SCHOOL OF ARCHITECTURE, COMPUTING & ENGINEERING

Dissertation

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

Student Name: Roopashree Ramachandraiah

Student Number: U2177053

Supervisor: Fahimeh Jafari

Module Code: CN7000

Acknowledgement

I would like to express my deepest gratitude to my professor Dr Fahimeh Jafari for her continuous feedback and invaluable patience which helped me in completing the project successfully. Her extended support and guidance throughout the project timeline are highly appreciable.

I would also like to extend my gratitude and sincere thanks to professor Dr Mitra Saeedi for her support and encouragement.

I am thankful to my family and everyone who helped me complete this project by showing their support.

Abstract

Cryptocurrencies are decentralized virtual currencies that do not have any governing centralized authorities which differs them from other currencies that are centralized and authorized by banks. Bitcoin works on peer-to-peer transactions which are based on a technology called blockchain. It is a public decentralized ledger to maintain transactions. Bitcoin is the first cryptocurrency that was built on blockchain technology. It is one of the famous cryptocurrencies having the highest market at present. The main characteristic of bitcoin or other cryptocurrencies is that they exhibit high volatility in terms of price. Such property makes it difficult for financial investors and economists to study its underlying price model. This dissertation 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 parameter 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.

Table of Contents

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. The analysis of historical data can further be used to predict the likelihood of future events, and this refers to time series forecasting. 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 a technique named cryptography to make the transactions secure[1]. Unlike other currencies that are centralized and authorized by banking systems across the world, 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].

Among all the existing cryptocurrencies, bitcoin is the most popular one and has gained attention from investors and traders. Bitcoin started gaining popularity in the year 2017 and had exponential growth in 2018. It recorded the highest close price and exhibited extreme price fluctuations in the year 2021 because of which it attracted huge public interest during these years. 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.

1.1 Problem statement:

Cryptocurrencies exhibit volatility and non-stationary behavior in terms of price. Hence, this nature enforces the need for understanding and analyzing the underlying patterns of their price models. The same nature also supports investors' and researchers' views that these cryptocurrencies are speculative assets. 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 or prices of these cryptocurrencies. However, determining the exact feature that affect bitcoin is highly difficult due to the rapid change in the pattern or trends in their price over the time. This adds the difficulty level for building a model based on a feature. Since the anticipation of future returns or profits of bitcoins depend 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.

1.2 Objectives and contribution:

Considering the problem statement, the primary objective of the dissertation is to develop machine learning based predictive models to forecast bitcoin prices as accurately as possible 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.

***Contribution**:*

- To build multivariate predictive models using Machine Learning techniques where the dependent feature is bitcoin price and independent features are historical exchange rates of major currencies listed above.
- To develop machine learning models like XGBoost, Random Forest, SVR and LSTM for prediction of bitcoin prices.
- To build models which can be used to forecast the next day's price based on different lags (lag1, lag 3 and lag 7).
- To perform exploratory data analysis to interpret the behavior of bitcoin prices with respect to other currencies.
- Compare the result of the models built to look for their performance and determine the best-model for the problem statement.

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?
- Which model performs the best in predicting bitcoin price?
- Which lag gives the best results compared to all the lags considered?

Overall, the primary goal of this dissertation is to develop machine learning based predictive models like Support Vector Regression (SVR) , XGBoost , Random Forest and Long Short Term Memory (LSTM) to forecast bitcoin price considering time series data of various foreign currencies as price determinants. As elucidated in literature review, the research is done in related topics such as the need for bitcoin price predictions that explains the problem statement, and particularly in, application of machine learning techniques in forecasting bitcoin price using time series historical data.

1.3 Structure of the dissertation:

The following points provide an overview of the report.

- Chapter-2: This chapter narrates the various relative research such as studies done around cryptocurrencies and bitcoin prices, the need for forecasting bitcoin prices, various approaches used for prediction of bitcoin prices.
- Chapter-3: This chapter briefs about bitcoin and how they are created and traded. It also explains the blockchain technology used in bitcoin's transactions.

- Chapter-4: This chapter explains the different kind of machine learning techniques used in the study.
- Chapter-5: This chapter describes the technical background of platform, tools and libraries used to carry out the implementation part of the research.
- Chapter-6: This chapter details the data extraction and preprocessing of the data used. It also describes the quick exploratory analysis of the data.
- Chapter-7: This chapter describes the exploratory analysis done on the data using graphs.
- Chapter-8: This chapter details the methodology proposed to carry out the practical implementation of research. It also describes the algorithms used to develop the predictive models and the performance measured used to interpret the results of the models.
- Chapter-9: The results obtained are described and the interpretation of results are discussed in this section.
- Chapte-10: This chapter discusses the conclusion derived from the research and lists the limitations and future scope of the study.

2 Literature Review:

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 like media attention (e.g., google searches, twitter mentions), various commodities (e.g., gold, stocks, crude oil) that are impacting the bitcoin price predictions.

2. *Bitcoin price and its attention on social media and google:*

From the time bitcoin started gaining attention in financial market, several research are done to explore the connectedness between bitcoin price in the market, and the influence of the internet and social media on its price. In 2013, Kristoufek analysed the impact of interest in bitcoin currency leading to the search queries on Wikipedia and google trends and bitcoin price [10]. He used vector autoregression method to find correlation between these elements. In his study, though he found an increased correlation between the search queries and bitcoin price, he concluded that this relationship is bi-directional. In 2015, Plosik et al., conducted empirical study on bitcoin price and showed that the returns on bitcoin are greatly influenced by several factors like its popularity, the total number of transactions and the sentiments described in newspaper and its frequent searches on google[11]. He also stated that unfavorable comments on bitcoins had a negative impact on their returns. However, in 2018 Panagiotidis et al., stated that the search intensity of bitcoin on google and Wikipedia had a reduced impact on bitcoin [12]. Another research by Dastgir et al., in 2019 examined the relationship between bitcoin's return and its attention on the internet measured by a number of searches on google trends by employing Copula-Granger Causality in Distribution (CGCD) test and observed bi-directional relationship among these factors [13]. In 2020, Kapar and Olmo, examined the long-term association between bitcoin price and a set of variables like the performance of financial market, gold price and google search using co-integrated methods for two different time periods (2010-2017 and 2010-2019) and found that in common the returns are influenced by google searches for both the time periods [14]. In 2022, Critien et al., used sentiments in Tweets and the volume of tweets for predicting magnitude of bitcoin price changes by using neural network models based on recurrent network and convolutional network and achieved a relative accuracy of 63% [15]. In 2022, Raza et al., investigated the asymmetric relationship between forex rates and cryptocurrencies using quantile to quantile approach and found that the digital currency prices and forex market rates varied based on several factors and suggested the policy makers and investors to use this as economically important factor to analyze [16].

3. Bitcoin price and other financial factors:

When it comes to other economic factors like gold price, US Dollar index, consumer price index, and federal funds rate, several studies have been conducted to analyze the influence of these factors on bitcoin prices. Zhu et al. built a Vector Error Correction model in 2017 to find cointegration relationship among the identified variables and observed that these factors negatively influence bitcoin prince on long-term basis except gold which has no relationship on bitcoin price in long run [17]. However, in the same year, Baur et al., analyzed statistical features of bitcoin and other traditional assets such as stocks, bonds, and commodities, and found that bitcoin's return has different properties compared to traditional assets and currencies [18]. In 2020, Matkovskyy et al., analyzed the influence of economic policy uncertainty on the connectedness between bitcoin and the traditional stock market and found that economic uncertainty decreased the interdependence between the stock market and bitcoin and a significant relationship exists between economic policy uncertainty and fluctuations in bitcoin market [19]. Another study conducted by Barson et al., in 2022 proved the existence of asymmetric relationship between cryptocurrencies and gold returns [20].

4. Approaches used in bitcoin price prediction:

When it comes to regression problems, algorithms based on various approaches are used for building predictive models. In the case of bitcoin price predictions, mainly two kinds of algorithms are used 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 [21], 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 [22] proved the existence of relationship between gold, EURO and USD returns using supaugmented Dickey-Fuller (SADF) and Generalized SADF. In 2021, Benzekri and Özütler [23] 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., [24] 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 [25], 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 [26]. 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 [27]. 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 [28]. 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) [29]. 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 [30]. 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[31]. 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[32]. In June 2022, Yan et al., implemented an integrated approach of deep neural network and bagging technique to forecast bitcoin prices [33]. 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.

The below table gives an overview of the studies done for forecasting bitcoin prices using machine learning based models which explored as part of this literature review.

Author/ Title	**Independent Features**	**Dependent Feature(s)**	**Timeframe**	**Frequency**	**Method/algorithms**	**Main findings/ Performance**	**Identified Gap**
[23]	Historical bitcoin prices	BTC price	2014Q1 to 2020Q2 – Train data 2020Q3 and Q4 test data	Quarterly data	Univariate Arima	MAPE 4.24%	A univariate model based on only historical BTC price.
[25]	Historical bitcoin prices (adjusted close price)	BTC price	2017-2021	Daily	Univariate Arima & FG Prophet	R^2 Arima- 0.94 R^2 FB Prophet- 0.93. Arima performed better than FB Prophet	A univariate model based on only historical BTC price
[26]	Technical indicators derived from raw features (ex., transactions, Block size, sent address etc.) of bitcoin price and transactions	BTC price	April 2013- July 2016 and April 2013 – April 2017	Daily	ANN, Stacked ANN, SVM and LSTM multivariate models	LSTM performed better. Bitcoin price could be forecasted with less error, however predicting its raise or fall seemed difficult	Only features related to bitcoin are used.

[27]	Historical data of BTC, ETH and LTC (Open, High, Low, and close price of each currency)	BTC, ETH and LTC prices	Jan 2018-Jun 2021	Daily	LSTM, BiLSTM and GRU	GRU performed well compared to other models	Only historical features of BTC, ETH and LTC are used to predict their prices
[28]	Historical mixed data of BTC, ETH and XRP	BTC, ETH and XRP price movement	Jan 2017 – Oct 2020	Daily	CNN-LSTM based Multi-Input Deep neural network	The new hybrid approach of exploiting different currency data separately for their movement prediction resulted in reduced overfitting	Though multiple cryptocurrency data are considered, the main aim was to predict the movement(raise or fall in price). Other features like open, close, low, trade volumes are not considered
[29]	Historical closing price of BTC, LTC, Ripple, CMC 200 and Thether	Multiple cryptocurrencies	Jan 2019 – Mar 2021	Daily	RA+ANN, GA+ANN , PSO+ANN, MLP, SVM and ARIMA	RA+ANN model performed good overall	Only closing price was used for forecasting. Scope for using other technical determinants
[30]	Day transactions having high, low, open features	Bitcoin price	Oct 2020 to Aug 2020	Hourly	Linear Regression and LSTM	LSTM performed	Only bitcoin features are considered
[31]	Various bitcoin featured like open, low, high, close, volume	Bitcoin price	2014 to 2020	Daily	Multivariate GRU model	Resulted MAPE 18.4905% and RMSE 1987.1057	Only bitcoin features are considered
[32]	Various bitcoin featured like open, low, high, close, volume	Bitcoin price	July 2021 to Nov 2021	Daily	Multivariate linear regression and ANN	ANN performed better	Only featured related to bitcoin exchange rates in USD are considered

[33]	Historical bitcoin features like block size, number of transactions, search volume etc along with gold price and dollar index	Bitcoin price	Nov 2014 to Mar 2020	Daily	Multivariate model-SDAE-B, an integration of deep neural network and bagging technique	Results compared with traditional LSSVM and BP and found that the integrated model SDAE-B performed good	Though a few extra features like gold rate and dollar index were considered, other currency exchange rates are not considered

Gap in the knowledge*:* From the literature review, we can see that most of the studies which considered forex rates or other financial price determinants were only intended to analyze the connectedness between these features and price and not the price prediction. Also, we can see that most of the studies with the objective of bitcoin price prediction considered only raw features of the currency like high, low, open, close prices 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 Brief on Bitcoin and 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]. Unlike other currencies that are authorized by central authority, the cryptocurrency is decentralized and work on peer-to-peer technology for managing transactions. All the transactions are maintained in a public digital centralized ledger using a technology called blockchain. Thus, bitcoins are not governed by any institutions or banks. The mining process used to create bitcoins involves solving complex mathematical puzzles which were once easier to solve by an average person using normal computer set up. However, the bitcoin code is written in such a way that solving puzzles becomes more and more complex over the time and needs super-fast computers with more computational resources.

Blockchain is the technology used to maintain the transactions done using cryptocurrencies. These transactions are stored in the form of blocks. The recent transactions created periodically are stored in blocks. Once the transactions are completed, these blocks are attached to the chain. Before these blocks are attached to chain, they are verified by majority of nodes to confirm the validity of the data. Once this verification is confirmed, the blocks are attached to ledger. The transactions are secured using cryptography that involves solving complex computations to process a transaction. Hence the blockchain technology to maintain transactions are considered highly secure.

4 Machine learning, Ensemble learning and Recurrent Neural networks:

Machine learning is a subset of artificial intelligence that aims at developing an algorithm or model that allows computers to learn from the data, explore the data patterns, tune the performance while learning and predict the outcome. The models built can be used for making predictions or decisions for similar data without having to program explicitly. Thus, machine learning enables computers to solve the problem by learning and exploring sample real-world data without being explicitly programmed to do so. Further machine learning models can be classified into supervised and unsupervised learning where supervised learning aims at solving regression and classification problems, and unsupervised learning aims at solving clustering and association problems.

This project aims at forecasting the bitcoin prices which belong to supervised machine learning problem, in particular regression problem. The various algorithms implemented as part of this research to solve the problem statement and address the research questions are XGBoost, Random Forest, Support Vector Regression and LSTM. These algorithms are built based on different machine learning techniques discussed below.

Regression: It is a technique used for prediction problem where the aim is to find the relationship between the independent and dependent features. In this study, Support Vector Regression is used which is an extension of linear regression where the goal is to fit a hyperplane on datapoints with a goal to reduce the error due to deviation.

Ensemble Learning: Ensemble learning is a type of machine learning technique where the results from multiple sub models or base models are combined to produce an optimal model. Ensemble techniques are further classified as boosting, bagging, and stacking techniques. In this research, XGBoost based on boosting technique and Random Forest based on bagging technique are used to develop predictive model.

Recurrent Neural Network(RNN): Another algorithm used in the study is LSTM, which is based on RNN. RNN is a type of neural network which are mainly used in time dependent sequential data problems. RNN works by maintaining information of current cell state which are used as input to future cells. This way, the output of each layer is fed as input to the next layer as the information progresses through different layer of neural network.

5 Technical background and libraries used:

This section briefly explains the platform, programming language and different libraries used to carry out the practical implementation of the research.

5.1 Platform used:

The platform used to develop the models is Google colab. It is a free online platform for writing and executing python code and provides free access to computing resources like GPUs and TPUs.

5.2 Programming language:

The practical implementation of the study is conducted using python language as it has various advantages over any other programming language in terms of machine learning engineering. Some of them are listed as follows.

- Platform independence: Python is one of those popular programming languages which can be executed on multiple platforms without having to change the code. It runs across different platforms like Windows, macOS, and Linux requiring little or no change.
- Consistency and simplicity: Python is the most consistent and simple programming language when compared to other object-oriented languages like Java, and C++. It has a clear simple syntax which is appealing to many programmers and beginners. Due to its simplicity with respect to syntax, it is easy for programmers to interpret the code.
- Support variety of libraries: Python has a rich technology stack in terms of libraries which makes it even simpler for developers to import and use them to solve problems by writing a few lines of code. When it comes to machine learning, python has various libraries. The most popular ones are TensorFlow, Keras, and Scikit-learn. To perform scientific computing and data analysis, it has NumPy; and SciPy for advanced computing. Pandas is yet another powerful library for structuring data in rows and columns and performing data analysis. For visualization, it has matplotlib and seaborn libraries.

5.3 Libraries used:

Below is the overview of various libraries used in this study.

- *Pandas:* It is the most widely used python library for data analysis when the data can be represented in a tabular form. It is particularly used for data exploration, cleaning, and processing data for analysis [34].
- *Numpy:* It is the most common library used when it comes to scientific computing in Python. It supports multidimensional arrays and is a convenient library to use while dealing with fast array operations like mathematical computations, logical operations, shape manipulations, and basic linear and statistical operations [35]. Many of the in-built algorithms and functions take input parameters in the form of arrays.
- *Matplotlib:* It is the widely used visualization tool to represent the data in the form of graphs and charts.
- *Seaborn:* This is another visualization library in python which is based on matplotlib. The functions can be used for a variety of operations ranging from basic plotting such as scatter plots and line plots to transformation and abstractions like histogram and kernel density estimation [36].
- *Sklearn:* also known as sci-kit learn is the powerful library in python used for machine learning projects. It provides variety of machine learning algorithms for classification, regression, and clustering problems. It also provides algorithms like standard scaler, robust scaler and MinMax scaler for preprocessing the data as part of data normalization and standardization; random search and grid search algorithms for parameter tuning [37].
- *Tensor-flow:* an open-source end-to-end library for building and deploying machine learning models. In particular, it is mainly used for training and inference of deep neural networks or deep learning-based models.
- *Keras:* It is a deep learning API in python built on top of TensorFlow for fast experimentation with a focus on modern deep learning [38].

- *Statsmodels:* It is a python library that provides various functions and classes to estimate different models[39]. It also provides functions to conduct statistical tests and statistical exploratory analysis [39].

6 Data extraction and preprocessing:

The below sections explain the practical implementation of the research. Firstly, it briefs about how the data is extracted for the study and how it is preprocessed to handle missing data. Next, it briefs about how the timeseries is converted to introduce lags in the data to feed it to aforementioned supervised machine learning models. Finally, it explains how the data is split for training and evaluating the model.

6.1 Data extraction:

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 collected is daily data consisting of overall 1836 observations for each variable for the chosen period.

The Below table lists the features and the description of each feature used in this research.

Feature names	**Description**	**Dependent(D)/Independent variable(I)**
XBTUSD	Bitcoin price in USD	I
AUDUSD	Australian dollar price in USD	I
CADUSD	Canadian Dollar price in USD	I
CNYUSD	Chinese yuan price in USD	I
EURUSD	Euro price in USD	I
GBPUSD	Great Britain Pounds in USD	I
JPYUSD	Japan Yen in USD	I
NZDUSD	NewZealand Dollar price in USD	I
SGDUSD	Singapore Dollar price in USD	I
Next_day_XBTUSD	Next day’s bitcoin price	D

Table 1: List of dependent and independent features

6.2 Handling missing values:

The data is examined and sorted in the chronological order. Cryptocurrencies are traded 24/7 all days a week. So, the bitcoin’s time series data did not have any missing values. However, the forex market is closed on weekends, and hence there were missing values for Saturdays and Sundays in other currencies’ data. Since the fluctuation of the forex market is minimal during weekends, the missing values are handled by substituting the weekend rate by Friday’s close value of corresponding currency rates.

6.3 Converting time series data for supervised ML models:

The important step in any time series prediction is to convert it into dependent and independent features so that it can be used in supervised learning models. In this study, it is achieved by using the shift function in the pandas dataframe. For example, to create a variable with lag-1 which indicates all the features captured at time t-1, the entire dataframe is shifted downwards once. Similarly, to have features with lag 3, the dataframe is shifted downwards three times and in this case, each time shift is performed, the resultant dataframe is appended with the new columns formed after shift operation. So generally, for a dataframe having m features, to consider lag-n, the dataframe is shifted downwards n times and the new columns($X_{1(t-1)}, X_{1(t-2)}, \ldots.. X_{1(t-n)}$, $X_{2(t-1)}, X_{2(t-2)}, \ldots.. X_{2(t-n)}, \ldots\ldots$, $X_{m(t-1)}, X_{m(t-2)}, \ldots.. X_{m(t-n)}$, *resulting in m*n number of independent features*) are appended.

The dataset considered for the research had nine features. The number of independent features formed after performing the above shift operation for lag-1, lag-3 and lag-7 are nine, twenty-seven and sixty-three respectively.

The independent feature is created by shifting the column XBTUSD(which is the target feature in the research) upwards. For example, to predict the 2nd day's bitcoin price, then the target column is shifted twice. Similarly, to predict the 7th day's bitcoin price, the target column is shifted upwards seven times. So, to predict the nth day's price, the target column is shifted upwards n times.

This research aims at forecasting the next day's bitcoin price. Since all the features are already shifted downwards for considered lags (lag1, lag3 and lag7) forming independent features(*at time t-1 for lag1, at t-1,t-2, t-3 for lag3 and at t-1, t-2,… t-7 for lag7*), and the very next day's bitcoin price is forecasted, the exiting target column (*at time t*) is fed as a dependent feature to the models.

After performing the above shift operation, the NaN values were found in rows; such rows are discarded before splitting the data into training and testing samples. Therefore, the final number of observations for lag1, lag3 and lag7 data are 1486, 1843 and 1481 respectively.

6.4 Train and Test data split:

The next step after preprocessing the data is to split the preprocessed data into train and test sets. In this work, the usual two-sub-sample method is used for data splitting which is the most common method used in any machine learning model. After converting the time series data for supervised learning as mentioned in the previous section, the daily data consisting of 1836 observations are separated into training and testing samples.

The first 80% of data is used as a training sample to train the model. The training data for lag1 has 1468 daily observations(from date 03-Jan-2017 till 18-Jul-2021), lag3 has 1466 daily observations(from date 03-Jan-2017 till 16-Jul-2021) and lag7 has 1463 daily observations (from date 03-Jan-2017 till 13-Jul-2021).

The remaining 20% data is used as test data to test the performance of the model. Therefore, the test sample for lag1 has 367 daily observations(from date 19-Jul-2021 till 20-Jul-2022), lag3 has 367 daily observations (from date 17-Jul-2021 till 18-Jul-2022), and lag7 has 366 daily observations (from date 15-Jul-2021 till 14-Jul-2022).

7 Exploratory Data analysis:

The next step while building the machine learning model is to perform some exploratory analysis. This section describes the brief analysis done on the data to derive some quick insights from the data and its patterns.

- ***Bitcoin price fluctuation over the time:***

The below graph shows how the bitcoin price has been increasing over the time from the year 2017 till 2022. It can be seen that, there was a noticeable growth during late 2017 till early 2018, after which it showed comparatively less fluctuations until mid of 2019. From 2019 mid till mid of 2020, there was a little fluctuation. However, during the end of 2021 till date the price has increased drastically exhibiting high fluctuations in its price. The below *figure 1* also shows that year 2021 has recorded all time high price in the month of November. According to Raynor, this particular rise(in November) is price is due to the launch of BTC ETF in the US [40]. However, the other rise in recorded in 2021 were because of the events involving Tesla's announcement about acquiring 1.5Billion USD's worth of digital coin [40]. This proves that Bitcoin shows high volatility in the financial market.

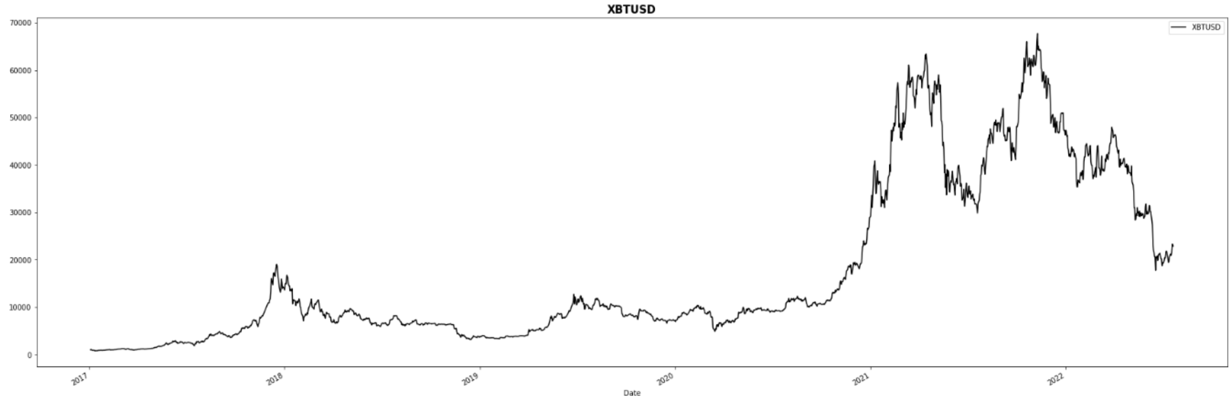
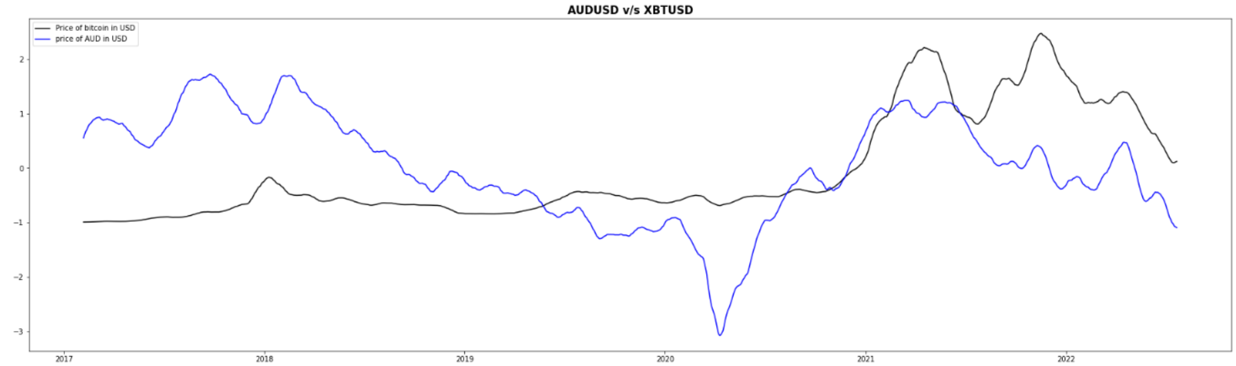


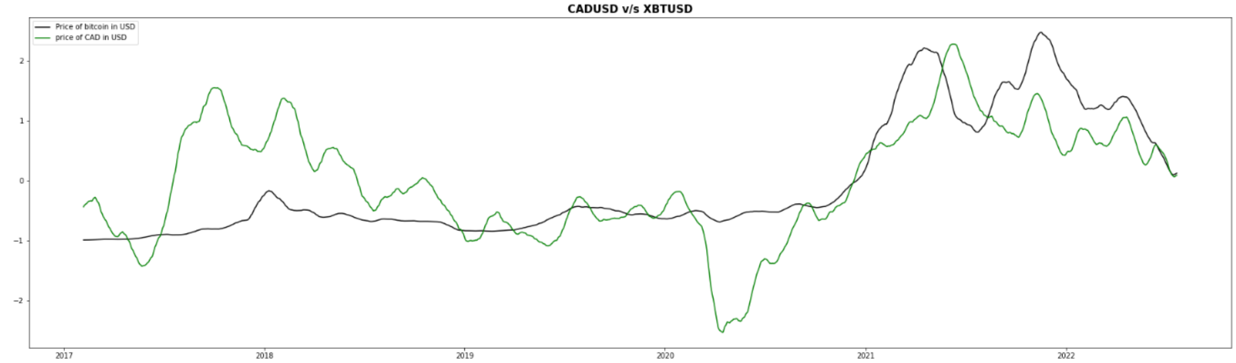
Figure 1: Bitcoin price over the time

- ***Bitcoin versus other currency patterns:***

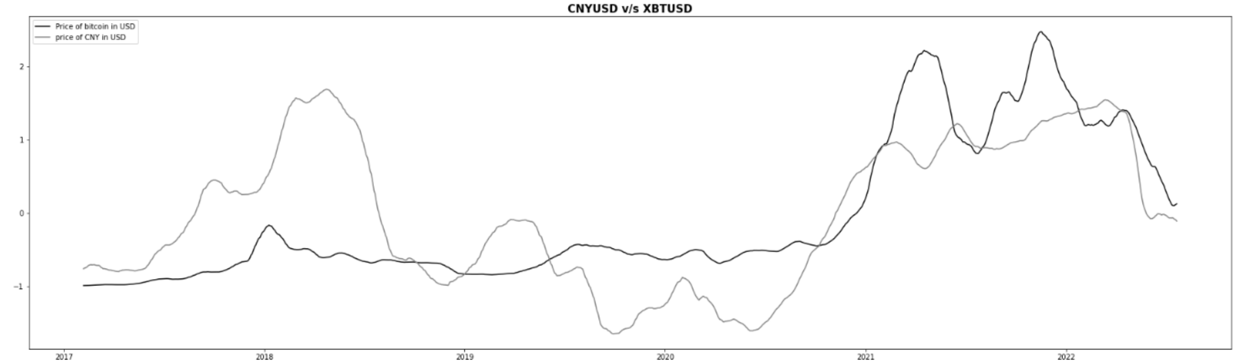
To visually understand the movement of bitcoin prices against other currencies, the pair data (BTC vs a currency pair) is plotted against the date. However, the magnitude of the bitcoin price is huge compared to other currency rates in the USD. To make the plot look visually better, the entire data is scaled using a standard scaler and smoothened, and then the currencies are plotted coupled with the bitcoin price. We can see in the below figures, how the price of bitcoin varies when compared to other currency prices in USD.



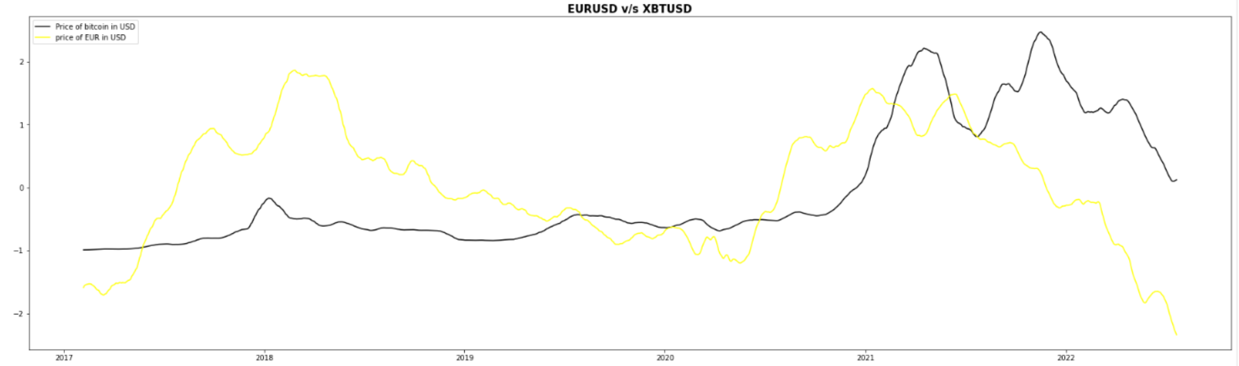
AUDUSD v/s XBTUSD



CADUSD v/s XBTUSD



CNYUSD v/s XBTUSD



EURUSD v/s XBTUSD

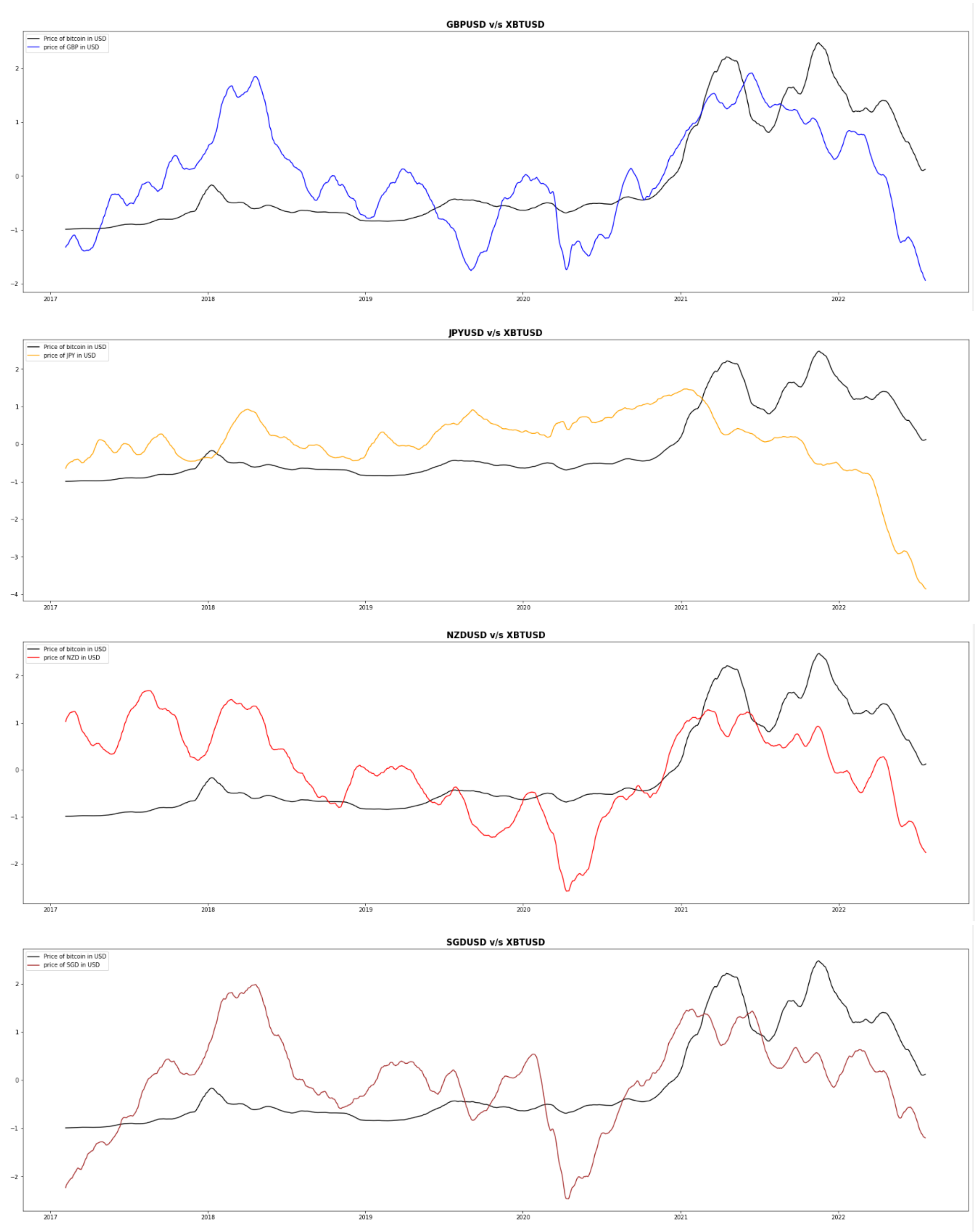


Figure 2: Bitcoin price v/s other currencies

The above graphs show the variation of bitcoin price in comparison to other currency rates. In most of the cases, the price movement of bitcoin and other currencies shows similarity during end of 2020 or early 2021 where the prices are rising. However, the movement is contrast in case of JPY currency during this period.

- ***Correlation among different features:***

Correlation is a statistical term that helps to understand how a group of variables are related to each other. The correlation coefficient value ranging from -1 to +1, determines the existence of the linear relationship between the variables. If the coefficient value is close to +1 between two variables, then those two variables are positively related, meaning the increase in the value of one variable increases the other variable. The coefficient value of 0 means there is no significant relationship between the two groups. The coefficient value -1 means the two variables are inversely related; meaning increasing the value of one variable decreases the value of another variable.

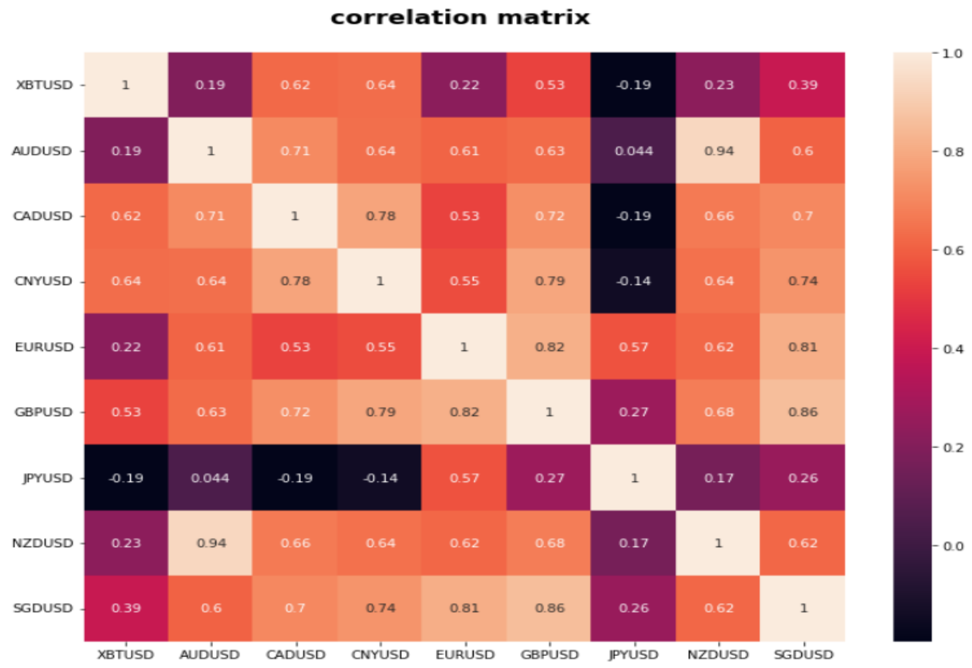


Figure 3: Correlation matrix for various features considered in the experiment.

The above *Figure 3* shows the correlation co-efficient matrix for all the features considered in this research. We can see that, the co-efficient values for [XBTUSD, CADUSD], [XBTUSD,CNYUSD] and [XBTUSD, GBPUSD] are greater than 0.5 with values 0.62, 0.64 and 0.53 respectively. This signifies that these currency pairs are positively corelated and there exists positive linear movement among these pairs. The other currencies have less than 0.5 co-efficient value varying around 0.1 to 0.3 which tells that their relationship between XBTUSD is nearly neutral and [XBTUSD,JPYUSD] has negative coefficient value.

- ***Autocorrelation and partial autocorrelation of the time series:***

Autocorrelation: It is a term used to determine the strength of relations between the time series using past data[41]. In this, the correlation of each time-series datapoint is calculated based on previous time steps or lags; meaning, the correlation coefficients of each time data point are calculated with values of the same series at the previous datapoint with respect to time.

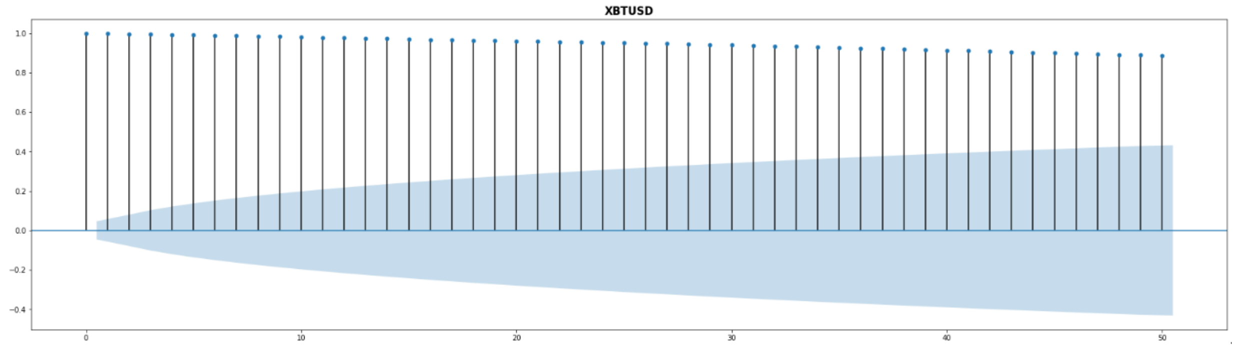


Figure 4: Autocorrelation graph for bitcoin price

The above *figure 4* shows the autocorrelation for bitcoin time series data and we can see that there exists a high autocorrelation in bitcoin time series. This is similar in case of other currency time series data as well.

Partial autocorrelation: Partial autocorrelation summarizes the relationship of a datapoint in time series with previous time step observations but without considering the effects of lag in the time series [41].

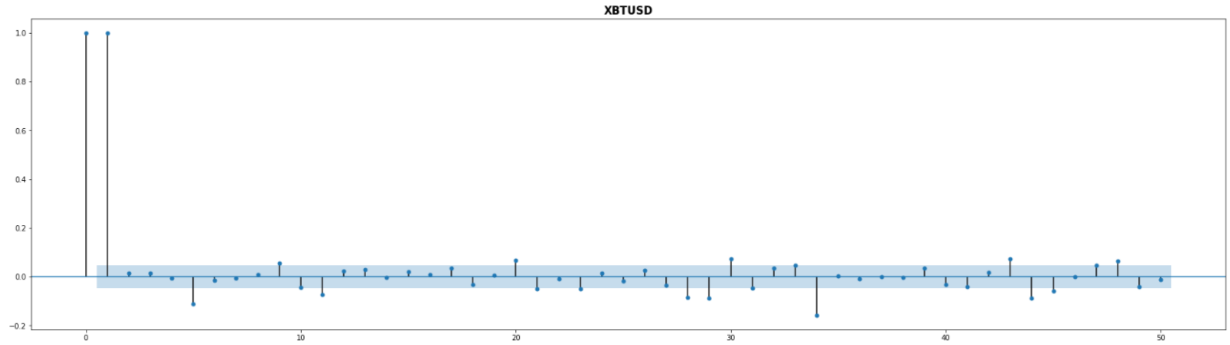
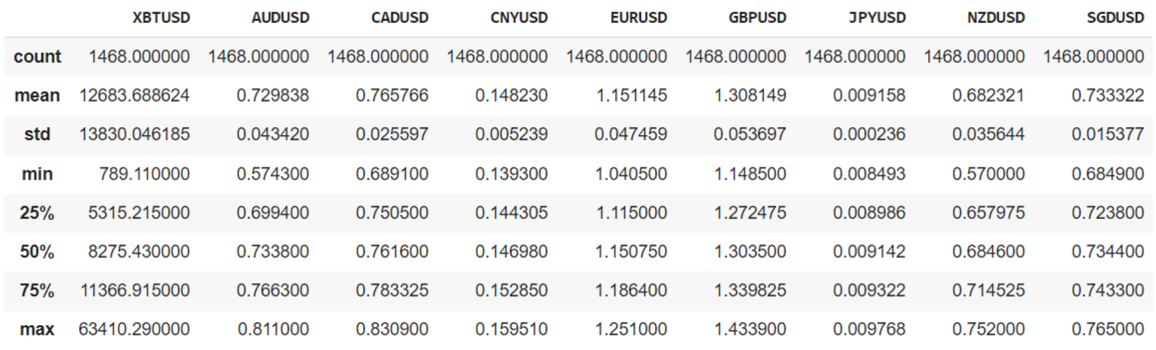


Figure 5: Partial autocorrelation graph

The *figure 5* shows that, in case of XBTUSD time series, the correlation without the effect of lags is dropped after the lag1. The same is observed in case of other currency time series as well.

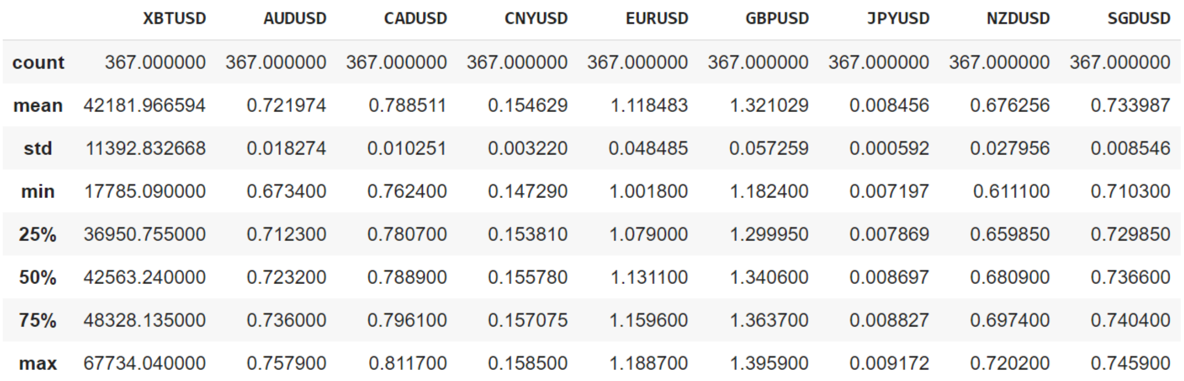
- ***Descriptive statistics:***

The descriptive statistic gives information about the quantitative summary of the data samples in terms of mean, median, max, min and standard deviation etc. *Figure 6 & 7* shows the descriptive statistics of training and testing dataset respectively.



	XBTUSD	AUDUSD	CADUSD	CNYUSD	EURUSD	GBPUSD	JPYUSD	NZDUSD	SGDUSD
count	1468.000000	1468.000000	1468.000000	1468.000000	1468.000000	1468.000000	1468.000000	1468.000000	1468.000000
mean	12683.688624	0.729838	0.765766	0.148230	1.151145	1.308149	0.009158	0.682321	0.733322
std	13830.046185	0.043420	0.025597	0.005239	0.047459	0.053697	0.000236	0.035644	0.015377
min	789.110000	0.574300	0.689100	0.139300	1.040500	1.148500	0.008493	0.570000	0.684900
25%	5315.215000	0.699400	0.750500	0.144305	1.115000	1.272475	0.008986	0.657975	0.723800
50%	8275.430000	0.733800	0.761600	0.146980	1.150750	1.303500	0.009142	0.684600	0.734400
75%	11366.915000	0.766300	0.783325	0.152850	1.186400	1.339825	0.009322	0.714525	0.743300
max	63410.290000	0.811000	0.830900	0.159510	1.251000	1.433900	0.009768	0.752000	0.765000

Figure 6: Descriptive statistics of Training Data



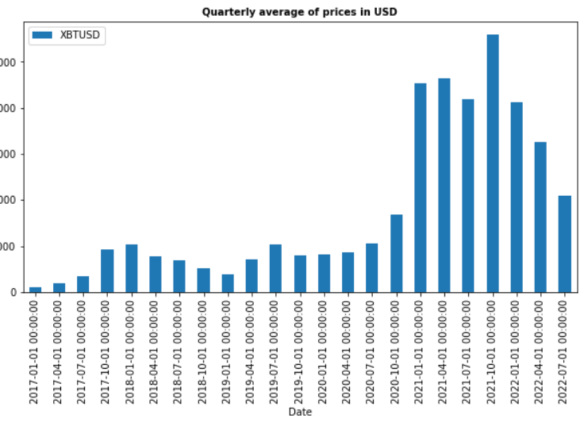
	XBTUSD	AUDUSD	CADUSD	CNYUSD	EURUSD	GBPUSD	JPYUSD	NZDUSD	SGDUSD
count	367.000000	367.000000	367.000000	367.000000	367.000000	367.000000	367.000000	367.000000	367.000000
mean	42181.966594	0.721974	0.788511	0.154629	1.118483	1.321029	0.008456	0.676256	0.733987
std	11392.832668	0.018274	0.010251	0.003220	0.048485	0.057259	0.000592	0.027956	0.008546
min	17785.090000	0.673400	0.762400	0.147290	1.001800	1.182400	0.007197	0.611100	0.710300
25%	36950.755000	0.712300	0.780700	0.153810	1.079000	1.299950	0.007869	0.659850	0.729850
50%	42563.240000	0.723200	0.788900	0.155780	1.131100	1.340600	0.008697	0.680900	0.736600
75%	48328.135000	0.736000	0.796100	0.157075	1.159600	1.363700	0.008827	0.697400	0.740400
max	67734.040000	0.757900	0.811700	0.158500	1.188700	1.395900	0.009172	0.720200	0.745900

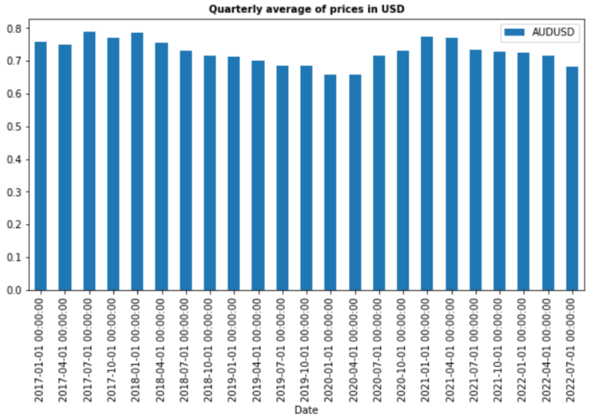
Figure 7: Descriptive statistics of Test data

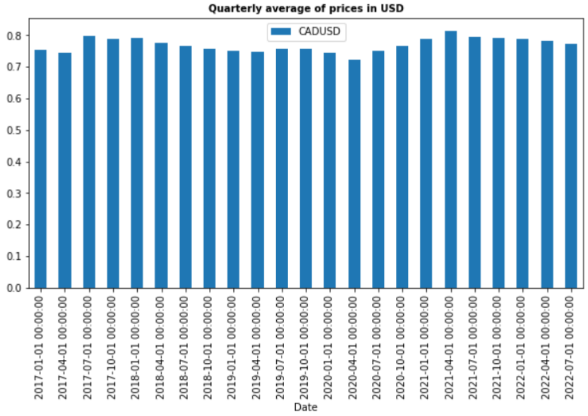
The test and train dataset consists of daily observations from Jan 2017 till mid July 2021 and mid July 2021 till mid July 2022. The bitcoin prices were comparatively less during 2017 till 2021 and hence we can see that the mean is comparatively less in train dataset. However, since the bitcoin price spiked during 2021 and showed all time high price during this time, the test set having the data for recent days has higher mean value compared to train data.

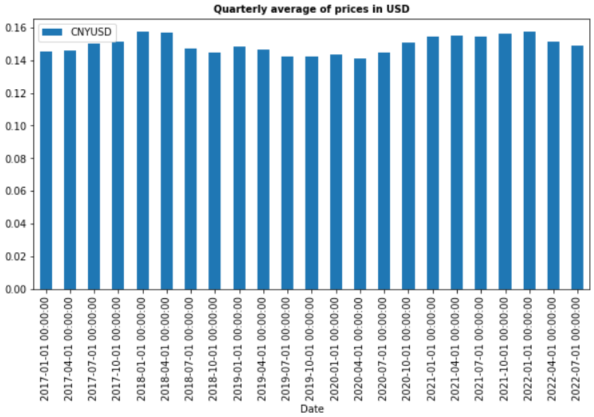
- ***Quarterly mean:***

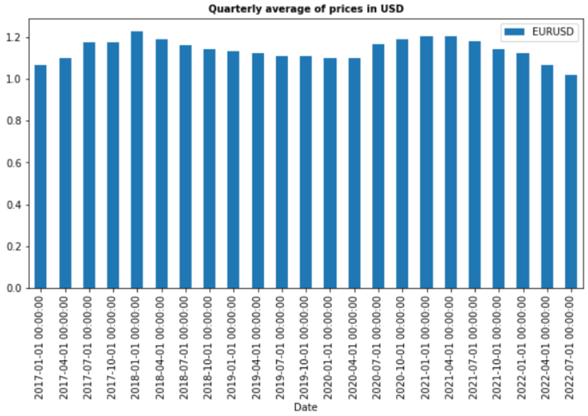
The below graphs show the quarterly mean for the entire dataset. The quarterly mean is varying over the time for all features which signifies the data is not stationary.

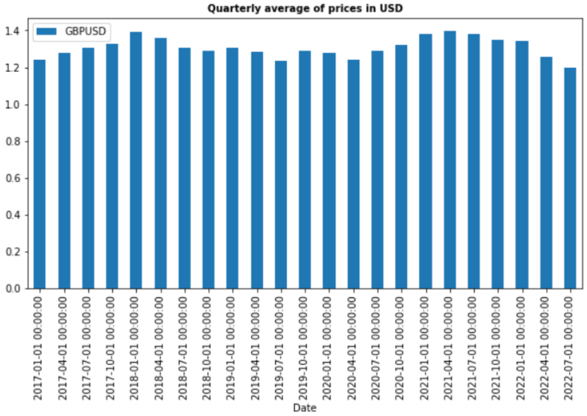

Quarterly average of prices in USD
XBTUSD
0
10000
20000
30000
40000
50000
2017-01-01 00:00:00
2017-04-01 00:00:00
2017-07-01 00:00:00
2017-10-01 00:00:00
2018-01-01 00:00:00
2018-04-01 00:00:00
2018-07-01 00:00:00
2018-10-01 00:00:00
2019-01-01 00:00:00
2019-04-01 00:00:00
2019-07-01 00:00:00
2019-10-01 00:00:00
2020-01-01 00:00:00
2020-04-01 00:00:00
2020-07-01 00:00:00
2020-10-01 00:00:00
2021-01-01 00:00:00
2021-04-01 00:00:00
2021-07-01 00:00:00
2021-10-01 00:00:00
2022-01-01 00:00:00
2022-04-01 00:00:00
2022-07-01 00:00:00
Date


Quarterly average of prices in USD
AUDUSD
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2017-01-01 00:00:00
2017-04-01 00:00:00
2017-07-01 00:00:00
2017-10-01 00:00:00
2018-01-01 00:00:00
2018-04-01 00:00:00
2018-07-01 00:00:00
2018-10-01 00:00:00
2019-01-01 00:00:00
2019-04-01 00:00:00
2019-07-01 00:00:00
2019-10-01 00:00:00
2020-01-01 00:00:00
2020-04-01 00:00:00
2020-07-01 00:00:00
2020-10-01 00:00:00
2021-01-01 00:00:00
2021-04-01 00:00:00
2021-07-01 00:00:00
2021-10-01 00:00:00
2022-01-01 00:00:00
2022-04-01 00:00:00
2022-07-01 00:00:00
Date


Quarterly average of prices in USD
CADUSD
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2017-01-01 00:00:00
2017-04-01 00:00:00
2017-07-01 00:00:00
2017-10-01 00:00:00
2018-01-01 00:00:00
2018-04-01 00:00:00
2018-07-01 00:00:00
2018-10-01 00:00:00
2019-01-01 00:00:00
2019-04-01 00:00:00
2019-07-01 00:00:00
2019-10-01 00:00:00
2020-01-01 00:00:00
2020-04-01 00:00:00
2020-07-01 00:00:00
2020-10-01 00:00:00
2021-01-01 00:00:00
2021-04-01 00:00:00
2021-07-01 00:00:00
2021-10-01 00:00:00
2022-01-01 00:00:00
2022-04-01 00:00:00
2022-07-01 00:00:00
Date


Quarterly average of prices in USD
CNYUSD
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
2017-01-01 00:00:00
2017-04-01 00:00:00
2017-07-01 00:00:00
2017-10-01 00:00:00
2018-01-01 00:00:00
2018-04-01 00:00:00
2018-07-01 00:00:00
2018-10-01 00:00:00
2019-01-01 00:00:00
2019-04-01 00:00:00
2019-07-01 00:00:00
2019-10-01 00:00:00
2020-01-01 00:00:00
2020-04-01 00:00:00
2020-07-01 00:00:00
2020-10-01 00:00:00
2021-01-01 00:00:00
2021-04-01 00:00:00
2021-07-01 00:00:00
2021-10-01 00:00:00
2022-01-01 00:00:00
2022-04-01 00:00:00
2022-07-01 00:00:00
Date


Quarterly average of prices in USD
EURUSD
0.0
0.2
0.4
0.6
0.8
1.0
1.2
2017-01-01 00:00:00
2017-04-01 00:00:00
2017-07-01 00:00:00
2017-10-01 00:00:00
2018-01-01 00:00:00
2018-04-01 00:00:00
2018-07-01 00:00:00
2018-10-01 00:00:00
2019-01-01 00:00:00
2019-04-01 00:00:00
2019-07-01 00:00:00
2019-10-01 00:00:00
2020-01-01 00:00:00
2020-04-01 00:00:00
2020-07-01 00:00:00
2020-10-01 00:00:00
2021-01-01 00:00:00
2021-04-01 00:00:00
2021-07-01 00:00:00
2021-10-01 00:00:00
2022-01-01 00:00:00
2022-04-01 00:00:00
2022-07-01 00:00:00
Date


Quarterly average of prices in USD
GBPUSD
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
2017-01-01 00:00:00
2017-04-01 00:00:00
2017-07-01 00:00:00
2017-10-01 00:00:00
2018-01-01 00:00:00
2018-04-01 00:00:00
2018-07-01 00:00:00
2018-10-01 00:00:00
2019-01-01 00:00:00
2019-04-01 00:00:00
2019-07-01 00:00:00
2019-10-01 00:00:00
2020-01-01 00:00:00
2020-04-01 00:00:00
2020-07-01 00:00:00
2020-10-01 00:00:00
2021-01-01 00:00:00
2021-04-01 00:00:00
2021-07-01 00:00:00
2021-10-01 00:00:00
2022-01-01 00:00:00
2022-04-01 00:00:00
2022-07-01 00:00:00
Date

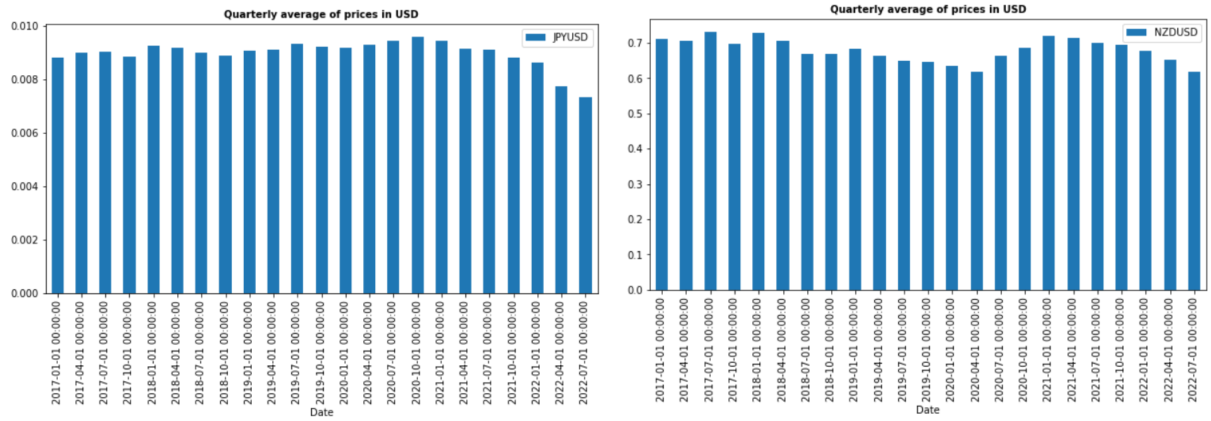


Figure 8: Quarterly average prices

Figure 8 shows that all the currency rates are varying over the time. This fluctuation is extremely more in case of bitcoin price compared to other currencies and started spiking during last two quarters(Q3 andQ4) of 2020. During 2021, it remained high with highest price recording in Q4 of 2021. From then, the price started falling during Q1,Q2 and Q3 of 2022. The variation of mean over the time in time series data also signifies that the data is non-stationary.

8 Methodology:

This section describes the proposed method to carry out the development of models, various algorithms used, hyperparameter tuning used for the algorithms, and the performance measures used to evaluate the models.

8.1 Proposed method:

The primary goal of the research is to build a predictive model for forecasting bitcoin prices based on exchange rates of other currencies. All these data are time series data which means the data is sequentially collected at equal intervals of time. Prediction of time series is different from normal supervised prediction because the former one considers date or 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 first step in time series prediction is to explore the data to understand the underlying patterns, seasonality, and trends. This is an important step if we are implementing a machine learning model based on statistical methods like variation of ARMA/ARIMA. However, the good news is that the time series can be converted to a data form which can be used in any supervised learning models for regression problems by shifting the features. In this study, this approach is used to convert and feed the data to supervised ML models by focusing on the multivariate prediction models to forecast the bitcoin price. The below *figure 9* shows the overview of the methodology used in building each model built as part of this study.

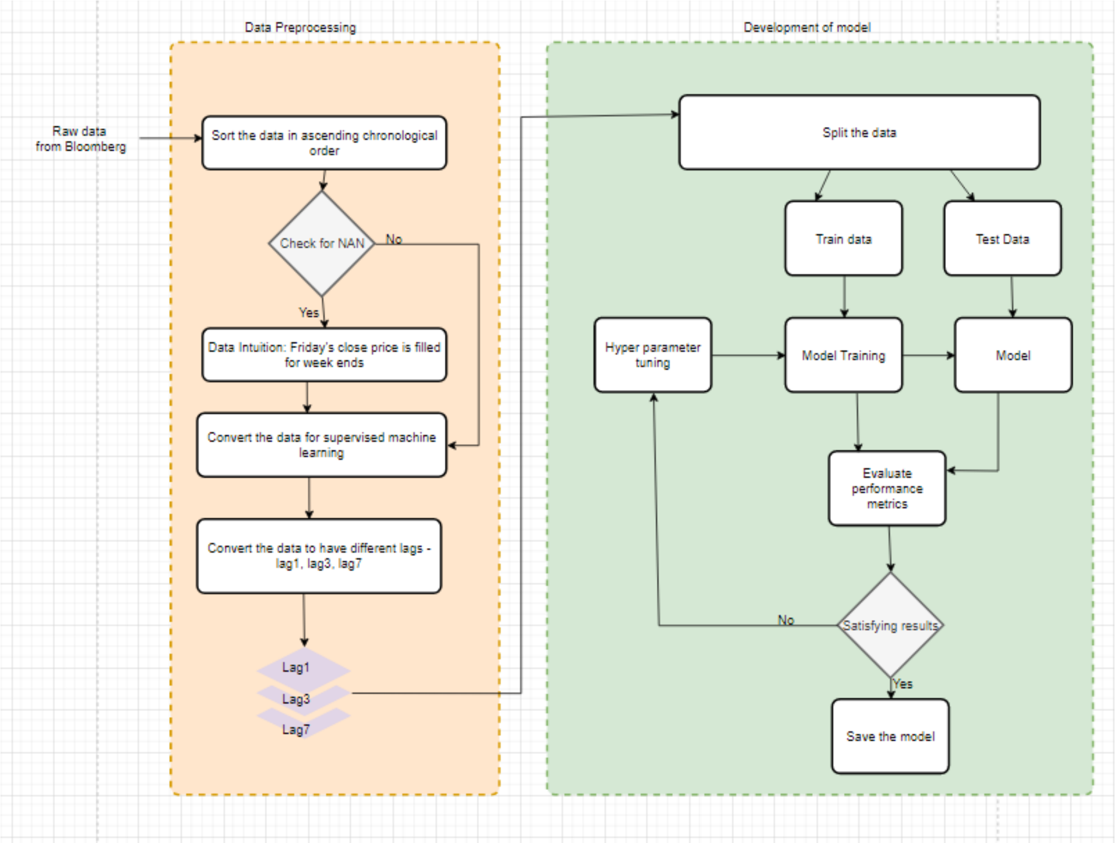

Data Preprocessing
Development of model
Raw data from Bloomberg
Sort the data in ascending chronological order
Check for NAN
No
Yes
Data Intuition: Friday's close price is filled for week ends
Convert the data for supervised machine learning
Convert the data to have different lags - lag1, lag3, lag7
Lag1
Lag3
Lag7
Split the data
Train data
Test Data
Hyper parameter tuning
Model Training
Model
Evaluate performance metrics
Satisfying results
No
Yes
Save the model

Figure 9: Proposed methodology

In this study, as part of data preprocessing, the data is cleaned to intuit NaNs observed on weekends with corresponding Friday's close price, then entire time series is converted to have lags which is used as data for supervised learning. The different lags used are lag1(t-1), lag3(t-1,t-2,t-3) and lag7(t-1,t-2….t-7). This data is split into training and testing dataset. A base model is built for the training data and performance is verified for the test dataset. Later, hyperparameter optimization is applied and a new variant is built to have an optimal model. The model is again evaluated using the performance metrics.

8.2 Algorithms:

This section explains the various machine learning algorithms used in the study to developer the prediction models.

8.2.1 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 [42] and supports multiple programming languages like python, julia and R. It is an end-to-end tree-based machine learning model which is highly scalable. Due to its scalability, it provides a fast and accurate way of solving data science problems by using the parallel tree boosting technique [42]. Some of the features of XGBoost that contribute to its fast processing are cache optimization, parallelization, and out of memory(out of core) 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 [43]. 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 controls how aggressively a model is trained. This parameter helps in controlling the effect of outliers on the prediction and also helps in preventing the overfitting of the model.
- Gamma: This is the parameter that helps in auto pruning the tree. This parameter is compared with gain (which is computed by subtracting the similarity score of a branch before spilt from the similarity score of a branch after split) to take the decision on pruning the tree. Meaning, the tree splitting is stopped if the gamma value is more than the gain value. A lesser gamma value signifies that the model is trained with less aggressive approach.
- Eta: this is an analogous to learning rate. The default value used is 0.3, however the typical value can vary from 0.01 to 0.2.
- Min_child_weight: Minimum total of all observations' weights required in a child which is used to control the overfitting. Too high value may result is under fitting.
- Max_depth: the depth of the tree used for controlling the overfitting as a very high depth of tree may result in learning relations specific to a particular sample.
- Booster: This is the type of model used in each iteration. It can be gbtree for parallel tree-based models and gblinear for linear or sequential decision tree models.

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.

Below *figure 10* explains the general architecture of XGBoost algorithm. In XGBoost, each time a model is built, the residuals are passed as a target parameter to the next model to build a stronger learner, thereby reducing the error in each iteration. The base learner is built by using the mean of a target for each data point and the initial residual is passed for the next iteration as the target label. Likewise, a stronger learner is built is using weak or base learners.

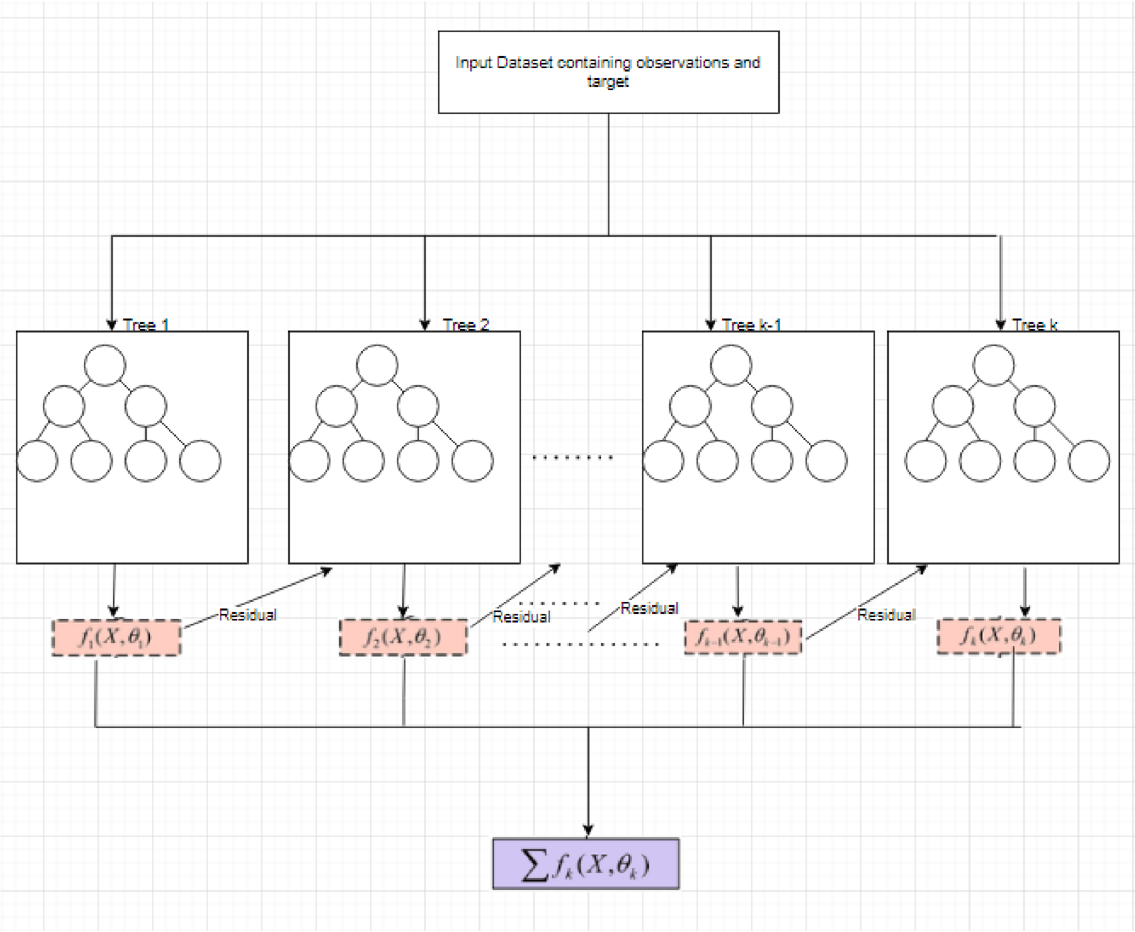


Figure 10: Generic architecture diagram of XGBoost algorithm.

8.2.2 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 advantages of random forest algorithm are:

- The decision trees are built based on the above-mentioned bootstrap sampling or bagging technique(row samples and column samples, with replacement) that reduces the high variance of the model (High variance means the model underperforms on unseen or test data compared to tarin data). Since the features are also randomly sampled, it suppresses the effect of strongest feature on the target thereby reducing the bias. Usually, 2/3rd of observations is considered as row samples and square root of number of columns is considered as column samples for each decision tree.
- Since each tree is created independently using bootstrap samples, it makes full use of CPU to build random forest.

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.

The below *figure 11* shows the architecture of random forest algorithm. The data is sampled based on the parameters(max_features and min_sample_split) passed and decision trees are created. The predictions from each decision tree are aggregated to get the final predictions of random forest regression model.

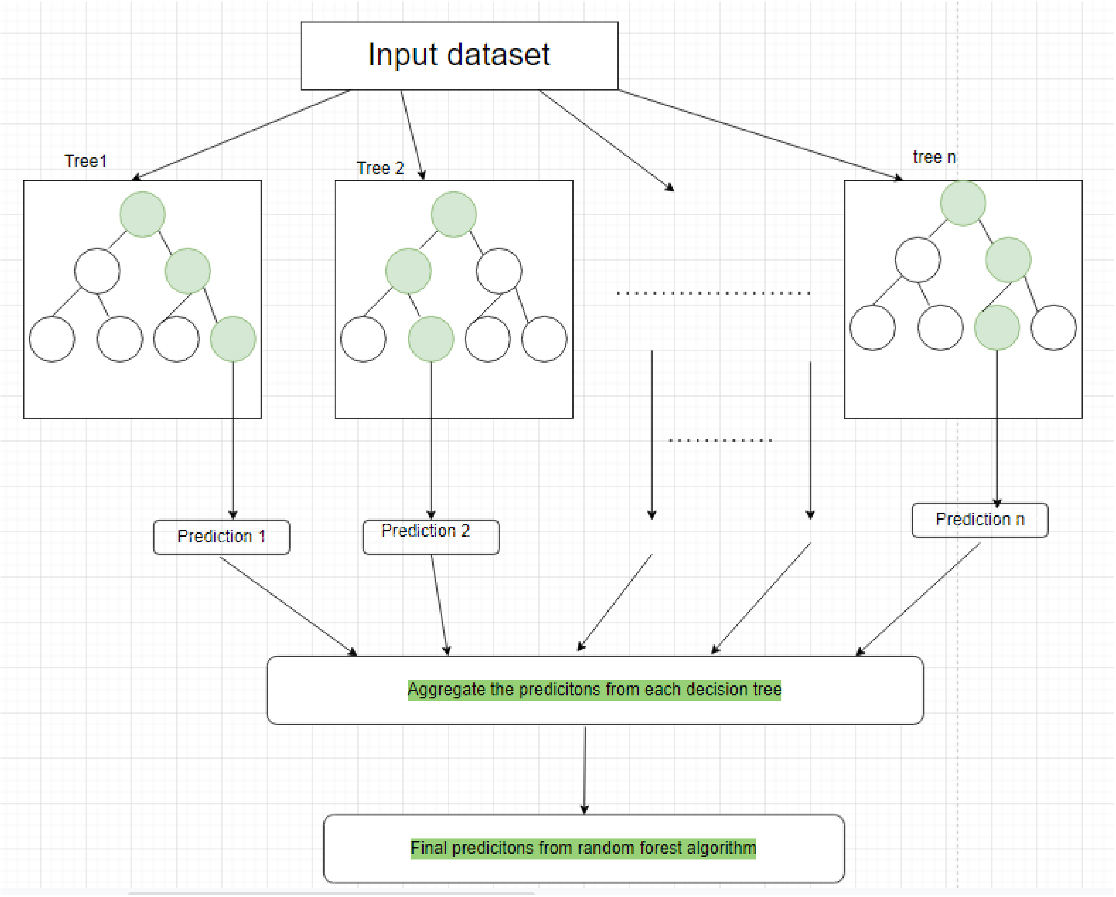

Input dataset
Tree1
Tree 2
tree n
Prediction 1
Prediction 2
Prediction n
Aggregate the predicitons from each decision tree
Final predicitons from random forest algorithm

Figure 11: Generic architecture diagram of Random Forest algorithm.

8.2.3 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 hyperplanes which are drawn by passing through the data points that are nearest to the hyperplane on both the sides. These hyperplanes are also called as positive and negative margin, or 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. All these can be visually understood from the below *figure 12*.

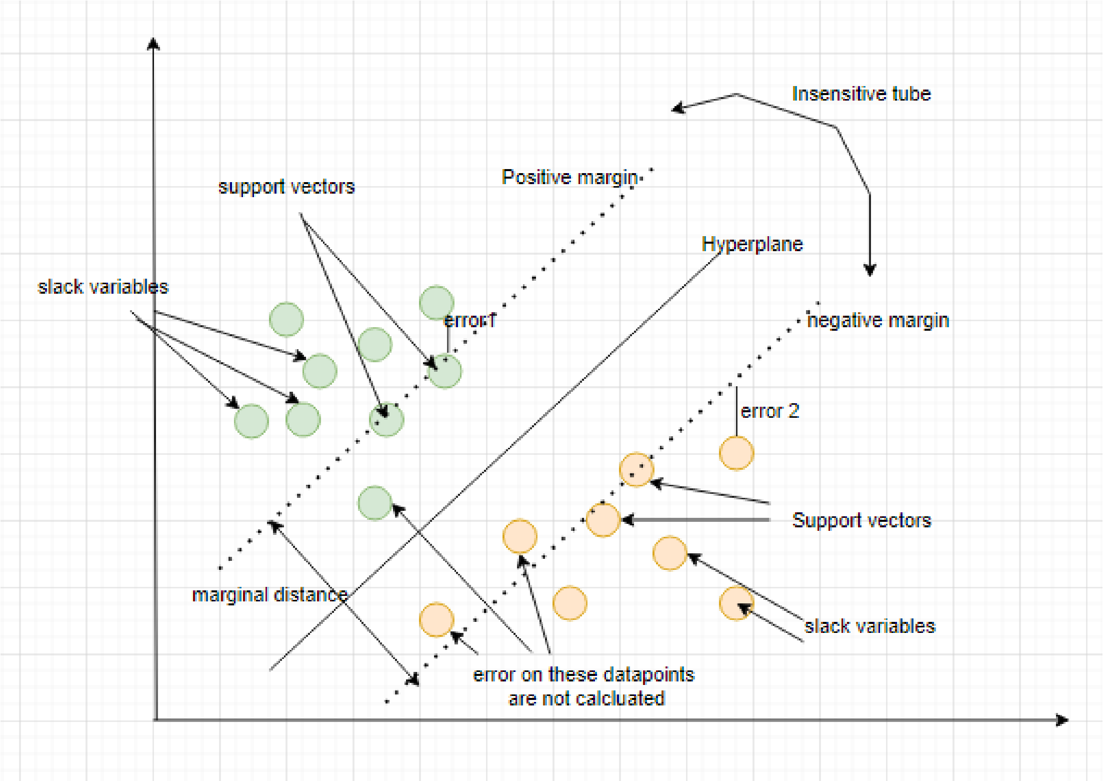


Figure 12: Support vector regression intuition.

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.

$(wx_i+b) - y_i \leq \varepsilon+\xi^*$ *and* $y_i-(wx_i+b) \leq \varepsilon+\xi$; 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,

$$y_i - wx_i - b \leq \varepsilon + \xi_i$$
$$wx_i + b - y_i \leq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

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 [44].

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 insensitive tube or epsilon tube on either side of hyperplane formed. Thus, this parameter is also known as error sensitivity parameter.

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.

8.2.4 LSTM – Long Short-Term Memory:

LSTM or long Short Term Memory is a deep learning model proposed by Hochreiter & Schmidhuber in the year 1997 [45]. It is a Gated Recurrent Neural Network which has an ability to store the information for future cell processing. So, it is a neural network with memory having two key vectors: 1. Short term state vector that keeps the output at the current time step and 2. Long term state that processes the information for long term while progressing through the network.

LSTM has a recurrent structure or multiple copies of same network where each copy passes an information to the successor one as shows in figure 5. LSTM makes use of memory blocks which are connected through layers, instead of having neurons. 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 below figure shows the design behind the LSTM algorithm.

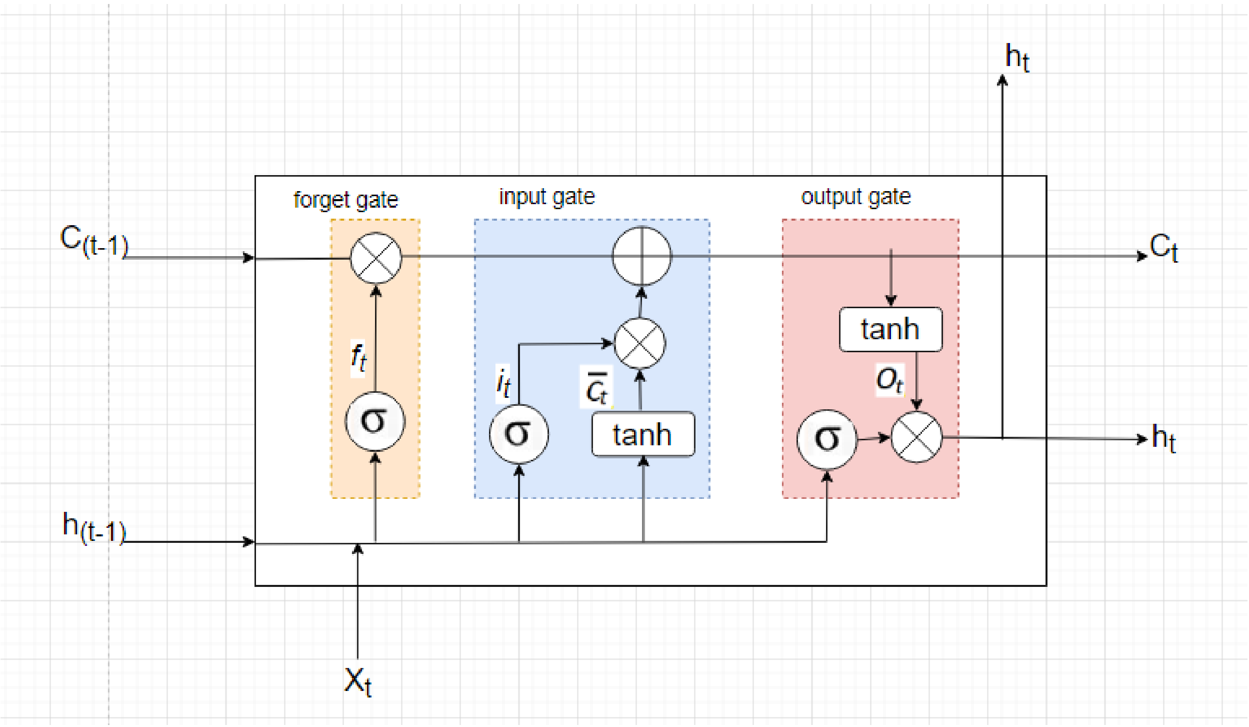


Figure 13: Long Sort Term memory module

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[46]. 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 [46] .

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

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

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 [46].

The entire explanation for LSTM is referred from the blog : https://colah.github.io/posts/2015-08-Understanding-LSTMs/

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

8.3 Hyperparameter Tuning:

The parameters that define an architecture of a machine learning model are called as hyperparameters. The process of searching the best parameter to build an optimal model is referred as hyperparameter tuning. In this case, a model is asked to perform the exploration of parameters and select the best one to have the best model. Thus, the hyperparameters are not the model's parameters, but these are something that a model learns during the training process to structure a good model. This is an important step while building any machine learning model as it increases the performance of the model by fine tuning the parameters on the data.

Below are the generic steps involved in hyperparameter tuning.

- Define a base model using ideal parameters.
- Create a dictionary containing list of range of possible parameters for all hyperparameters that the model uses.
- Pass the dictionary created in previous step to a built-in search method like randomsearch or gridsearch.
- Fit the model built in previous step on the training dataset.
- Get the list of optimal parameters suggested by above model.
- Pass these parameters in the final model.

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. Conversely, the gridsearch algorithm uses all the possible combination of hyperparameters in the learning process to give the best one. However, the studies [47] show that random search is more efficient compared to grid search in hyperparameter optimization.

Below are the parameters used for each algorithm after optimization using random search.

XGBoost: Below are the parameter set after tuning for each lag for XGBoost algorithm.

Hyperparameter	Values(lag1)	Values(lag3)	Values(lag7)
base_score	0.75	0.5	0.5
booster	gblinear	gblinear	gblinear
learning_rate	0.4	0.4	0.4
max_depth	5	5	5
min_child_weight	6	3	3
n_estimators	500	2500	2500

Random Forest: Below table shows the hyperparameters set for Random Forest algorithm for different lags considered.

Hyperparameter	Values(lag1)	Values(lag3)	Values(lag7)
max_depth	20	10	25
min_samples_split	4	2	2
n_estimators	900	2000	1000

Support Vector regression:

Hyperparameter	Values(lag1)	Values(lag3)	Values(lag7)
C	3500	4000	4000
epsilon	1e-07	0.01	0.01
degree	1	3	3
cache_size	100	800	800

LSTM:

Hyperparameter	Values(lag1)	Values(lag3)	Values(lag7)
Shuffle	False	False	False
Batch_size	8	8	8
Epochs	30	30	40
Layer1_unit	64	64	64
Layer2_unit	32	32	32
Layer3_unit	16	16	16

In case of LSTM, though parameters are tuned using random search, it still needed tweaking of parameters manually to increase the performance further.

8.4 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. There are many performance measures that can be chosen depending on the type and objective of a model. For example, for classification problems, the performance of the model is measure by accuracy or AUROC, and for regression problems, it measured by r-squared or root mean squared error. 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 by taking the absolute of mean of errors(the difference between the actual value and the predicted value) of each data points of prediction set. Mean absolute error 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 how accurately a model can forecast the data. In specific, it is a measure of loss of a model. It is the average of absolute percentage errors of a forecast. It is calculated using below formula.

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

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. The gives the measure of errors of a predictive model for continuous data. It is calculated by first finding the errors of each datapoint, then find average of squared errors and finally take the square root of resulted value. This gives RMSE. 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. It normalizes the value of RMSE between 0 and 1. The RMSE is scaled between maximum of true value and minimum of true value. 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 fit the data points and predict the dependent feature.

9 Results and discussion:

The results of all four models are captured in terms of the performance metrics such as mean absolute error, mean absolute percentage error, root mean squared error and scaled root mean squared error. Along with these performance measures, prediction graphs are also captured for all the models comparing with the actual values of the bitcoin prices. The below section explains results organized under two sections; first section demonstrates in-detail analysis of results for each lag considered in the experiment and the second section discusses the overall performance of the models.

9.1 Results for each lag:

The results captured for all four models for lag1 are summarized in the order of their performance with the better model on the top of the list in the *table 2*. This also has the performance measures captured before and after hyperparameter optimization for each model. The model with the best performance is highlighted.

Lag1					
Algorithm	**Hyperparameter tuning**	**MAE**	**MAPE(%)**	**RMSE**	**Scaled RMSE**
XGBoost	N	1800.807	4.497	2328.165	0.0466
	Y	**1051.628**	**2.5797**	**1475.9624**	**0.0295**
SVR	N	1663.3167	3.5917	2794.0275	0.0559
	Y	1383.9007	3.1379	2161.4242	0.0433
LSTM	N	2208.5275	5.2224	2573.0993	0.0515
	Y	1566.5123	3.7254	1926.3234	0.0386
Random Forest	N	1938.0568	4.8581	2390.3203	0.0479
	Y	1856.7038	4.6943	2294.4461	0.0478

Table 2:Results of models for Lag1

For the lag1 dataset, XGBoost algorithm performed better before and after parameter tuning. Before parameter tuning, it resulted in MAE of 1800.807, MAPE of 4.497%, RMSE of 2328.165 and scaled RMSE of 0.0466. When the hyperparameters are tuned, the MAPE and scaled RMSE are reduced to almost half of what they were before i.e., 2.5797% and 0.0295 respectively. MAE and RMSE also showed a significant decrease to 1051.628 and 1475.9624 respectively. XGBoost is followed by SVR in terms of performance with MAE 1663.3167, MAPE 0.0359%, and scaled RMSE 0.0559 and all these metrics are further reduced to 1383.9007, 0.0309%, and 0.0433 respectively after parameter tuning. However, in terms of RMSE, LSTM performed better with RMSE at 2573.0993 before tuning compared to SVR with RMSE at 2794.0275. After parameter tuning, RMSE is reduced to 1926.3234 in case of LSTM and 2161.4242 in case of SVR. Though Random forest showed better performance before parameter tuning with MAE 1938.0568, MAPE 4.8581%, RMSE 2390.3203 and scaled RMSE 0.479 compared to LSTM without parameter tuning with MAE 2208.5224, MAPE 5.2224, RMSE 2573.0993 and scaled RMSE 0.0515; LSTM performed better after parameter tuning with performance metrics MAE, MAPE, RMSE and scaled RMSE at 1556.5123, 3.7254%, 1926.3234 and 0.0386 respectively when compared to random forest model whose values are reported as1856.7038, 4.6943, 2294.4461 and 0.0478 respectively.

Below are the graphs captured for XGBoost before and after parameter tuning.

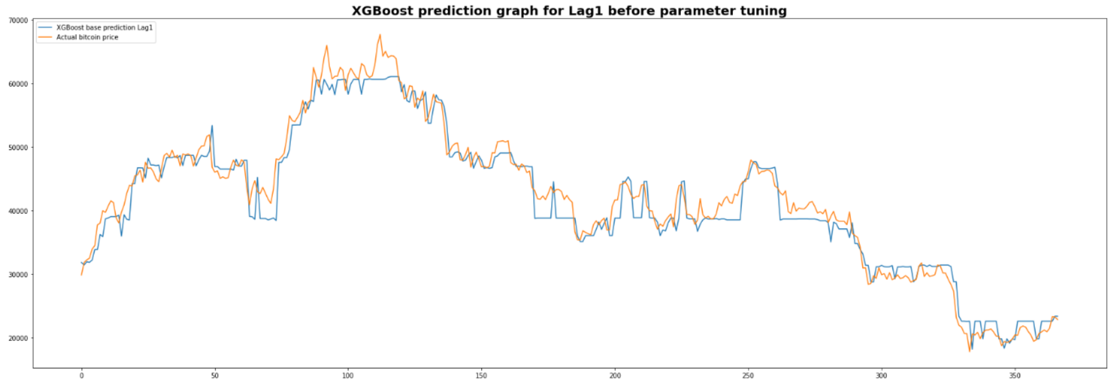


Figure 14: XGBoost - actual v/s prediction graph before parameter tuning

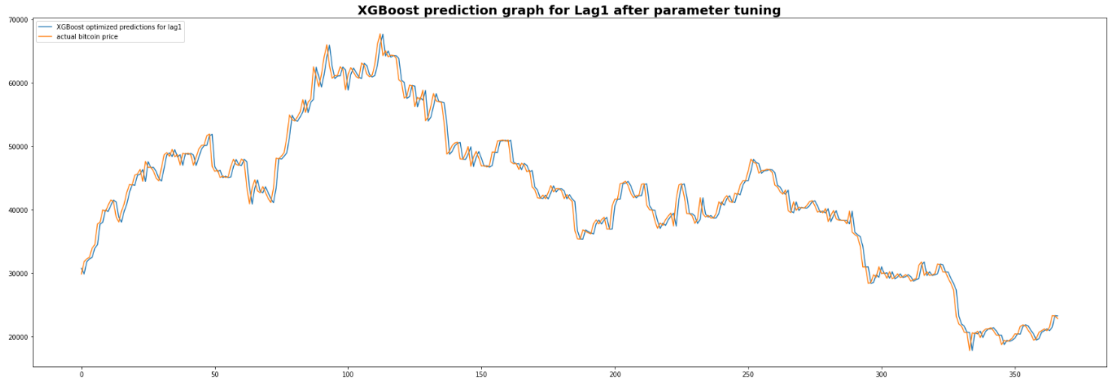


Figure 15: XGBoost: actual v/s predictions after parameter tuning for lag1

In *Figure 14*, there is a little difference observed between actual and predicted curves when the model is trained without applying parameter tuning. However, in *figure 15*, when the model is tested after parameter tuning, there is an improvement in capturing patterns; the actual and precited datapoints are almost overlapping or following each other which shows considerable improvement in terms of performance metrics as well. The graphical representation of the remaining models' performance is provided in the appendix section[**Performance graphs of other models for Lag1:**].

The results captured for lag3 are summarized in the below table. Similar to lag1, all the metrics are captured before and after parameter tuning of each model.

Lag3					
Algorithm	**Hyperparameter tuning**	**MAE**	**MAPE(%)**	**RMSE**	**Scaled RMSE**
XGBoost	N	1871.2265	4.8023	2396.4732	0.048
	Y	1093.0044	2.6802	1519.9507	0.0304
SVR	N	1854.7759	4.0892	2910.3755	0.0583
	Y	1515.0595	3.491	2236.684	0.0448
LSTM	N	2782.1207	6.5498	3213.5568	0.0643
	Y	1604.9684	3.9735	2313.5768	0.0423
Random Forest	N	2018.0528	5.111	2496.7564	0.05
	Y	1924.5841	4.9045	2400.4814	0.0481

Table 3: Performance metrics for lag3

In case of lag3 the order of performance of models is similar to that in lag1. XGBoost showed better performance compared to the other three models. The MAE, MAPE, RMSE and scaled RMSE for XGBoost without parameter tuning are 1871.2265,4.8023%, 2396.4732 and 0.048 respectively. All these values improved to 1093.0044, 2.6802%, 1519.9507 and 0.0304 after parameter optimization. The second model to show better performance than XGBoost is SVR which resulted in 1854.7759 MAE, 4.0892% MAPE, 2910.3755 and 0.0583 scaled RMSE. After parameter optimization, all these values are further reduced to 1515.0595 MAE, 3.491% MAPE, 2236.684 and 0.0448 scaled RMSE. Similar to lag1, though random forest showed better performance before parameter tuning with MAE 2018.0528, MAPE 5.111%, RMSE 2496.7564, and scaled RMSE 0.05 compared to LSTM with values 2782.1207, MAE, 6.5498% MAPE, 3213.5568 RMSE and 0.0643 scaled RMSE; LSTM performance was significantly improved after parameter tuning with values 1604.9684 MAE, 3.9735% MAPE, 2496.7564 RMSE and 0.05 scaled RMSE. Random forest showed a minimal difference in performance improvement after parameter tuning and its MAE, MAPE, RMSE and scaled RMSE are 1924.5841, 4.9045%, 2400.4814, and 0.0481 respectively.

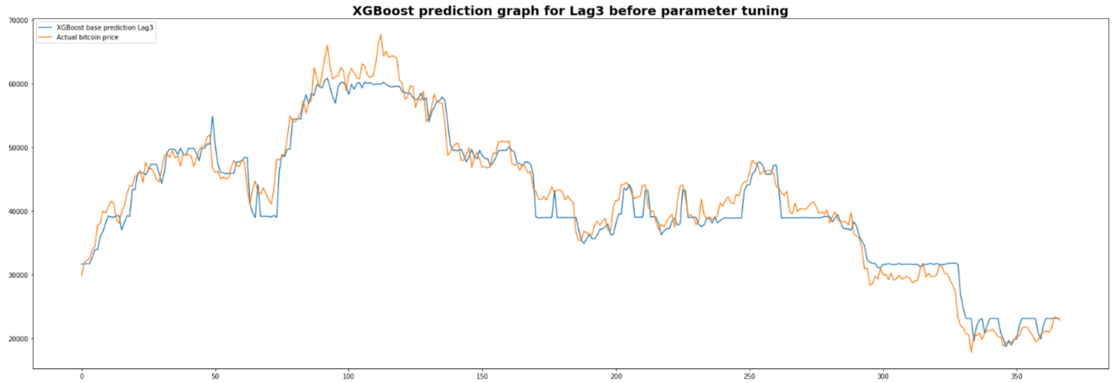


Figure 16: XGBoost prediction v/s actual for lag3 before parameter tuning

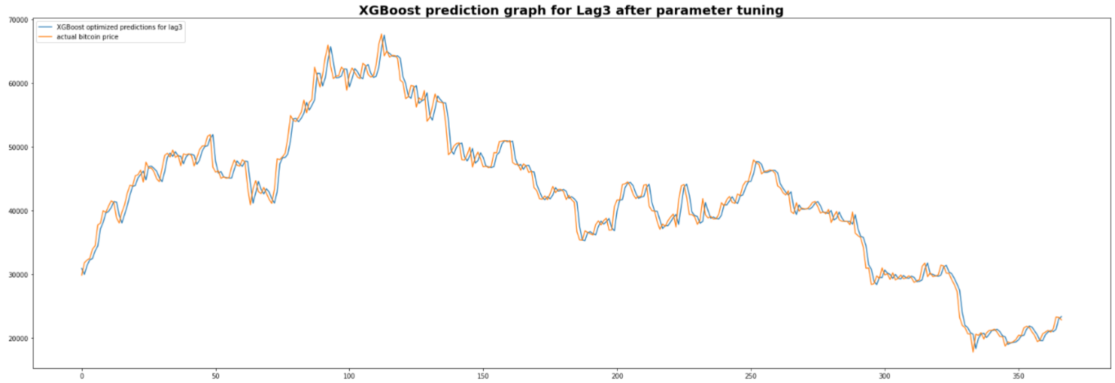


Figure 17: XGBoost prediction v/s actual for lag3 after parameter tuning

Figures 16 and 17 show the prediction graph for XGBoost model without and with optimization respectively. The predicted datapoints in the graph without tuning show little difference compared to actual data points. Whereas, in the graph captured for the optimized model, the predicted and actual data points of bitcoin price follow each other closely. The graphical representation of the remaining models' performance is provided in the appendix section[**Performance graphs of other models for Lag3:**].

Like lag1 and lag3, for lag7 also XGBoost performed better followed by SVR, LSTM and random forest algorithms. The performance measures for lag 7 are summarized in the below table for all models.

Lag7					
Algorithm	**Hyperparameter tuning**	**MAE**	**MAPE(%)**	**RMSE**	**Scaled RMSE**
XGBoost	N	2005.5028	5.2266	2590.4579	0.0519
	Y	**1111.3384**	**2.7165**	**1534.9107**	**0.0307**
SVR	N	2287.7362	5.1055	3334.3208	0.0668
	Y	1676.6268	3.8663	2407.6039	0.0482
LSTM	N	3774.286	9.0144	4319.0396	0.0865
	Y	2001.99	5.0681	2694.1189	0.0539
Random Forest	N	2269.9888	5.7289	2779.6151	0.0556
	Y	2251.7415	5.6312	2765.9959	0.0554

Table 4: Performance metrics for lag7

For lag7, XGBoost outperformed compared to other models with 2005.5028 MAE, 5.2266% MAPE, 2590.4579 RMSE and 0.0519 scaled RMSE. After parameter tuning, all these performance measures are decreased to 111.3384, 2.7165%, 1534.9107 and 0.0307 respectively. However, the models' performances are slightly different compared to lag1 and lag3 when the parameters are not optimized. In case of lag7, when parameters are not tuned, Random Forest performed better with values 2269.9888 MAE, 2779.6151 RMSE and 0.0556 scaled RMSE compared to SVR which showed slightly higher performance metrics with values 2287.7362 MAE, 3334.3208 RMSE and 0.0668 scaled RMSE and LSTM with values 3774.286 MAE, 2694.1189 RMSE and 0.865 scaled RMSE. Whereas when the performance is measured in terms of MAPE, SVR performed better with MAPE at 5.1055% compared to Random Forest with 5.6289% MAPE and LSTM with 9.0144%. After optimization, the order of performance of models is like lag1 and lag3. Optimized SVR performed better with MAE at 1676.6268, MAPE at 3.8663%, RMSE at 2407.6093 and scaled RMSE at 0.0482 compared to LSTM with the values at 2001.99, 5.0681%, 2694.1189 and 0.0539 respectively. Random forest reported more errors even after parameter tuning compared to the other three models with MAE, MAPE, RMSE and scaled RMSE at 22517415, 5.6312%,2765.9959 and 0.0554 respectively.

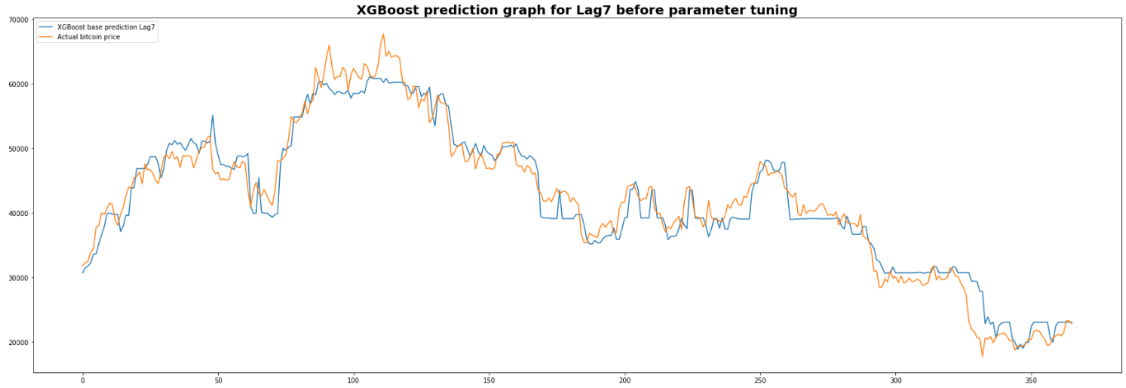


Figure 18: XGBoost prediction v/s actual for lag7 before parameter tuning

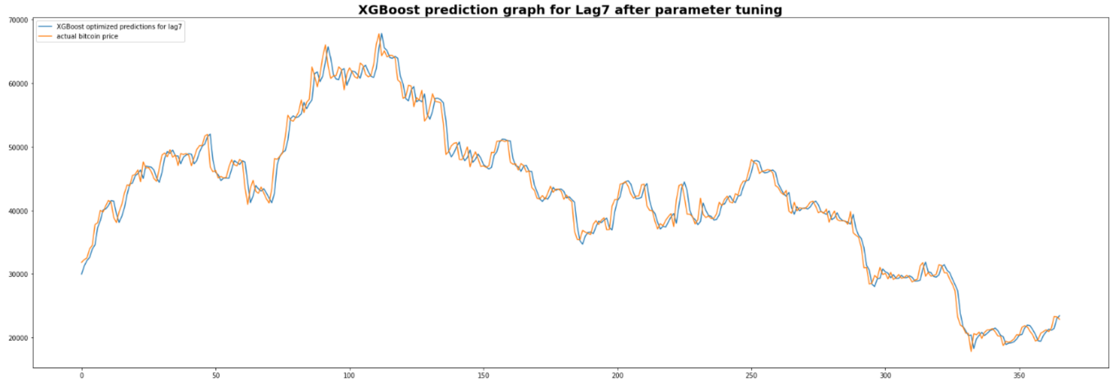


Figure 19: XGBoost prediction v/s actual for lag7 after parameter tuning

Figures 18 and 19 show the prediction graphs for XGBoost which performed the best for the given dataset. Graph plotted for optimized model shows minimal difference between actual and predicted price compared to graph of the model without optimization. The graphical representation of the remaining models' performance is provided in the appendix section[**Performance graphs of other models for Lag7:**].

9.2 Overall results:

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

Mean absolute Error:

The below graph shows how the MAE value varied for each lag and how the value is decreased after parameter optimization.

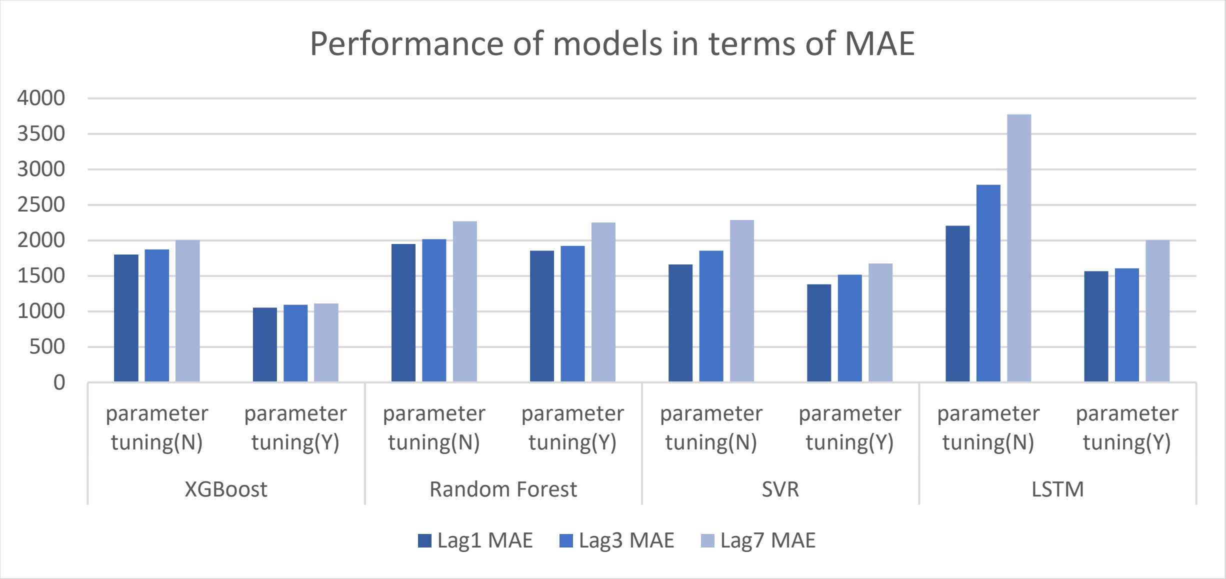


Figure 20: Performance of models in terms of MAE

The *figure 20* shows that all the models performed comparatively well for all lags after parameter tuning except random forest which resulted minimal difference in MAE after optimization. Out of all models, XGBoost and LSTM models showed significant decrease in MAE after optimization for all lags. These are followed by SVR which showed slightly less decrease in MAE after optimization compared to XGBoost and LSTM.

After optimization, all models performed better for lag1 and reported higher MAE values for lag7. However, XGBoost reported nearly equal MAEs for all three lags. XGBoost is followed by SVR in terms of MAE and then by LSTM with a slight difference compared to SVR. Random forest reported highest MAE compared to all other algorithms.

Mean Absolute Percentage Error:

The below chart shows the performance of models in terms of MAPE for all lags

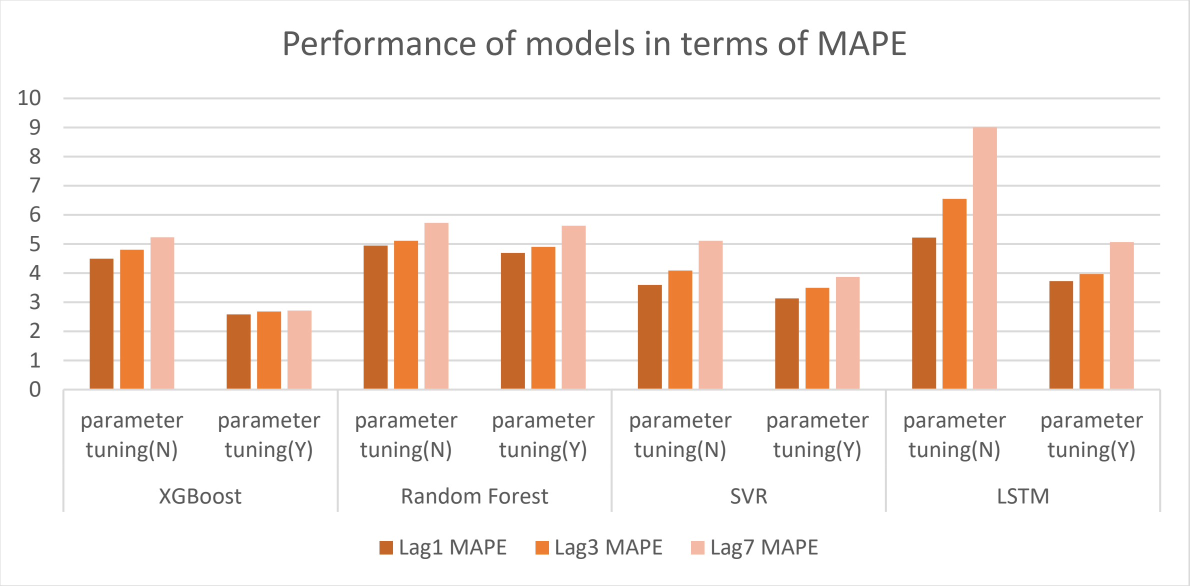


Figure 21: Performance of models in terms of MAPE

Figure 21 shows the performance of all models in terms of MAPE. All models showed significant decrease in error percentage after optimization except Random Forest which showed minimal difference as in case of MAE. All models reported less than 10% errors in test data forecasting. Though most of the models reported 3 to 3.5% of errors for lag1, XGBoost reported minimum error percentage as lowest as 2.5% for all three lags. XGBoost is followed in SVR and then LSTM in terms of MAPE after optimization. Random forest resulted in more error percentage compared to all remaining three models.

Root Mean Squared Error:

The below chart shows the performance of models in terms of RMSE.

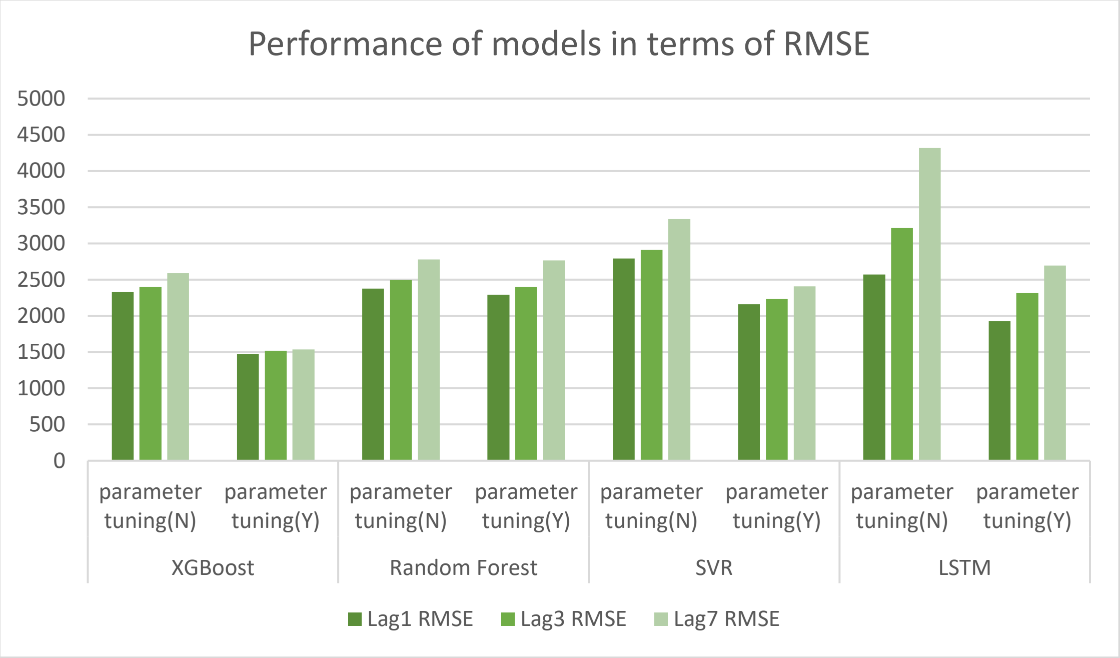


Figure 22: Performance of models in terms of RMSE

In case of RMSE as performance measure, all models showed significant decrease in RMSE after optimization except random forest. RMSE values after parameter tuning ranged from 1500 to less than 3000 for all models for all lags which is significantly less compared to ≈42000,the mean of actual bitcoin price which signifies that having 3000 deviation while predicting a magnitude of around 42000 is nearly acceptable and hence all the models gave satisfactory results. However, the RMSE reported by XGBoost for all lags is around 1500 which indicates XGBoost performed well. In terms of MAPE for lag1, XGBoost is followed by LSTM then by SVR. However, for other lags SVR performed better compared to LSTM. Random Forest reported higher RMSE for all three lags.

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.

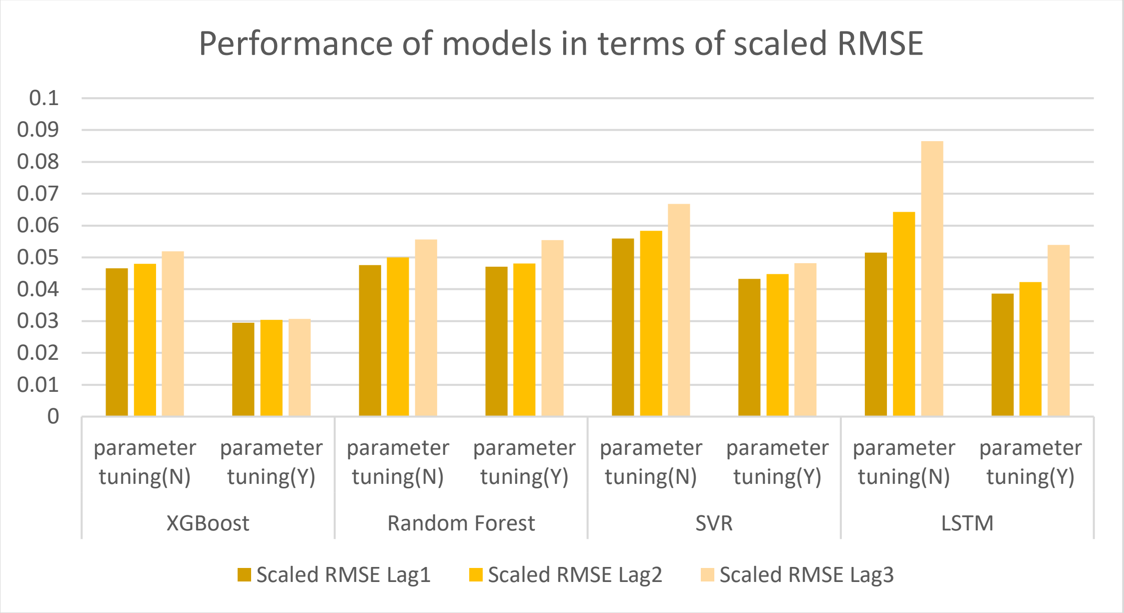


Figure 23: Performance of models in terms of scaled RMSE

Figure 23 shows the performance of models in terms of scaled RMSE. All the models performed well after parameter tuning reporting scaled RMSE varying from 0.29 to 0.55. Compared to all models, XGBoost reported the lowest scaled RMSE of 0.29 for lag 1. 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.

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.

9.3 Discussion:

The primary purpose of this quantitative research was to develop multivariate predictive models to forecast short term(the next day's) bitcoin prices based on other currencies' exchange rates by considering different lags in the input data. This chapter includes various discussions based on major findings of the performance of the models. At the end, this chapter discusses the limitations of study and concludes with a note on the best model.

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 10% . Any model resulting an MAPE of less than 10% is considered a good model for forecasting. XGBoost performed the best by reporting the better performance metrics for all lags. The MAPE reported by XGBoost varied from ranging from 2.6% to 4.7%. This gives the measure of the error loss in predictions by XGBoost. The RMSE and MAE reported by XGBoost after parameter tuning are way less (less than 1600 for all lags) compared to 42000, the mean of actual datapoints. These values determine the deviation of error from the actual datapoint. This again signifies that the algorithm did the best job in predicting the bitcoin prices. The scaled RMSE for XGBoost for all lags is around 0.03 which is close to 0 which again indicates that the model is good in fitting the datapoints for the forecast operation. 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 2.6%. Meaning, that it could predict around 98.4% of data accurately in the given dataset. This was followed by SVR and LSTM which could predict the prices almost equally with a marginal 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 which 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.

10 Conclusion and future scope:

This chapter discusses the conclusions derived from the results and discussion. 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 machine learning models which are based on different machine learning techniques to forecast bitcoin prices based on forex exchange rates. 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 bitcoin price determinants or independent features, and the next day's bitcoin price is considered as dependent features. The historical daily exchange rates and bitcoin prices for over more than five years (from 2017 to 2022) are collected from Bloomberg for this study, and 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 10% and gave satisfactory results for the lags1, 3 and 7. Though all the models performed better, XGBoost resulted in a minimal error percentage with MAPE around 2.6% for lag1 with a nearly 0.1% increase for lag3 and lag7.

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 favorable 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.

11 Appendix:

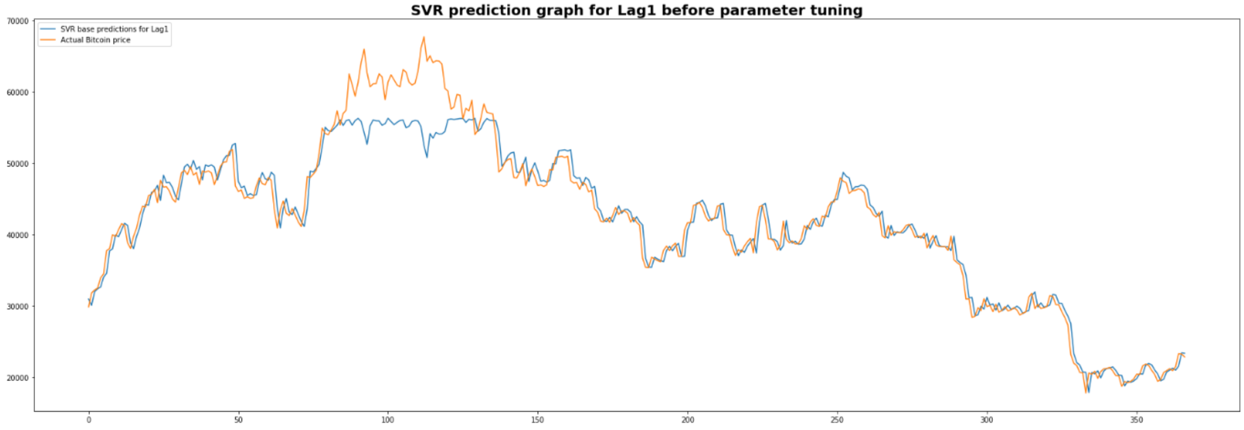
11.1 Code Repository:

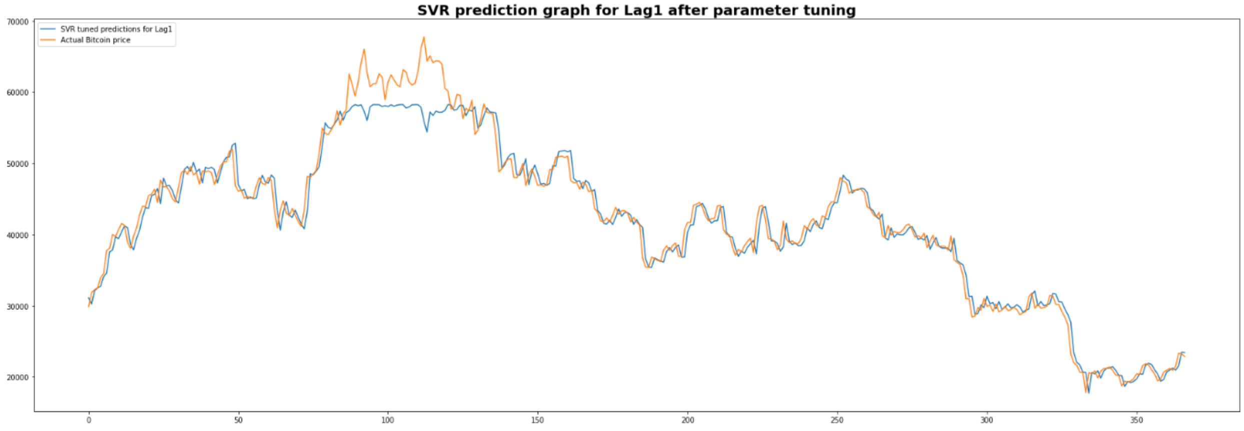
The code developed as part of this project can be found in below github repo.

https://github.com/RoopashreeRamachandraiah/Bitcoin-price-prediction

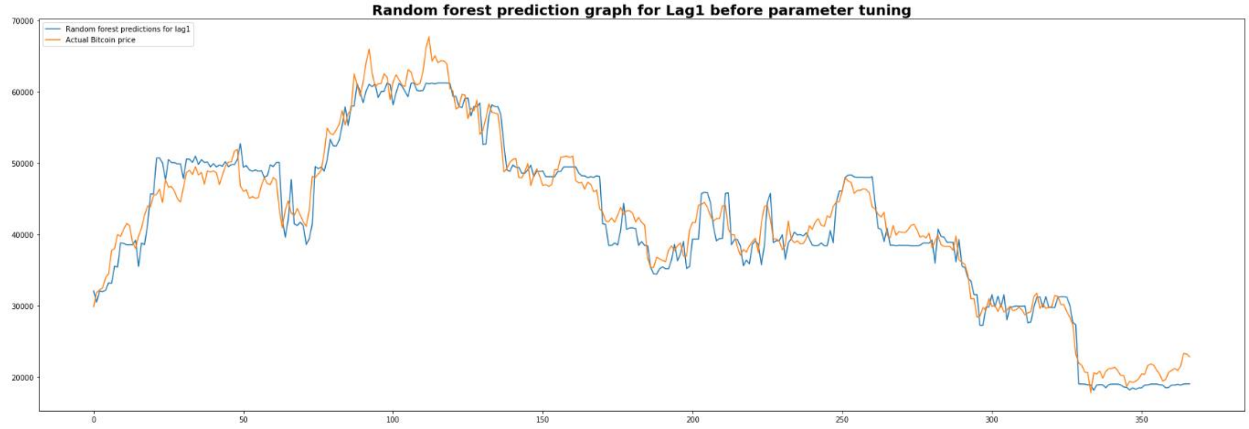
11.2 Performance graphs of other models for Lag1:

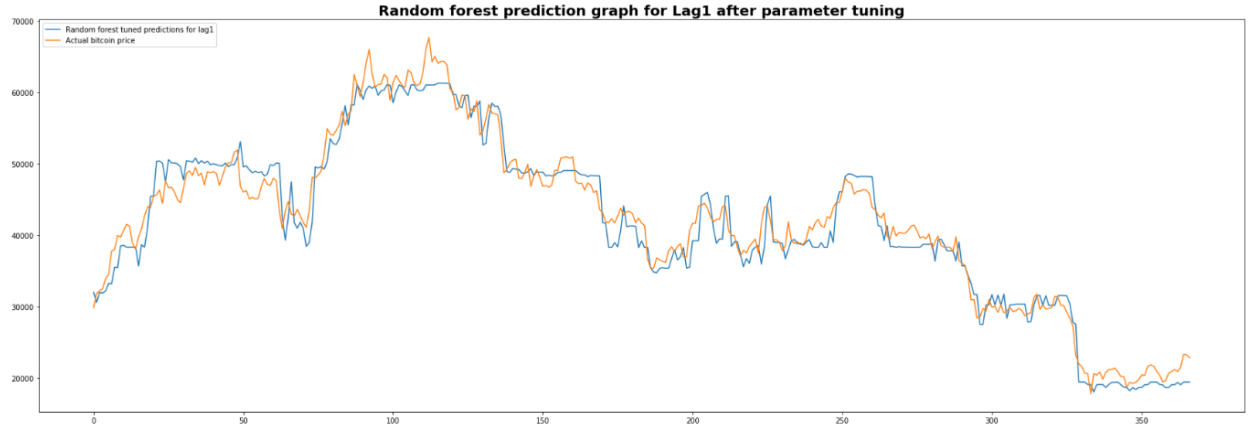
1) SVR performance for Lag1



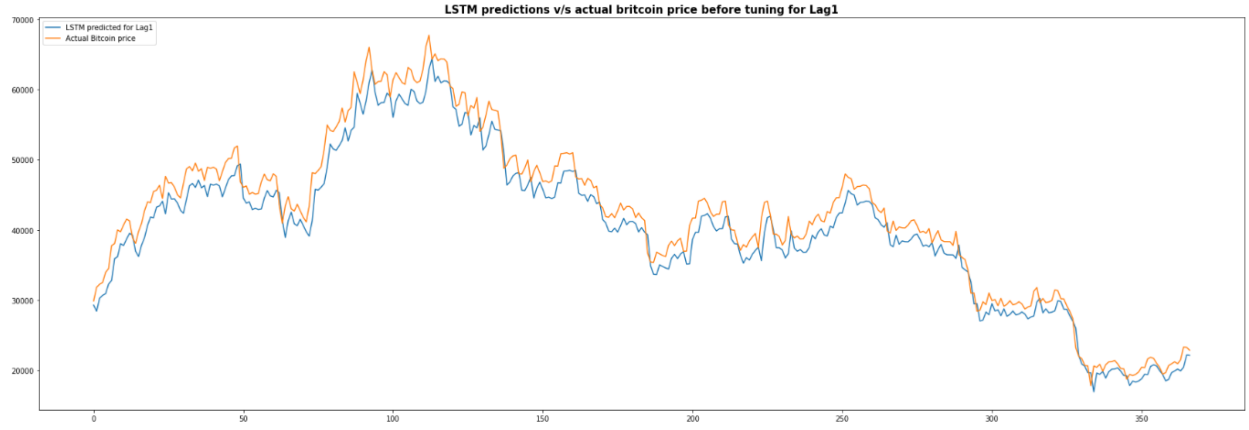


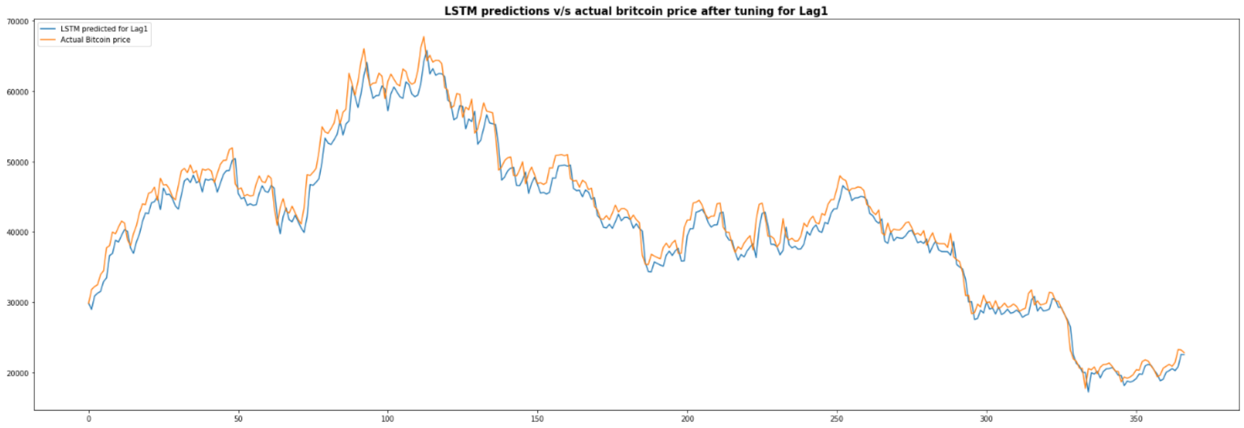
2) Random Forest performance for Lag1


Random forest prediction graph for Lag1 before parameter tuning


Random forest prediction graph for Lag1 after parameter tuning

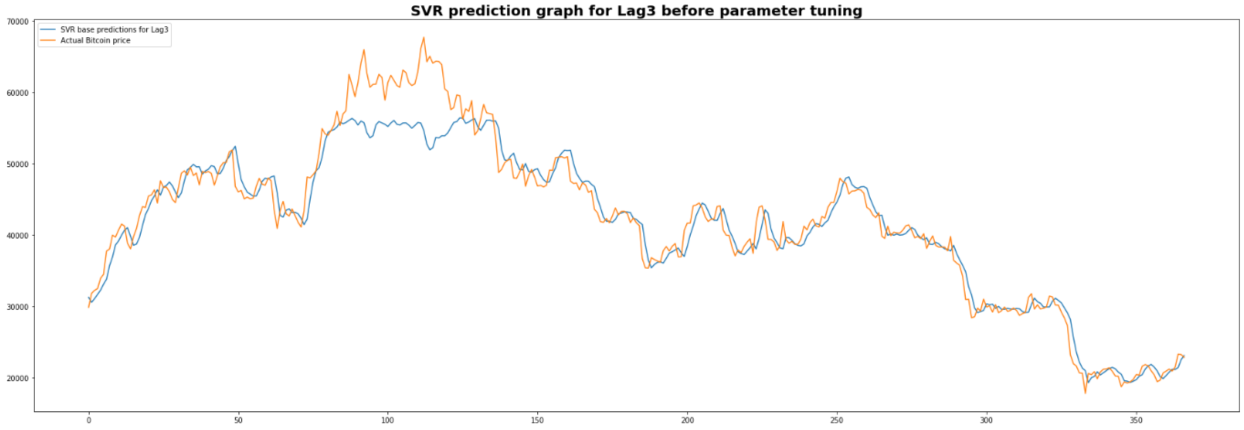
3) LSTM performance for Lag1

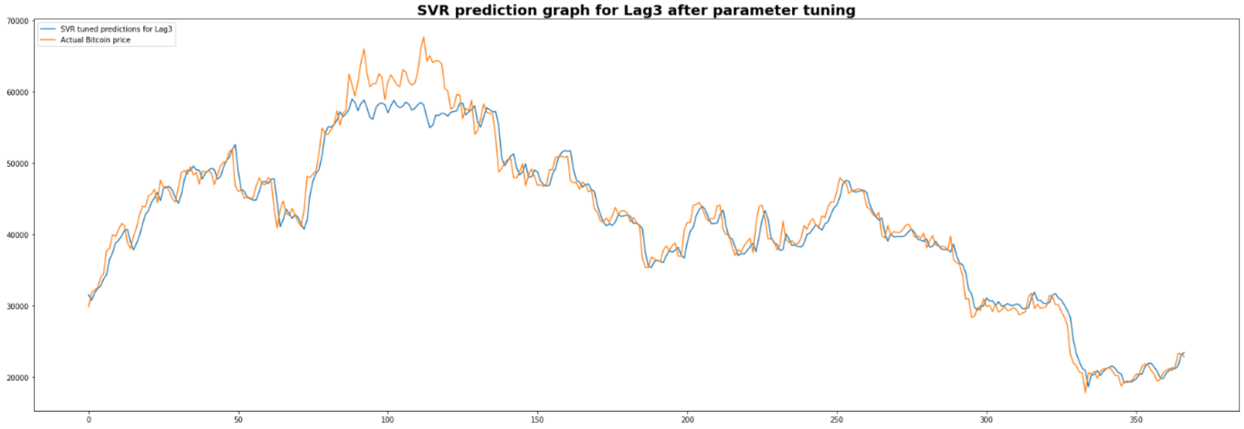

LSTM predictions v/s actual britcoin price before tuning for Lag1



11.3 Performance graphs of other models for Lag3:

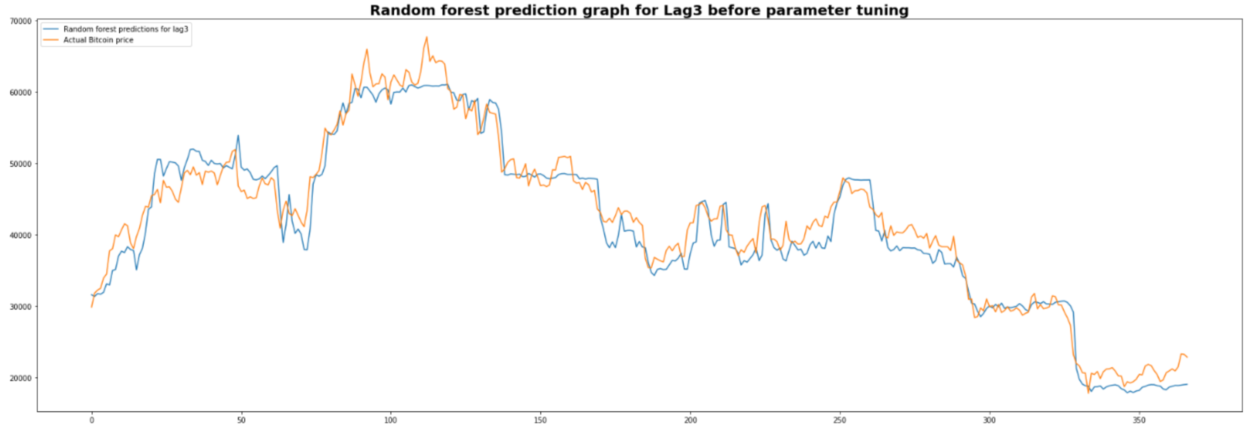
1) SVR performance for Lag3



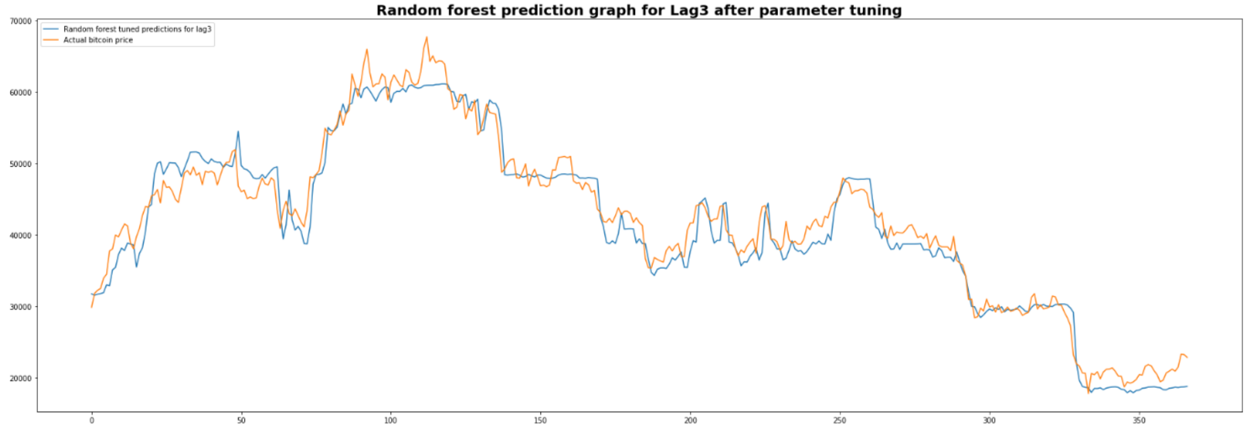


2) Random Forest performance for Lag 3

Random forest prediction graph for Lag3 before parameter tuning

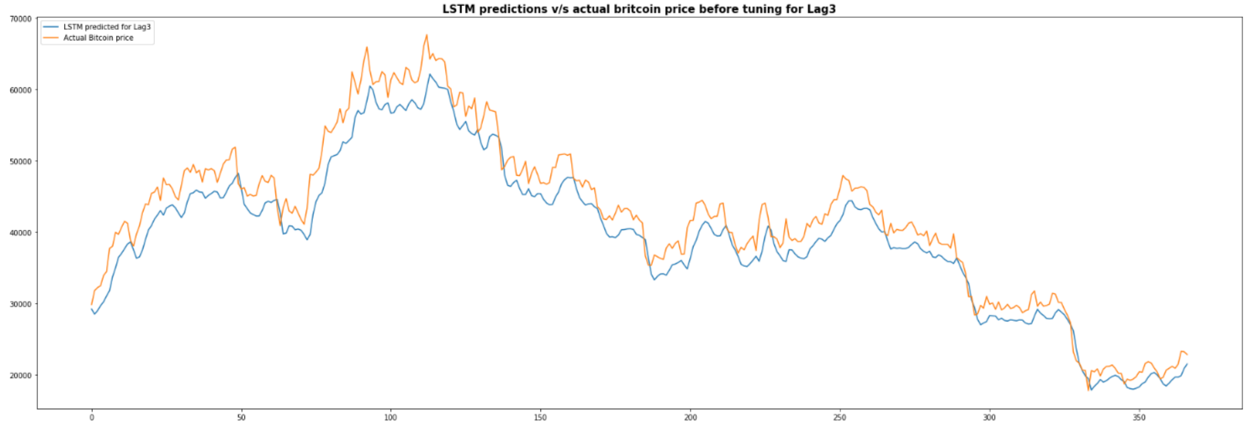


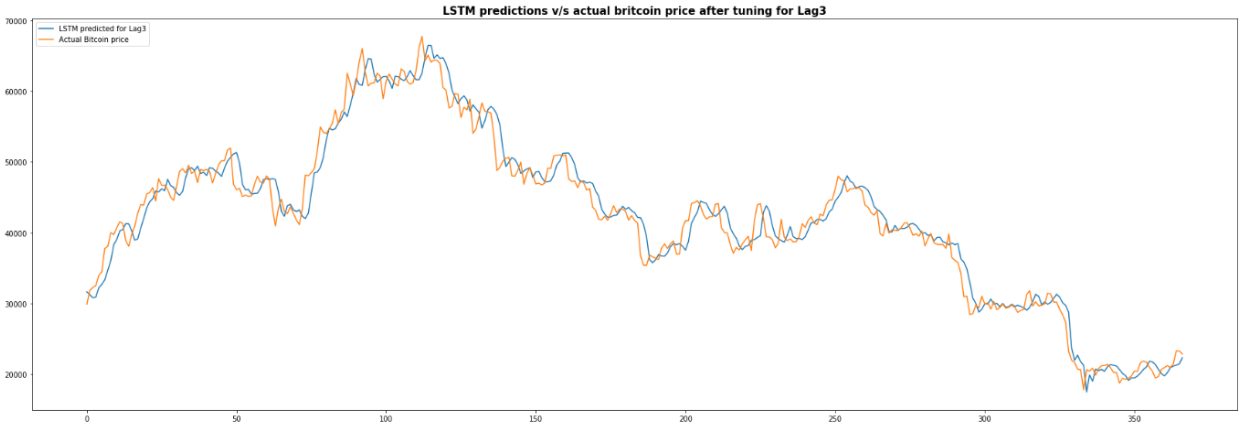
Random forest prediction graph for Lag3 after parameter tuning



3) LSTM performance for Lag3

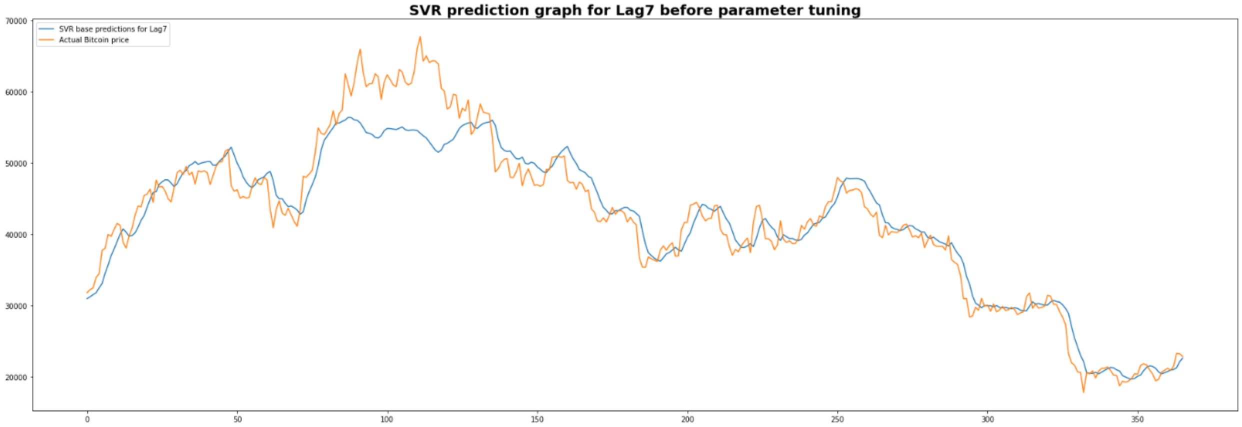
LSTM predictions v/s actual britcoin price before tuning for Lag3

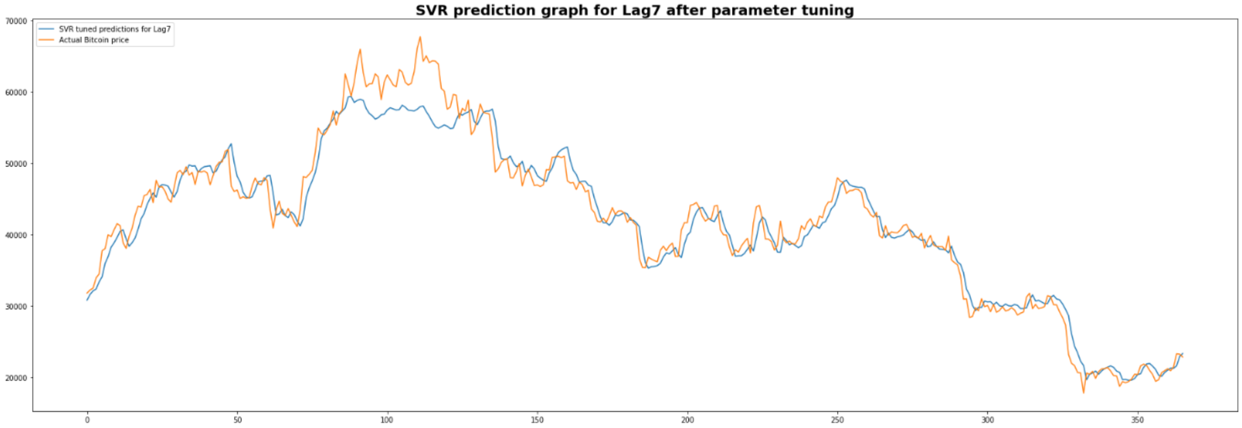




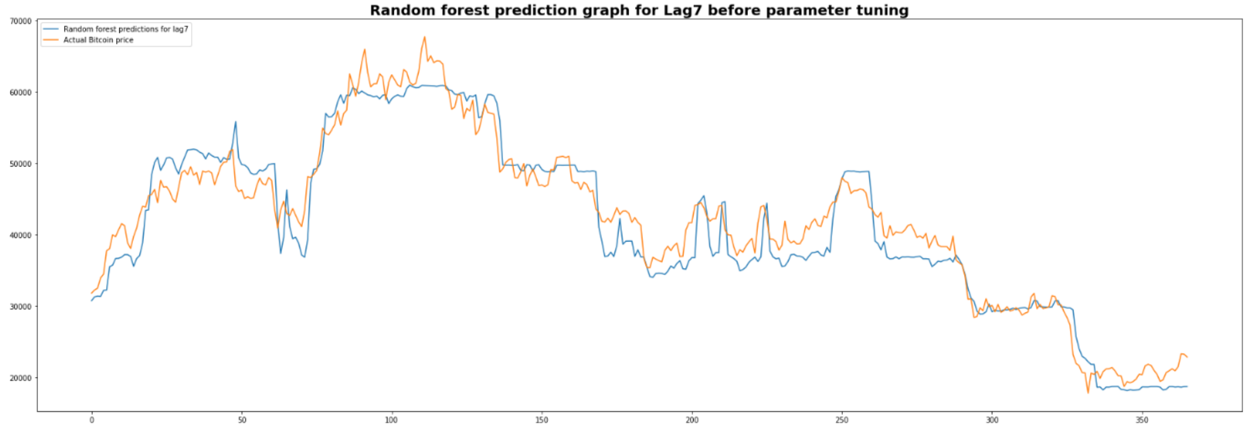
11.4 Performance graphs of other models for Lag7:

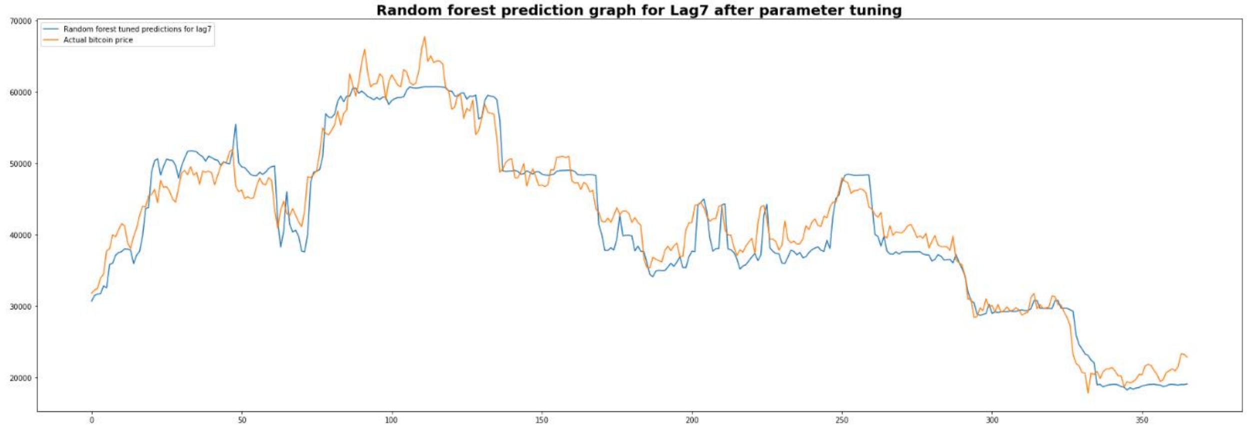
1) SVR performance for lag7



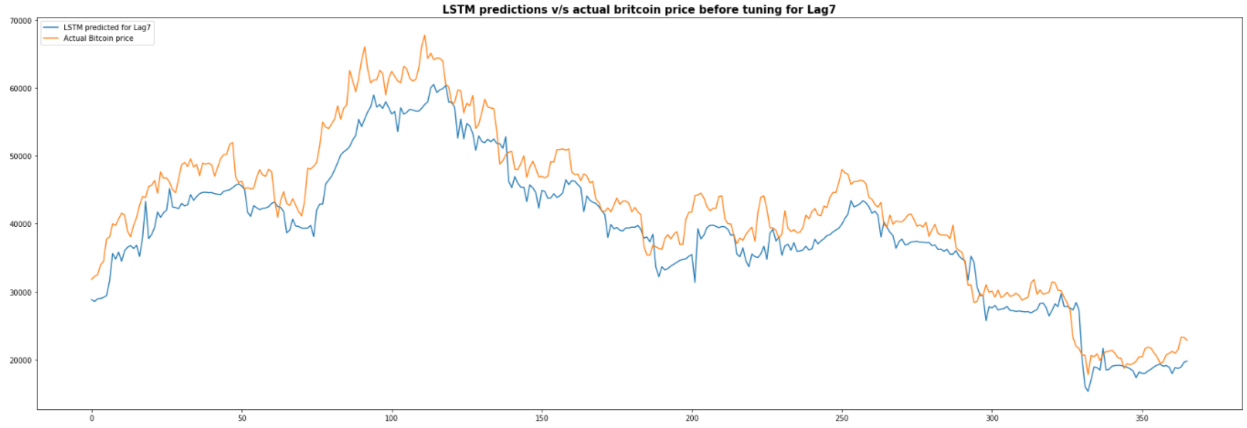


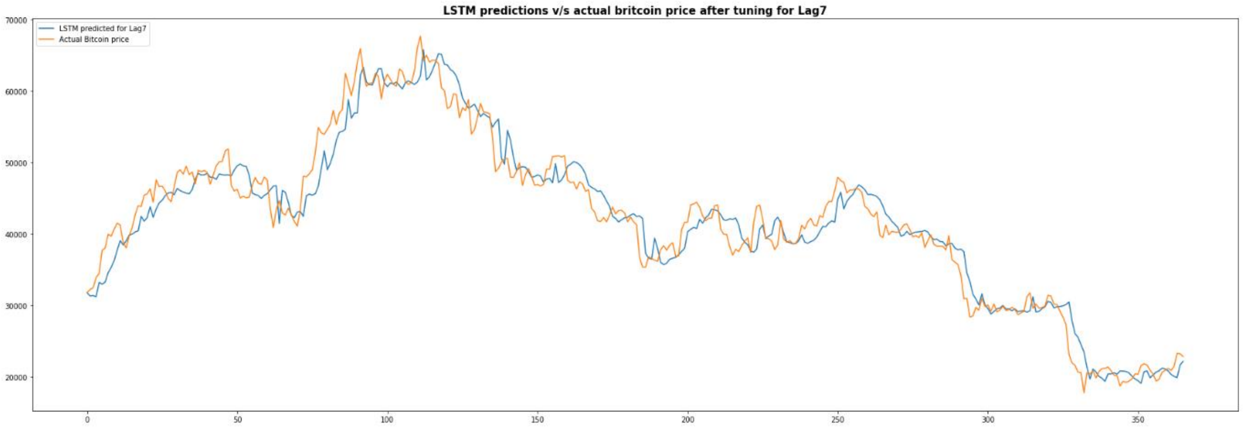
2) Random Forest performance for lag7





3) LSTM performance for lag7





12 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. Ladislav Kristoufek (2013). “BitCoin meets Google Trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era”. https://doi.org/10.1038/srep03415.

11. Michal Polasik, Anna Piotrowska, Tomasz Piotr Wisniewski, Radoslaw Kotkowski, Geoffrey Lightfoot. (2015). “Price Fluctuations and the Use of Bitcoin:An Empirical Inquiry”. https://www.ecb.europa.eu/pub/conferences/shared/pdf/retpaym_150604/polasik_paper.pdf
12. Theodore Panagiotidis, Thanasis Stengos, Orestis Vravosinos. (2018). “the effects of markets,uncertainty and search intensity on bitcoin returns”. http://rcea.org/RePEc/pdf/wp18-39.pdf.
13. Shabbir Dastgir, Ender Demir, Gareth Downing, Giray Gozgor, Chi Keung Marco Lau. (2019). “The Causal Relationship between Bitcoin Attention and Bitcoin Returns: Evidence from the Copula-based Granger Causality Test”. https://pure.hud.ac.uk/ws/portalfiles/portal/13340876/Bitcoin_Google_Trends_Submission_version.pdf.
14. Burcu Kapar, Jose Olmo. (2020) .” Analysis of Bitcoin prices using market and sentiment variables”. https://onlinelibrary.wiley.com/doi/epdf/10.1111/twec.13020?saml_referrer
15. Jacques Vella Critien, Albert Gatt and Joshua Ellul. (2022). “Bitcoin price change and trend prediction through twitter sentiment and data volume”. DOI:10.1186/s40854-022-00352-7
16. Syed Ali Raza,Maiyra Ahmed b and Chaker Aloui c. (2022). "On the asymmetrical connectedness between cryptocurrencies and foreign exchange markets: Evidence from the nonparametric quantile on quantile approach". https://www.sciencedirect.com/science/article/pii/S0275531922000150?via%3Dihub
17. Yechen Zhu, David Dickinson & Jianjun Li. (2017). “Analysis on the influence factors of Bitcoin’s price based on VEC model”. https://jfin-swufe.springeropen.com/articles/10.1186/s40854-017-0054-0.
18. Dirk Baur, Kihoon Hong, Adrian D. Lee. (2017). “Bitcoin: Medium of Exchange or Speculative Assets?”. DOI:10.1016/j.intfin.2017.12.004
19. Roman Matkovskyy,Akanksha Jalan, Michael Dowling. (2020). “Effects of economic policy uncertainty shocks on the interdependence between cryptocurrency and traditional financial markets”. https://doi.org/10.1016/j.qref.2020.02.004.
20. Zynobia Barson , Peterson Owusu Junior , Anokye M. Adam , and Emmanuel Asafo-Adjei, (2022). “Connectedness between Gold and Cryptocurrencies in COVID-19 Pandemic: A Frequency-Dependent Asymmetric and Causality Analysis” . https://doi.org/10.1155/2022/7648085.
21. Mustafa Ozyesil (2019). “A research on interaction between bitcoin and foreign exchange rates”. http://www.pressacademia.org/archives/jefa/v6/i1/5.pdf.
22. 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.
23. 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
24. 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

25. 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
26. 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
27. 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
28. 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
29. 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
30. 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
31. Monisha Mittal, G. Geetha. (2022). "Predicting Bitcoin Price using Machine Learning". https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9740772
32. 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
33. 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
34. Wes McKinney. (2022). https://pandas.pydata.org/docs/pandas.pdf
35. NumPy community. (2022). https://numpy.org/doc/stable/numpy-user.pdf
36. Michael L. Waskom. (2021). https://joss.theoj.org/papers/10.21105/joss.03021
37. https://scikit-learn.org/stable/
38. https://keras.io/about/
39. https://www.statsmodels.org/stable/index.html
40. Raynor de Best. (2022). "Bitcoin BTC/USD price history up until August 29, 2022". https://www.statista.com/statistics/326707/bitcoin-price-index/
41. Himani Gulati. (2021). "Time Series Analysis — Data Exploration and Visualization.". https://blog.jovian.ai/time-series-analysis-data-exploration-and-visualization-9dbede5cbb8d
42. Tianqi Chen and Carlos Guestrin. (2016). "XGBoost: A Scalable Tree Boosting System". https://arxiv.org/pdf/1603.02754.pdf

43. 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
44. https://link.springer.com/article/10.1007/s00521-020-05129-6#Tab2
45. 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
46. Christopher Olah. (2015). "Understanding LSTM Networks". https://colah.github.io/posts/2015-08-Understanding-LSTMs/
47. James Bergstra, Yoshua Bengio. (2012). “Random Search for Hyper-Parameter “. https://jmlr.csail.mit.edu/papers/volume13/bergstra12a/bergstra12a.pdf.