threats, and machine-learning security solutions, *IEEE Access*, **0** (2018), 67189-67205.

[43] S. Rajabia, P. Roozkhoshb and N. M. Farima, MLP-based learnable window size for bitcoin price prediction, *Applied Soft Computing*, **129** (2022), 109584.

[44] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, Adapt. Comput. Mach. Learn. MIT Press, Cambridge, MA, 2006.

[45] M. Rizwan, S. Narejo and M. Javed, Bitcoin price prediction using deep learning algorithm, *13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, IEEE, (2019), 1-7.

[46] M. Rostami, M. Bahaghighat and M. M. Zanjireh, Bitcoin daily close price prediction using optimized grid search method, *Acta Universitatis Sapientiae, Informatica*, **13** (2021), 265-287.

[47] M. Saad, J. Choi, D. Nyang, J. Kim and A. Mohaisen, Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions, *IEEE Systems Journal*, **14** (2020), 321-332.

[48] H. Singh and P. Agarwal, Empirical analysis of Bitcoin market volatility using supervised learning approach, *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Noida, India, (2018), 1-5.

[49] S. Tandon, S. Tripathi, P. Saraswat and C. Dabas, Bitcoin price forecasting using LSTM and 10-fold cross validation, *2019 International Conference on Signal Processing and Communication (ICSC)*, IEEE, (2019), 323-328.

[50] P. Thearasak and N. Thanisa, Machine learning models comparison for Bitcoin price prediction, *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, (2018), 506-511.

[51] B. Tripathi and R. K. Sharma, Modeling Bitcoin prices using signal processing methods, Bayesian optimization, and deep neural networks, *Computational Economics*, (2022), 1-27.

[52] Q. Wang, Y. Ma, K. Zhao and Y. Tian, A comprehensive survey of loss functions in machine learning, *Annals of Data Science*, **9** (2022), 187-212.

[53] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei and S.-H. Deng, Hyperparameter optimization for machine learning models based on Bayesian optimization, *Journal of Electronic Science and Technology*, **17** (2019), 26-40.

[54] L. Yang and A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, *Neurocomputing*, **415** (2020), 295-316.

[55] S. Yang, B. Liu, Z. Hong and Z. Zhang, Bayesian optimization-based beam alignment for mmWave MIMO communication systems, *In 2022 IEEE 33rd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, IEEE, (2022), 825-830.

[56] Z. Ye, Y. Wu, H. Chen, Y. Pan and Q. Jiang, A Stacking Ensemble deep learning model for Bitcoin price prediction using twitter comments on Bitcoin, *Mathematics*, **10** (2022), 1307.

[57] C. Yu, X. Qi, H. Ma, X. He, C. Wang and Y. Zhao, LLR: Learning learning rates by LSTM for training neural networks, *Neurocomputing*, **394** (2020), 41-50.

[58] T. Yu and H. Zhu, Hyper-parameter optimization a review of algorithms, preprint, 2020, arXiv:2003.05689.

[59] N. Zafar, I. U. Haq, J. Chughtai and O. Shafiq, Applying hybrid Lstm-Gru model based on heterogeneous data sources for traffic speed prediction in urban areas, *Sensors*, **22** (2022), 33-48.

[60] P. Zhanglin, W. Dequan, W. Anning, L. Xiaonong and M. P. Panos, Deep learning-based recommendation method for top-K tasks in software crowdsourcing systems, *Journal of Industrial and Management Optimization (Jimo)*.

Received December 2022; revised May 2023; early access July 2023.