Time series forecasting of Bitcoin price based on major currencies exchange rates using machine learning models.

Dr. Fahimeh Jafari, PhD FHEA PGCHE MSc BSc
Senior Lecturer
School of Architecture, Computing and Engineering,
University of East London (UEL),
University Way, Dockland Campus.

Roopashree Ramachandraiah
MSc Student
School of Architecture, Computing and Engineering,
University of East London (UEL),
University Way, Dockland Campus.

Abstract:

This paper aims at demonstrating forecasting of bitcoin prices based on forex rates of major currencies using multivariate machine learning-based algorithms like Support Vector Regression, XGBoost, Random Forest and Long Short Term Memory. All these algorithms are based on different machine learning techniques like regression, ensemble boosting and bagging, and Recurrent Neural Networks. These models are built on the data collected from Bloomberg for the period Jan-2017 to Jul-2022 which included over 1836 daily observations of bitcoin price and other currencies like CAD, CNY, GBP, EUR, SGD, NZD, AUD, and JPY exchange rates in USD. The performance of the models is evaluated using different metrics like Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Scaled RMSE. All the developed models exhibited satisfactory results with MAPE of less than 6% after hyperparameter tuning. However, XGBoost performed well compared to the remaining three by reporting MAPE of around 2.5% for all lags and it is followed by SVR, LSTM and Random Forest in terms of performance.

Key Words: Bitcoin forecasting, Time series, Machine Learning, XGBoost, Random Forest, Support Vector Regression, LSTM.

1 INTRODUCTION:

The Time series analysis has led an innovative path in the field of fintech where the major area of study is predicting and forecasting the prices of financial market elements like Forex rates, cryptocurrencies, stocks, etc. It helps organizations to understand the underlying patterns and behaviors of the data over the interval of time. Since the invention of cryptocurrencies, application of time series analysis on historical data of cryptocurrencies for forecasting their prices has developed an extreme interest among researchers.

Cryptocurrency is a major invention in the field of economy and finance that has proven the digital revolution in these fields. Cryptocurrency refers to a virtual or digital form of money that operates using cryptography to secure transactions [1]. Unlike other currencies, cryptocurrencies are decentralized, and their transactions are maintained and recorded in public distributed digital ledgers using a technology called Blockchain. Bitcoin is the first blockchain-based cryptocurrency invented in 2008/9 by Satoshi Nakamoto that gained worldwide attention at the end of 2013[2].

Bitcoin is the most popular cryptocurrency having highest market value and has gained attention from investors and traders. A lot of studies have been done on time series analysis of bitcoin prices by considering individual and combination of various predictors like the forex market, high investment assets like gold, oil, stock markets, sentiment analysis of tweets and bitcoin mentions in news and social media with the primary objective being forecasting its price. For an example, [3] used machine learning algorithm for prediction of various cryptocurrencies like Ether, Monero and Litecoin and [4] used neural network-based algorithms like LSTM, BiLSTM and CNN to forecast by bitcoin price using sentiment analysis to analyze tweets as price determinants or features.

The recent study conducted by Yan claims that the bitcoin exhibits the characteristic of hedging asset [5]. Thus, financial traders and investors are more interested in the returns of these cryptocurrencies. However, determining the exact feature that affects them is highly difficult due to the rapid change in the pattern or trends in their price. This adds the difficulty level for building a model based on a feature. Since the anticipation of future returns or profits of bitcoins depends on the accurate forecast of their prices using the existing information, it is extremely important to build a highly reliable model for this task. This paves the foundation for the problem statement of this dissertation.

Considering the problem statement, the primary objective of the dissertation is to develop machine learning-based multivariate predictive models to forecast bitcoin prices by considering historical time-series exchange rates of major currencies like Great Britain Pounds, Australian Dollars, Chinese Yuan , Euro, Canadian Dollars, New Zealand Dollars, Singapore Dollars and Japanese Yen, along with bitcoin price. The research also aims at determining the lag to be looked back at for efficient forecasting.

Research questions:

Aligning with the objective and contribution of the research, the study aims at addressing the following research questions.

- How accurately the models built in the study can forecast the bitcoin price based on the considered features?
- Which model performs the best in predicting bitcoin price?
- Which lag gives the best results compared to all the lags considered?

2 RELATED WORKS:

Analyzing time-series financial data and predicting the financial features are the most challenging areas. The invention of cryptocurrency like bitcoin in the year 2009 has led to yet another research on its prediction in finance. Though the concept of bitcoin started in the year 2009 and started gaining attention in the year 2013[2], its popularity immensely increased in the year 2017 as it had exponential growth in the financial market [6]. Since then, the Application of data analytics and feature engineering to identify features impacting the bitcoin price and developing forecast models based on machine learning techniques for predicting bitcoin price is a major area of interest for many researchers.

1. Need for bitcoin price prediction:

Since the invention of the bitcoin several studies have been done till recent time to determine its characteristics and see if it can be considered as another type of currency or just an asset [7][8][9]. Most of these studies claim bitcoin is a speculative asset because of its volatility and high fluctuations in its price. The opinion that the bitcoin is an asset attracted the interest of investors and justified the need for bitcoin prediction and has led to further different studies on its features and various price determinants.

2. Approaches used in bitcoin price prediction:

When it comes to regression problems, algorithms based on various approaches are used for building predictive models which include mainly two kinds of algorithms and they are traditional statistical algorithms and machine learning algorithms. Some of the research done using these approaches are as follows.

1) Traditional statistical-based approach:

In 2019 Ozyesil [10], analyzed the relationship between bitcoin price and exchange rates. In his study, he considered only Euro (EUR) and the US dollar (USD) currencies. The analysis was based on Vector Autoregressive (VAR) method using variance decomposition and impulse-Response functions. This research showed that bitcoin and foreign exchange are not substitutes for each other and bitcoin's price and EUR were not affected by the USD rate. Another research by Kormaz in 2018 [11] proved the existence of relationship between gold, EURO and USD returns using supaugmented Dickey-Fuller (SADF) and Generalized SADF. In 2021, Benzekri and Özütler [12] used a statistical-based univariate model - ARIMA (AutoRegressive Integrated Moving Average) for forecasting bitcoin price by using quarterly data for the period 2014Q1 till 2020Q2 and the data from 2020Q3 and 2020Q4 were used for testing the model. The predicted values for 2020Q3 9.08(actual value being 9.27) showed more accuracy compared to 2020Q4 9.09 (actual value 9.71). In 2022, Shakeri et al., [13] analyzed the connection between oil and gold prices and stock market indices and cryptocurrency prices using BEKK multivariate GARCH method and confirmed the mutual relationship among these features. Another study by Saini and Shobana in 2022 [14], compared the accuracy of ARIMA and Facebook prophet for forecasting bitcoin where ARIMA showed an accuracy of 94% compared to prophet 93%.

2) Machine learning and neural network approach:

Much research has been done in recent years to predict bitcoin prices using machine learning algorithms. In 2020, Mudassir et al., analyzed Bitcoin's data for different timeframes like next day, 7th day, 30th day and 90th day by developing classification and regression models based on machine learning and achieved an accuracy of 65% for the next day, 62%-64% for 7th day and 9th days' forecast [15]. Another study by Hamayel & Owda in 2021, explored the prediction of three different cryptocurrencies- bitcoin, Litecoin and Ethereum using GRU(Gated Recurrent Unit), LSTM (Long Short-term Memory) and Bi-LSTM (Bi-directional LSTM) which are based on Recurrent Neural network and found that GRU outperformed compared to other two [16]. In 2021, Livieris et al., proposed Multi-Input Deep neural network (MICDL) model for forecasting bitcoin, Litecoin and ripple by handling and exploiting different cryptocurrency data separately and processing the data using the proposed model which showed reduced overfitting and computational costs [17]. Another study in late 2021 by Nayak et al, used a hybrid ANN+RA (Artificial neural network + Rao algorithm) for predicting six cryptocurrencies and developed six models (GA-Genetic Algorithm + ANN, PSO- Particle Swarm Optimization +ANN, MLP-Multi-Layer Perceptron, SVM-Support Vector Machines, LSE- Least Squared Estimator and Arima for comparison of performance and found that ANN+RA showed least MAPE (Mean Absolute Percentage of Error) and ARV(Average Relative Variance) [18]. In 2022, Patil used machine learning based multi-linear regression and RNN based LSTM to predict bitcoin's price and compared the results and found that though Linear regression performed well with prediction accuracy as close as 99.97, LSTM model's prediction rate was comparatively higher only with a little difference [19]. In 2022, Mittal and Geetha implemented ANN based Gated Recurrent Unit model to predict bitcoin price and found that the model was capable of learning similarities and liabilities of data[20]. The model resulted an RMSE of 1987.1057 and MAPE of 18.4905%. In June 2022, Wiliani et al., linear regression and neural network-based(or ANN) models to predict bitcoin price and found that neural network-based model performed well compared to linear regression[21]. In June 2022, Yan et al., implemented an integrated approach of deep neural network and bagging technique to forecast bitcoin prices [22]. Their study integrated stacking denoising autoencoders(SDAE) with bootstrap (B) aggregation and developed SDAE-B model. The price determinants considered for their study are different bitcoin features like block size, hash rate, number of transactions etc., along with gold price and dollar index. They found that the new model was efficient in prediction with lower errors.

***Gap in the knowledge**:* From the literature review, we can see that most of the studies considered only raw features of the currency like high, low, open, close prices, transaction volume and block size as independent features or the technical indicators derived from these features. Identifying this gap, the project aims at exploring behavior of bitcoin price with respect to forex rates of currencies considered and build machine learning-based models to forecast the bitcoin price.

3 DATA EXTRACTION AND PREPROCESSING:

The data consists of bitcoin prices and other currency rates in USD. The other currencies considered for the research are, Great Britain Pound, Australian dollars, Canadian dollars, Chinese yuan, New Zealand dollars, Euros, Japanese Yen, and Singapore dollars. All the data is extracted from Bloomberg for the period 02-Jan-2017 till 21-July-2022. The entire data has 1836 daily observations for each feature. Bitcoin's time series data did not have any missing values as cryptocurrencies are traded 24/7. However, the forex market is closed on weekends, and hence there were missing values for Saturdays and Sundays in other currencies' data which is handled by substituting the weekend rate with Friday's close value of corresponding currency rates.

After converting the time series data for supervised learning by introducing lags(this study considered lag1, 3 and 7), the resultant input data consisting of 1836 observations are separated into training and testing samples. The first 80% of data having around 1468 observations are used as a training sample. The remaining 20% of data having 367 daily observations are used as test data to test the performance of the model.

4 METHODOLOGY:

The primary goal of the research is to build a predictive model for forecasting bitcoin prices based on the exchange rates of other currencies. All these data are time series data collected at equal intervals of time and the prediction of time series is different from normal supervised prediction because the former one considers time interval as one of the important criteria. The proposed methods for prediction of bitcoin prices are machine learning algorithms like XGBoost, Random Forest, SVR and LSTM. These algorithms are based on different machine learning techniques like tree-based ensemble boosting, tree-based ensemble bagging, regression, and Recurrent Neural Networks. The different lags used are lag1(t-1), lag3(t-1,t-2,t-3) and lag7(t-1,t-2….t-7).

XGBoost:

XGBoost (known as eXtreme Gradient Booting) is an ensemble machine learning which is based on a gradient boosting framework and can be used for both regression as well as classification problems. This algorithm was developed by Tianqi Chen and Carlos Guestrin in the year 2016 [23]. It provides a fast and accurate way of solving data science problems by using the parallel tree boosting technique [23]. The features of XGBoost that contribute to its fast processing are cache optimization, parallelization, and out of memory optimization. The features that contribute to its performance are auto pruning of the tree that helps in maintaining bias variance, regularization which prevents the model from overfitting, and auto handling of missing values.

In boosting principal, the strong learners are produced by combining the weak learners [24]. Here the weak learners are functions that produce results slightly better than chance whereas strong learners are functions with high accuracy.

The important parameters of the algorithm used in the implementation are as follows.

- Regularization parameter: Also known as lambda, is a parameter that helps in controlling the effect of outliers on the prediction and helps in preventing the overfitting of the model.
- Gamma: This is the parameter that helps in auto pruning the tree.
- Eta: This is an analogous to learning rate.
- Min_child_weight: Minimum total of all observations' weights required in a child to control the overfitting. Too high value may result is under fitting.
- Max_depth: the depth of the tree used for controlling the overfitting. A very high depth may result in learning relations specific to a particular sample.
- Booster: The type of model used in each iteration. It can be gbtree or gblinear.

After each iteration, the similarity score is calculated as below:

$$Similarity\ score = \frac{(Sum\ of\ residuals)^2}{number\ of\ observations + lambda}$$

The gain on root is calculated as below:

$Gain = Left_{similarity} + Right_{similarity} - Root_{similarity}$

This gain is compared with the gamma value passed to prune the tree. If the gain is less than gamma, the split continues else it is stopped to avoid overfitting.

Overall, the algorithm can be formulated as *F2(x) =H0(x)+eta(H1(x)) +eta(H2(x))* where *F2(x)* is the predictions from XGBoost model, *H0(x), H1(x)… etc* are the predictions in each iteration.

Random Forest:

Random Forest is also an ensemble machine learning algorithm based on the bagging technique which is used for solving regression and classification problems. In the bagging technique, several models are fit on various subsets of the training set and then the predictions from all models are aggregated. In bagging technique, several trees are created from a different bootstrap sample of the training. Here, the bootstrap sample is a subset of the training dataset where a sample can occur multiple times in different subsets which is referred as sampling with replacement. However, unlike bagging, along with row sampling, the random forest randomly selects the features forming a subset at each split point while constructing a tree. So, the Random Forest is often referred as extension of bagging technique. The effect of predictions and the errors from each tree in random forest are not correlated to each other. Hence the aggregation of such prediction often results in better performance.

The hyperparameters used in random forest algorithm are:

- *n_estimators:* This is a parameter that determines the number of decision trees to be built before aggregating the result.
- *Max_depth:* The maximum depth of the decision tree.
- *Max_features:* The maximum number of features to be considered while subsampling.
- *Min_sample_leaf:* The minimum number of samples required to be at leaf node.
- *Min_sample_split:* The minimum number of samples required to split an internal node.
- Below are the hyperparameters that increases the speed.
- *N_job:* the number of processors allowed to use while processing.
- *Random_state:* the parameter to control the randomness of the samples.
- *Oob_score:* Out of Bang, the samples that are not used in training process. If this is set true, the 1/3rd of data is not used while training a decision tree.

SVM – Support Vector machine:

Support Vector Machine is a supervised machine learning algorithm which can be used to solve both classification and regression problems. The algorithm works by creating a n dimensional hyperplane(where n is number of independent features in a given input dataset) so that it fits the maximum number of data points by ignoring errors for the datapoints fall in the marginal distance. Here, the marginal distance is the distance between two positive and negative margins which are drawn by passing through the data points that are nearest to the hyperplane on both the sides. These margins are also called as decision boundaries. The hyperplane and decision boundaries on either side form an insensitive tube or epsilon tube. The errors on the datapoints falling within this insensitive tube are not penalized or calculated. The data points through which the positive and negative margins or decision boundaries passe are called as support vectors. The datapoints that falls outside the insensitive tube are called as slack variables. The hyperplane is fit with an objective to maximize the marginal distance so that the model built is more generalized and not aggressive.

The equations for different hyperplanes will be,

$Wx_i+b = y_i$; for hyperplane.

$Wx_i+b = +\varepsilon$; for positive hyperplane.

$Wx_i+b = -\varepsilon$; for negative hyperplane.

where w is the co-efficient value or weight, b is intercept or slope and ε is the distance between the hyperplane and decision boundary.

Thus, the objective of SVR is to fit the hyperplane in such a way that the below condition is satisfied.

$(wx_i+b) -y_i \le \varepsilon+\xi^*$ *and* $y_i-(wx_i+b) \le \varepsilon+\xi$; ξ^* and ξ is the error observed for slack variable.

Since any regression algorithm tries to minimize the error, the goal here is to minimize $\sum(\ \xi_i+\ \xi_i^*)$.

Here, lower ε value signifies that the model tolerance towards error is low and higher value signifies that the model is tolerance towards error is more.

The aim of the model is to minimize the below:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i+\xi_i^*\right)$$

With a constraint that,

$$\begin{aligned} &y_i - wx_i - b \le \varepsilon + \xi_i \\ &wx_i + b - y_i \le \varepsilon + \xi_i^* \\ &\xi_i, \xi_i^* \ge 0 \end{aligned}$$

The hyperparameters used in support vector regression are as follows:

Kernels: This is a parameter that determines the type of hyperplane to fit. It is a function that converts non separable datapoints to a separable datapoints by applying complex transformations on data which results in converting a low dimension data space to a higher dimensional space. Different kernels are linear kernels, and nonlinear kernels. The nonlinear kernels are RBF(Radial basis function), hyperbolic tangent and Polynomial [25].

Gamma: It is kernel coefficient and mainly used in nonlinear kernels. Higher the gamma value, the model exactly tries to fit the data which may result in overfitting issue. Hence it is better to parameter tuning to decide on the value of the gamma.

Epsilon: Also known as error margin is the parameter that determines the threshold of the tolerance of errors on the data points where penalty is not given to errors. It is the radius of the epsilon tube on either side of the hyperplane formed.

C: This is a regularization parameter that determines the trade-off between the training error and the flatness of the solution. Higher value of c makes model to lose the generalization property as it tries to fit as accurately as possible resulting in overfitting.

LSTM – Long Short-Term Memory:

LSTM or long Short Term Memory is a deep learning model proposed by Hochreiter & Schmidhuber in the year 1997 [26]. It is a Gated Recurrent Neural Network which has an ability to store the information for future cell processing.

LSTM makes use of memory blocks which are connected through layers. Each memory block consists of three nonlinear gates; input gate that determines whether the information to be updated in memory, forget gate that handles which information needs to be erased from the memory and output gate that determines the output based on the input and the memory state. Each LSTM cell maintains a cell state vector; using the information in cell state, the next LSTM can decide whether to read from it, or write to it or reset the cell using this gated mechanism.

LSTM overcomes the disadvantages of normal RNN by preventing gradient vanishing and exploding (avoiding long-term dependency problem), reducing the complexity of the training, and making the process of longer sequences easier.

The first step in a LSTM module is to decide what information needs to be retained and what information needs to be deleted. This is done by sigmoid(σ) function[27]. It looks at the information from previous module and the current information to out a number that lies between 0 and 1 for each number in cell state vector. If this number is 1, it means keep all the information; if it is 0, it means delete all the information. This can be represented as below.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

The next step to decide on what all information needs to be stored in the cell state. This is done in two steps, first applying sigmoid (input gate layer) which decides on the information to be updated; next using tanh layer that created a vector of new values. These two will be combined to have an update to the cell state [27] .

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

The third step is to update the old cell state(C_{t-1}) to a current cell state (C_t). This is the extension of previous two steps. That is, previous cell state(C_{t-1}) is multiplied with the information that need to be retained(f_t) which is decided in the first step, then we add this to the result obtained in the previous step [27].

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

4.1 HYPERPARAMETER TUNING:

The hyperparameters define the architecture of a machine learning model and they are learnt by the model during training process. In this study, the randomsearchCV is used for hyperparameter tuning. RandomsearchCV as the name suggests, is a technique that uses random combination of hyperparameters to find the best ones to build an optimal model.

4.2 PERFORMANCE MEASURES:

The summary of the skill and the capability of how accurately a model can predict and forecast the data is measured by performance metrics. In this study, the performance of the time series prediction models is measured based on the below parameters.

Mean absolute error: It measures the average magnitude of errors in the predictions irrespective of the direction. It is calculated using the below formula.

$$\boxed{Mean\ Absolute\ Error = \frac{1}{n}\sum_{i=1}^{n} |Y_i - Y_i^{\wedge}|}$$

Where, Y_i is the true value of each data point in the test set and $Y_i^{\wedge}$is the predicted value for a test set having n observations.

Mean absolute percentage errors: Mean absolute percentage error is also known as mean absolute percentage deviation. It gives a statistical measure of loss of a model.

$$\boxed{Mean\ Absolute\ Percentage\ Error = \frac{1}{n}\sum_{i=1}^{n} \left\|\frac{A_i - P_i}{A_i}\right\|}$$

Where A_i is the actual value of each datapoint, P_i is the predicted value for each datapoint, n is the number of observations.

Root mean squared error: Root mean squared error also known as root mean squared deviation is the standard deviation of the residuals or errors of the forecast model. Below is the formula for RMSE.

$$\boxed{Root\ mean\ Squared\ Error = \sqrt{\frac{\sum_{i=1}^{n} ||(Y_i - Y_i^{\wedge})||^2}{n}}}$$

The value of RMSE can be anything between 0 to infinity. The value of RMSE is compared with the mean of true values to check the performance of the model. If the RMSE is way less than mean value, then the performance of the model is good.

Scaled RMSE: The scaled RMSE is the normalized value of RMSE where the value is normalized between 0 and 1. The value can be calculated as mentioned below.

$$\boxed{Scaled\ RMSE = \frac{RMSE}{(\max(actual) - \min(actual))}}$$

Since the value of RMSE can be anything between 0 to infinity, usually this value is scaled between max and min of actual value to get a scaled RMSE between 0 to1. If the value is closer to 0, then the model is performing good; conversely, if the value is closer to 1, the model is failing to predict the dependent feature.

5 RESULTS AND DISCUSSION:

This section discusses overall results for all models in terms of MAE, MAPE, RMSE and scaled RMSE.

Mean absolute Error:

The below table and graph shows how the MAE value varied for each for all models.

Algorithm	Lag1 MAE	Lag3 MAE	Lag7 MAE
XGBoost	1051.628	1093.0044	1111.3384
Random Forest	1856.7038	1924.5841	2251.7415
SVR	1383.9007	1515.0595	1676.6268
LSTM	1566.5123	1604.9684	2001.9859

Table 1: Performance in terms of MAE.

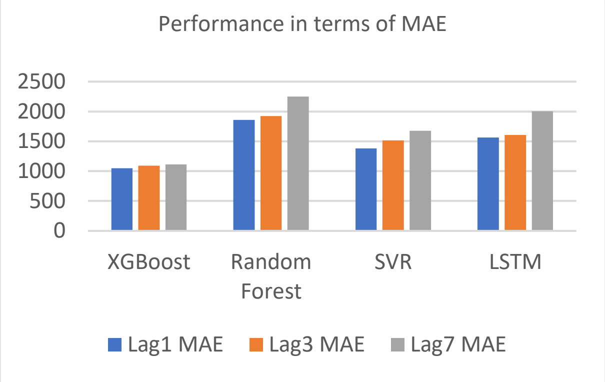


Figure 1: Performance of models in terms of MAE

Figure 1 and table 1 shows that all the models performed comparatively well for lag1 followed by lag3 and then lag7. Out of all models, XGBoost performed well for all lags compared to others and reported MAE of 1051.628, 1093.0044 and 1111.3384 for lag1, lag2 and lag3 respectively. XGBoost is followed by SVR which reported MAE of 1383.9007, 1515.0595 and 1676.6268 for the three lags considered. SVR is followed by LSTM and finally Random forest.

Mean Absolute Percentage Error:

The below table and chart show the performance of models in terms of MAPE for all lags.

Algorithm	Lag1 MAPE	Lag3 MAPE	Lag7 MAPE
XGBoost	2.5797	2.6802	2.7165
Random Forest	4.6943	4.9045	5.6312
SVR	3.1379	3.491	3.8663
LSTM	3.7254	3.9735	5.0681

Table 2: Performance of models in terms of MAPE

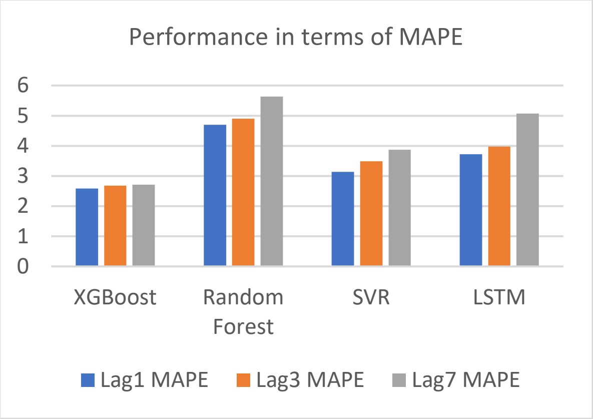


Figure 2: Performance of models in terms of MAPE.

Table 2 and Figure 2 shows the performance of all models in terms of MAPE which is less than 6%. All models performed good for lag1 dataset followed by lag3 and lag7. However, XGBoost performed best reporting least MAPE of 2.5797, 2.6802, 2.7165 for lag1, lag3 and lag7 respectively. XGBoost is followed by SVR which reported MAPE of 3.1379, 3.491, and 3.8663 for the three lags which is followed by LSTM and Random Forest algorithms.

Root Mean Squared Error:

Table 3 and Figure 2 show the performance of models in terms of RMSE.

Algorithm	Lag1 RMSE	Lag3 RMSE	Lag7 RMSE
XGBoost	1475.9624	1519.9507	1534.9107
Random Forest	2294.4461	2400.4814	2765.9959
SVR	2161.4242	2236.684	2407.6039
LSTM	1926.3234	2313.5768	2694.1189

Table 3: Performance in terms of RMSE

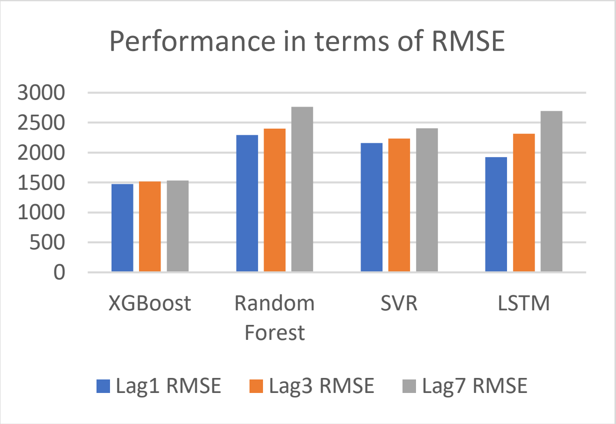


Figure 3: Performance of models in terms of RMSE.

In case of RMSE as a performance measure, all models showed less RMSE for lag1. However, XGBoost performed best for all lags with RMSE at 1475.9624, 1519.9507 and 1534.9107 for lag1, lag3 and lag7 respectively with a slight variation in values. The next model to perform better for lag1 is LSTM which reported RMSE of 1926.3234 followed by SVR with RMSE of 21621.4242. However, for lag2 and lag3, SVR performed better with RMSE of 2236.684 and 2407.6039 respectively. Random forest reported higher RMSE compared to all other models. The RMSE reported by all the models are way less than the average of true values (which is ~42000) which signifies that these modules are satisfactorily good at predicting task.

Scaled RMSE:

The below chart shows the performance of models in terms of scaled RMSE. Since RMSE can vary from 0 to infinity, scaling the obtained RMSE based on max and min of true value gives better understanding on how well a models performs predicting task. The value close to 0 signifies a perfect model whereas the value close to 1 implies the model's failure towards prediction.

Algorithm	Scaled RMSE Lag1	Scaled RMSE Lag2	Scaled RMSE Lag3
XGBoost	0.0295	0.0304	0.0307
Random Forest	0.0471	0.0481	0.0554
SVR	0.0433	0.0448	0.0482
LSTM	0.0386	0.0423	0.0539

Table 4: Performance of models in terms of scaled RMSE.

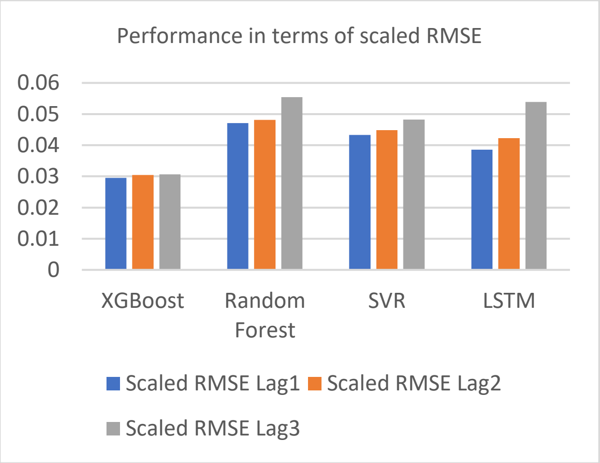
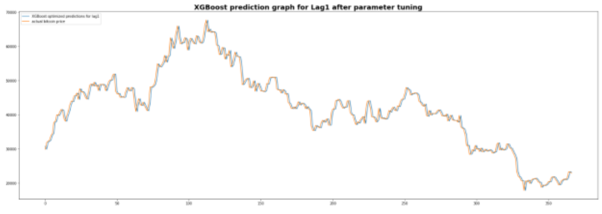


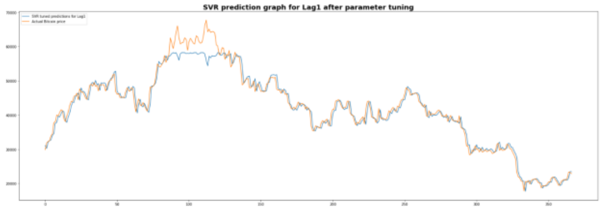
Figure 4: Performance of models in terms of scaled RMSE

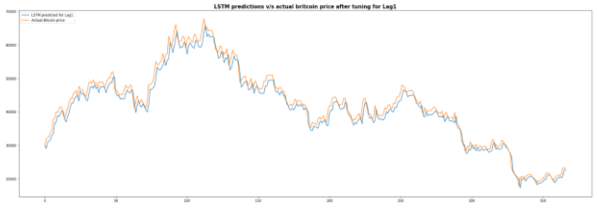
Table 4 and Figure 4 show the performance of models in terms of scaled RMSE. All the models performed well reporting scaled RMSE varying from 0.0295 to 0.055 and they reporter lesser scaled RMSE for lag1 dataset. Compared to all models, XGBoost reported the lowest scaled RMSE of around 0.03 for all lags. The next model to report lower scaled RMSE is LSTM for lag1 and lag2 followed by SVR. However, SVR reported slightly lesser scaled RMSE for lag 3 compared to LSTM. Random Forest report highest scaled RMSE values compared to other models.

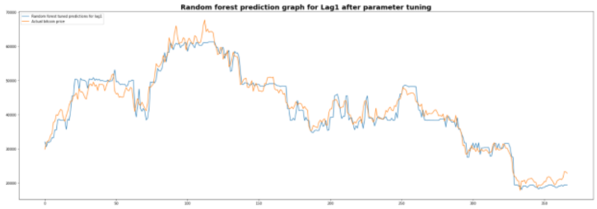
Overall, the results indicate that though all models performed well after parameter optimization for lag1. XGBoost outperformed compared to all other models for all lags and in terms of all performance measures considered in the study and compared to other algorithms, Random Forest reported fairly good performance with minimal difference in performance metrics before and after parameter tuning.

Actual v/s prediction graphs of all models for each lag:
For lag1:


XGBoost prediction graph for Lag1 after parameter tuning

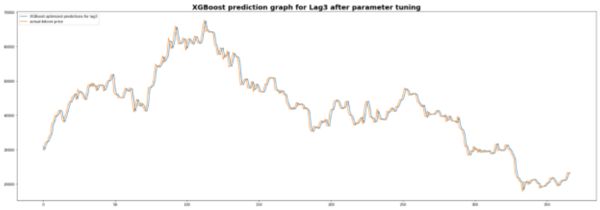

SVR prediction graph for Lag1 after parameter tuning

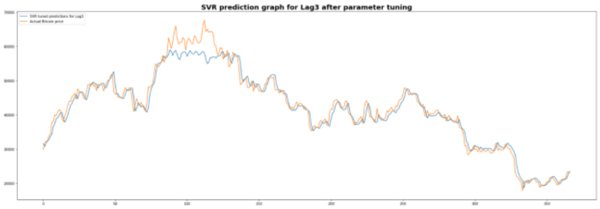

LSTM predictions v/s actual britcoin price after tuning for Lag1

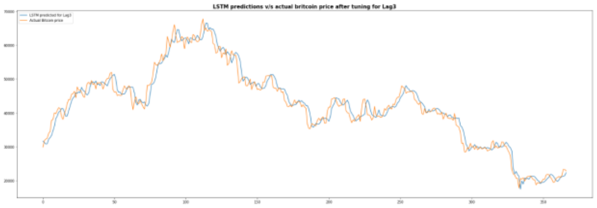

Random forest prediction graph for Lag1 after parameter tuning

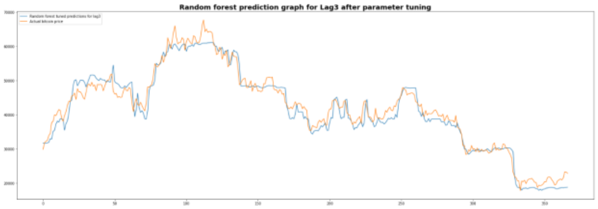
The above graphs show the actual and prediction curves for lag1. In case of XGBoost, these curves follow each other closely which signifies the error in prediction in this model is very less. This is followed by SVR where most of the actual and predicted datapoints are close to each other. In case of LSTM, though the price movements are captured properly, the curves are not as close as seen in XGBoost.

For Lag3:


XGBoost prediction graph for Lag3 after parameter tuning

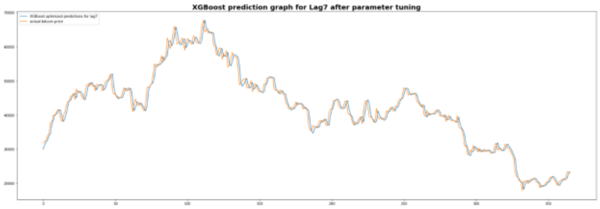

SVR prediction graph for Lag3 after parameter tuning

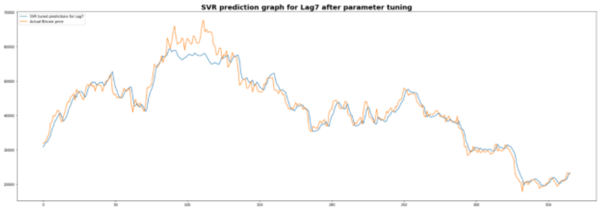

LSTM predictions v/s actual britcoin price after tuning for Lag3

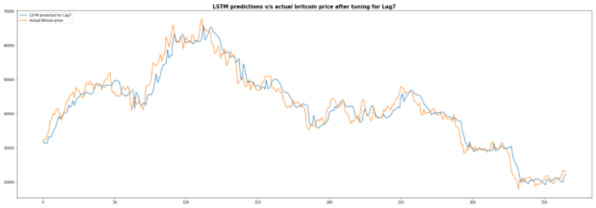

Random forest prediction graph for Lag3 after parameter tuning

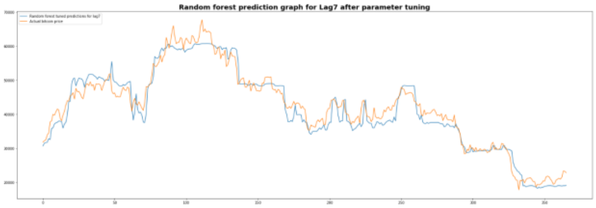
The above graphs show the difference in predictions compared to actual data points. In case of XGBoost the difference in actual and predicted data points is very less. This is followed by SVR, LSTM and Random Forest. Also, when all these compared with lag1 graphs, lag3 graphs shows decreased performance with slight increase in the difference between actual and predicted values.

For lag7:


XGBoost prediction graph for Lag7 after parameter tuning


SVR prediction graph for Lag7 after parameter tuning





The above graphs the comparison of actual and prediction curves for lag7. XGBoost prediction curve closely follow the actual curve. This is followed by SVR, LSTM and finally Random Forest. When lag 7 graphs are compared with lag3 and lag1, there is a further increase in the difference between actual and predicted datapoints and the curve is smoothened more in case of lag compared to lag3. However, in case of XGBoost this difference cannot be seen easily.

5.1 DISCUSSION:

This chapter discusses the major findings with an aim to address the following research questions that have been stated before.

- Which model performed the best in predicting task?
- How accurately the best model could predict the future bitcoin price?
- Which lag gave the best results?

The results indicate that all models agreeably performed well after parameter tuning with MAPE less than 6% which implies that they are good models for forecasting. XGBoost performed the best by reporting good performance metrics for all lags. This implies that, for the considered dataset, the ensemble-based boosting technique achieved good results. In this type of machine learning mechanism, the strong learners are formed using feedback from the weak learner to achieve better performance. The next algorithms that performed well are SVR and LSTM.

XGboost resulted in the lowest error deviation while predicting the bitcoin prices and reported a minimum loss percentage of 2.5%. Meaning, that it could predict around 98.6% of data accurately in the given dataset. This was followed by SVR and LSTM with a slight difference.

Among all the considered lags, all algorithms reported better results for lag1. This means that for the given dataset, independent features at timestep t-1 were efficient in predicting the bitcoin price at time t.

From the results obtained, it is found that random forest performed fairly good with the minimum difference in performance before and after hyperparameter tuning. Random forest is based on ensemble-based bagging technique which performs random sampling of observations in training datasets that results in the reordering of data points of a given input time series. This might be the reason for the fair performance of the random forest algorithm. Having said that, further research can be done in this area which is beyond the scope of this study.

In a nutshell, XGBoost performed the best in predicting bitcoin prices considering historical exchange rates of the various currencies and bitcoin prices as price determinants.

6 CONCLUSION AND FUTURE SCOPE:

Bitcoin prices are highly fluctuating and hence it is more difficult to forecast its prices. However, the returns on the bitcoins are anticipated from the future bitcoin price and hence it is necessary to build an accurate model to predict bitcoin price. This study aimed at developing such machine learning models for the prediction of bitcoin prices based on currency exchange rates in forex market. The models built can be used for short-term forecasting like predicting next day's bitcoin price.

The study contributed to the existing knowledge by developing the four multivariate predictive models which are based on different machine learning techniques to forecast bitcoin prices. These models are: an ensemble-based boosting algorithm – Xgboost, an ensemble-based bagging algorithm – Random Forest, Machin learning regression model – Support Vector Regression and deep learning based recurrent neural networks model – LSTM are built as part of this study.

In this study, exchange rates of major currencies and historical bitcoin prices are considered as independent features, and the next day's bitcoin price is considered as dependent features. The above-mentioned models are built on this data to forecast the next day's bitcoin price. Overall, all the models showed preferable performance with MAPE being less than 6% and gave satisfactory results for the lags1, 3 and 7. Though all the models performed better, XGBoost outperformed other models by reporting least performance metrics for all lags.

The study addressed all the research questions and demonstrated that it is possible to forecast bitcoin prices by considering historical bitcoin prices and currency rates in the forex market as independent features. Also, this study proves that the bitcoin forecast models can be built to have optimal performance with fewer errors. All the models gave favourable results and exhibit potential for further usage in applications related to fintech.

Limitations and Future scope:

Although this study accomplished in developing the bitcoin price forecast models based on the forex market by achieving satisfactory results, due to time constraints the study was limited to daily historical data of forex currency rates and bitcoin prices. Further study can be done as an extension of this project by considering the below.

- Since bitcoin exhibits high volatility, further study can be done to examine the hourly data and forecast the price on an hourly basis.
- Further study can be done on more advanced neural network models combined with ensemble-based or reinforcement techniques.
- Further, this study can be expanded to include sentiment analysis of tweets and bitcoin mentions in the news and other social media along with the forex market as one set to forecast the bitcoin price.
- It can also be extended to include the stock market and forex market as another set of determinants in bitcoin price forecasting.

7 REFERENCES:

1. Monia Milutinović (2018), "cryptocurrency", Ekonomika - Journal for Economic Theory and Practice and Social Issues. https://www.ceeol.com/search/article-detail?id=695295.
2. Mohammed Mudassir, Shada Bennbaia, Devrim Unal, Mohammad Hammoudeh (2020). "Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach" . https://link.springer.com/article/10.1007/s00521-020-05129-6
3. Zeinab Shahbazi and Yung-Cheol Byun.(2022)."Knowledge Discovery on Cryptocurrency Exchange Rate Prediction Using Machine Learning Pipelines".https://doi.org/10.3390/s22051740
4. Jacques Vella Critien, Albert Gatt & Joshua Ellul (2022). "Bitcoin price change and trend prediction through twitter sentiment and data volume". https://jfin-swufe.springeropen.com/articles/10.1186/s40854-022-00352-7#Sec1
5. Yu Yan,Yiming Lei,Yiming Wang (2022). "Bitcoin as a Safe-Haven Asset and a Medium of Exchange".https://doi.org/10.3390/axioms11080415
6. Laura Alessandretti , Abeer ElBahrawy , Luca Maria Aiello , and Andrea Baronchelli (2018), "Anticipating Cryptocurrency Prices Using Machine Learning". https://doi.org/10.1155/2018/8983590.
7. David Yermack (2013). "is bitcoin a real currency? an economic appraisal". http://www.nber.org/papers/w19747.pdf
8. Jamal Bouoiyour, Refk Selmi (2015). "What Does Bitcoin Look Like?". https://down.aefweb.net/AefArticles/aef160211Bouoiyour.pdf.
9. Ismet Voka, Filipos Ruxho (2022), "bitcoin as a substitute for current currencies". https://www.esd-conference.com/upload/book_of_proceedings/Book_of_Proceedings_esdVarazdin2022_Online.pdf#page=60.
10. Mustafa Ozyesil (2019). "A research on interaction between bitcoin and foreign exchange rates". http://www.pressacademia.org/archives/jefa/v6/i1/5.pdf.
11. Özge Korkmaz. (2018). "The relationship between Bitcoin, gold and foreign exchange returns: The case of Turkey".http://www.kspjournals.org/index.php/TER/article/view/1807.
12. Mohamed Khalil BENZEKRİ, Hatice Şehime ÖZÜTLER. (2021). "On the Predictability of Bitcoin Price Movements: A Short-term Price Prediction with ARIMA". https://doi.org/10.26650/JEPR.946081
13. Behrouz Shakeri, Artin Beytarib, Mohammadreza Ghorbanianb, Rouhollah Javadi. (2022). "Evaluation of the association between cryptocurrencies with oil and gold prices using the BEKK multivariate GARCH model". 10.22075/IJNAA.2022.27155.3523. https://ijnaa.semnan.ac.ir/article_6654.html
14. Gowtham Saini, Dr. M. Shobana.(2022). "cryptocurrency price prediction using prophet and Arima time series". https://www.irjmets.com/uploadedfiles/paper/issue_4_april_2022/21042/final/fin_irjmets1650284252.pdf
15. Mohammed Mudassir, Shada Bennbaia, Devrim Unal & Mohammad Hammoudeh. (2020). "Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach". https://link.springer.com/article/10.1007/s00521-020-05129-6
16. Mohammad J. Hamayel, Amani Yousef Owda (2021). "A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms". https://doi.org/10.3390/ai2040030
17. Ioannis E. Livieris, Niki Kiriakidou, Stavros Stavroyiannis and Panagiotis Pintelas. (2021). "An Advanced CNN-LSTM Model for Cryptocurrency Forecasting". https://www.mdpi.com/2079-9292/10/3/287
18. Nayak, S.K., Nayak, S.C. and Das, S. (2021) . "Modeling and Forecasting Cryptocurrency Closing Prices with Rao Algorithm-Based Artificial Neural Networks: A Machine Learning Approach". https://doi.org/10.3390/fintech1010004
19. Pratiksha Patil. (2022). "Bitcoin Price Prediction Using Machine Learning and Neural Network Model". http://ijaem.net/issue_dcp/Bitcoin%20Price%20Prediction%20Using%20Machine%20Learning%20and%20Neural%20Network%20Model.pdf
20. Monisha Mittal, G. Geetha. (2022). "Predicting Bitcoin Price using Machine Learning". https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9740772
21. Ninuk Wiliani, Rizki Hesananda, Nidya Sari Rahmawati and Erdham Hestiadhi Prianggara. (2022). "APPLICATION OF MACHINE LEARNING FOR BITCOIN EXCHANGE RATE PREDICTION AGAINST US DOLLAR". http://ejournal.nusamandiri.ac.id/index.php/jitk/article/view/2880/895
22. Chunxiao Yan, Mengze Li, and Shengao Zhang. (2022). "The Empirical Analysis of Bitcoin Price Prediction Based on Deep Learning Integration Method". https://doi.org/10.1155/2022/1265837
23. Tianqi Chen and Carlos Guestrin. (2016). "XGBoost: A Scalable Tree Boosting System". https://arxiv.org/pdf/1603.02754.pdf
24. EL Houssainy A. Rady, Haitham Fawzy and Amal Mohamed Abdel Fattah. (2021). "Time Series Forecasting Using Tree Based Methods". https://www.naturalspublishing.com/files/published/l259iab891zec2.pdf
25. https://link.springer.com/article/10.1007/s00521-020-05129-6#Tab2
26. Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735
27. Christopher Olah. (2015). "Understanding LSTM Networks". https://colah.github.io/posts/2015-08-Understanding-LSTM