Short term Bitcoin Price Prediction and Analysis Using Machine Learning

Sanskar Jaiswal
Dept. of Material Science
IIT Bombay, India
19d110017@iitb.ac.in

Anish Satpati
Dept. of Material Science
IIT Bombay, India
190110007@iitb.ac.in

Khush Jain
Dept. of Mathematics
IIT Bombay, India
200010040@iitb.ac.in

Rishab Khantwal
Dept. of Mechanical Engineering
IIT Bombay,India
180100095@iitb.ac.in

***Abstract*—Bitcoin is a decentralized crypto-currency, which is a type of digital asset that provides the basis for peer-to-peer financial transactions based on blockchain technology. One of the major problems with decentralized cryptocurrencies is price volatility, which indicates the need for studying the underlying price model. Moreover, Bitcoin prices exhibit non-stationary behavior, where the statistical distribution of data changes over time. This paper demonstrates high-performance machine learning-based classification and regression models for predicting Bitcoin price movements and prices in short term. We found that ARIMA outperforms other machine learning models tested. Deep learning solutions like Long short Term Networks were also explored.**

KeyWords: Crypto-currencies, Deep learning, ARIMA, Forecasting,

I. Introduction

Bitcoin is a digital currency, introduced in 2008 by Nakamoto. It is enabled by the blockchain technology and allows for peer-to-peer transactions secured by cryptography. In this study, we analyze the short-term predictability of the bitcoin market. Therefore, we utilize a variety of machine learning methods and consider a comprehensive set of potential market-predictive features.

Empirical asset pricing is a major branch of financial research. Machine learning methods have been implemented increasingly within this domain, due to the ability to flexibly select amongst a potentially large number of features and to learn complex, high-dimensional relationships between features and target labels.

II. Literature Review

Due to the novelty of the Bitcoin technology, the research on predictive features for the bitcoin price is still in its very early stages; also, the findings of several researchers indicate that bitcoin might represent a new asset class. Most machine learning approaches demonstrate the ability to flexibly incorporate a large number of features. Together with the availability of large amounts of multidimensional data, this flexibility might render machine learning methods suitable for bitcoin pricing. This flexibility is especially important, since the stream of research on bitcoin pricing is still young and there exists limited guidance in the scientific literature about the nature of the bitcoin price formation process.

The authors of [1] demonstrated that incorporating cryptocurrency into a portfolio improved its effectiveness in two ways listed here; The first is by reducing standard deviation, and second is by providing investors with more allocation options. The best cryptocurrency allocation reported was in the range from 5% to 20% depending on the risk tolerance of the investor. The authors of [2] focused on time series data forecasting and applied two ML models, random forests and stochastic gradient boosting machine. Their results achieved showed that the ML ensemble technique could be used to predict Bitcoin rates. But the decision-making process needs to make the appropriate decision at the right time, thus reducing the risks associated with the investment process. In [3], a hybrid cryptocurrency prediction system based on LSTM and GRU is presented, focusing on two cryptocurrencies, Litecoin and Monero. The authors of [4] used Bitcoin returns which was minute-sampled over 3 h periods to aggregate RV data. A variety of ML methods, such as ANN (MLP, GRU, and LSTM), SVM, and ridge regression. These were used to predict future values based on past samples. Their findings show that the suggested model correctly predicts prices with a very high accuracy, indicating that this method may be used to forecast prices for a variety of cryptocurrencies. The authors of [5] employ the traditional support vector ma-

chine and linear regression methods to forecast Bitcoin values.

The authors of [6] used ML techniques to address both multiple regression that relies on highly correlated characteristics and a deep learning mechanism that uses a conjugate gradient mechanism in conjunction with a linear search for BTC price prediction. In [7], the price movements of Bitcoin, Ethereum, and Ripple are analyzed. The authors utilize powerful artificial intelligence frameworks, including a fully linked artificial neural network (ANN) and a long short-term memory (LSTM) recurrent neural network, and they discovered that ANN relies more on long-term history, whereas LSTM relies more on short term dynamics, implying that LSTM is more efficient at extracting meaningful information from historical memory than ANN.

The study in [8] on Bitcoin daily price prediction with high-dimensional data reveals that logistic regression and linear discriminant analysis achieve an accuracy of 66% On the other hand, surpassing (a sophisticated machine learning algorithm) outperforms the benchmark results for daily price prediction, with statistical techniques and machine learning algorithms having the greatest accuracies of 66% and 65.3% respectively. The study in [9] examines the use of neural networks (NN), support vector machines (SVM), and random forest (RF). The findings demonstrate that machine learning and sentiment analysis may be used to anticipate cryptocurrency markets (with Twitter data alone being able to predict specific coins) and that NN outperforms the other models.

In [10], the LSTM model is used to predict and find methods for forecasting Bitcoin on the stock market through Yahoo Finance that may predict a result of more than 12,600 USD in the days after the prediction. Due to the importance of the development of a robust and reliable method for predicting cryptocurrency prices, researchers have focused on more innovative models. In [11], both linear and non-linear time-series components of the stock dataset were used for forecasting using the hybrid model. In the non-linear time series forecast, CNN and Seq2Seq LSTMs were successfully coupled for dynamic modeling of short- and long-term dependent patterns.

III. Data and Methodology

The Dataset used is Bitcoin Historical Data. There are 8 columns as described below:

- Timestamp : Start time of time window (60s window), in Unix time
- Open : Open price at start time window
- High : High price within time window
- Low : Low price within time window
- Close : Close price at end of time window
- Volume_BTC : Volume of BTC transacted in this window
- Volume_currency : Volume of corresponding currency transacted in this window
- Weighted price : VWAP- Volume Weighted Average Price

Pre-processing and Exploratory Data analysis - The Bitcoin dataset was originally from 2012 to 2021 with frequency of 1 minute. For close data analysis, the year 2019 was considered. We converted frequency of the data changed to day as we averaged for the weighted price all over the day to obtain price corresponding to date. Maximum value of price was taken to obtain highest price of the day and Minimum value of price was taken to obtain lowest peak of the day. The close price of any day was considered the price at 23:59:00. We defined daily return ratio as difference between close

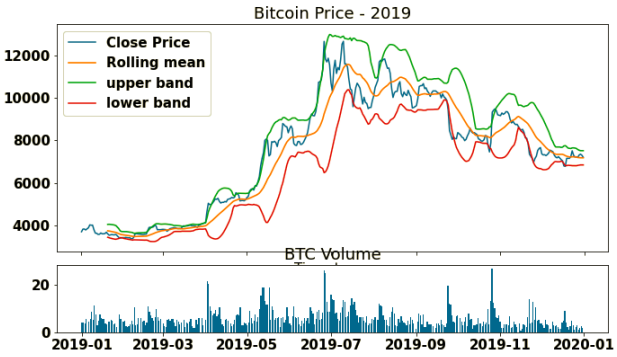

Fig. 1. Bollinger plot of Bitcoin for year 2019

price of the day and its previous date divided by close price of last day. Another feature introduced is volatility which is defined as the difference of Highest price and lowest price in the day divided by weighted price. We compared the volatality of Bitcoin against Etherium. Absolute mean of volatility of ETH in year 2019 is 0.06223 whereas absolute mean of volatility of BTC in year 2019 is 0.05256. We also performed the time series analysis to determine the correlation between prices of simulatneous days.

Following Machine Learning models have been implemented:

1) Support Vector Machine
2) Random Forests
3) ARIMA

4) LSTM(Long Short Term Memory) Networks

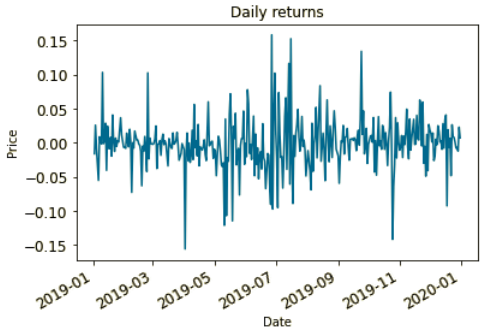

Fig. 2. Daily returns of Bitcoin for year 2019

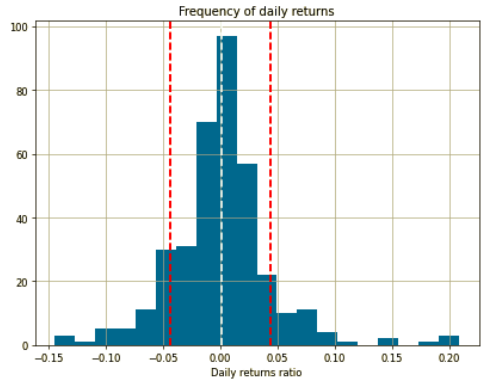

Fig. 3. Histogram for daily returns of Bitcoin for year 2019

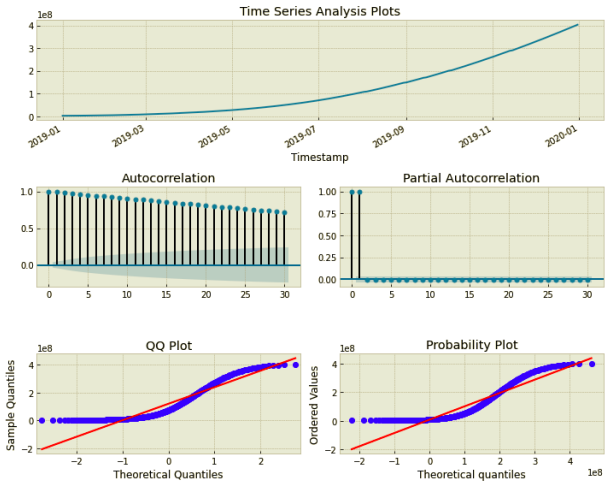

Fig. 4. Time Series analysis plot of Bitcoin for year 2019

A. Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

We implemented an SVR (Support Vector Regressor) model that takes data over the past `seq_len` days and predicts the output for the next day. A linear kernel is used for regression.

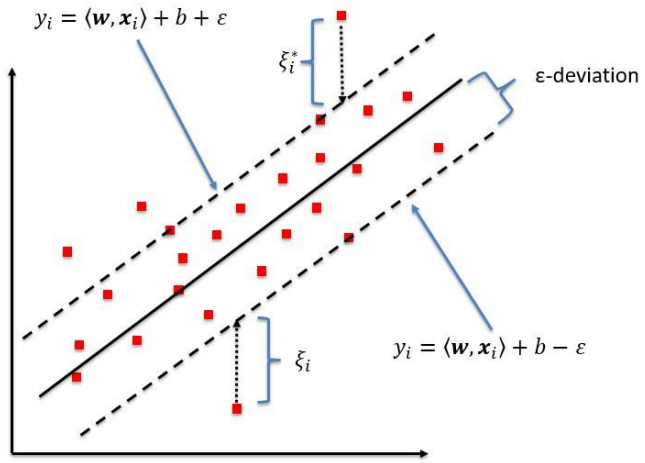

Fig. 5. Schematic of the one-dimensional support vector regression (SVR) model. Only the points outside of the 'tube' are used for making predictions.

As we can see a support vector regressor considers the data to be linear in the kernel space and then gives a robust regression model that minimises the regularised cost.

B. Random Forests

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees.

C. ARIMA

An auto regressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to better understand the data set and to predict future trends. A statistical model is auto regressive

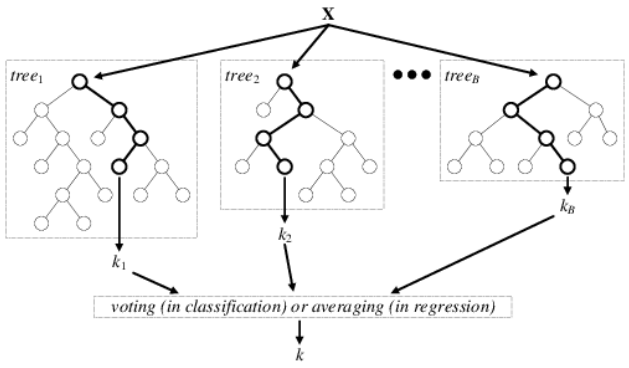

Fig. 6. Architecture of the random forest model

if it predicts future values based on past values. For example, an ARIMA model might seek to predict a stock's future prices based on its past performance or forecast a company's earnings based on past periods. An ARIMA model can be understood by outlining each of its components as follows:

- Autoregression (AR): This refers to a model that shows a changing variable that regresses on its own lagged, or prior values
- Integrated (I): This represents the differencing of raw observations to allow for the time series to become stationary (i.e., data values are replaced by the difference between the data values and the previous values)
- Moving average (MA): This represents the moving average, which is the dependency between an observed value and a residual error from a moving average model applied to previous observations

Each component in ARIMA functions as a parameter with a standard notation of p, d and q. Where p is the number of lag observations in the model, d is the degree of differencing and q is the size of the moving average window.

Generally, in an auto regressive integrated moving average model, the data are differenced in order to make it stationary. A model that shows stationarity is one that shows there is constancy to the data over time. Most economic and market data show trends, so the purpose of differencing is to remove any trends or seasonal structures. Seasonality, or when data show regular and predictable patterns that repeat over a calendar year, could negatively affect the regression model. If a trend appears and stationarity is not evident, many of the computations throughout the process cannot be made with great efficacy. Hence, we perform the Augmented Ducky Fuller Test, which is a statistical test called a unit root test. This can help us determine the extend of stationarity.

The null hypothesis (H0) of the test is that the time series can be represented by a unit root that is not stationary and the alternative hypothesis (H1) is that the time series is stationary. Here we define another most commonly metric that is the p value. In this work, we decided to accept the null hypothesis if p $>$ 0.05 and reject it if p $<=$ 0.05.

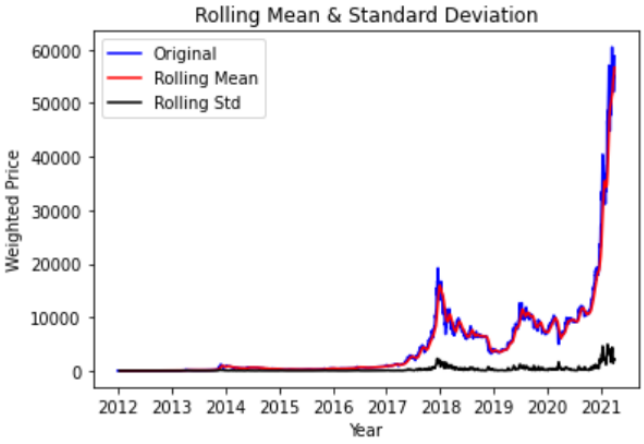

Fig. 7. Result of the ADF Test

From this test, the p value achieved was very close to 1, thus we concluded that the time series is non-stationary. Now, we must apply suitable transformations in order to make this time series stationary. The first method we try is taking the logarithm of the values. As log transformation is used to unskew highly skewed data, thus helping in the forecasting process. The plot achieved after applying the log transformation is shown in Fig.8. The p value decreased to 0.71 after applying the transformation which means we are moving in the right direction towards making our time series data stationary.

The p value for the data is still greater than 0.05, so we need to apply further transformations. We proceeded with differencing. In case of differencing, to make the time series stationary the current value is subtracted with the previous values. Due to this the mean is stabilized and hence the chances of stationarity of time series are increased. We subtract the logarithm of the weighted prices with the logarithm of the previous values. Then we again perform the ADF test in order to get the resulting p value. The p value achieved in this case was very close to 0, hence we succeeded in converting the non-stationary data into stationary data.

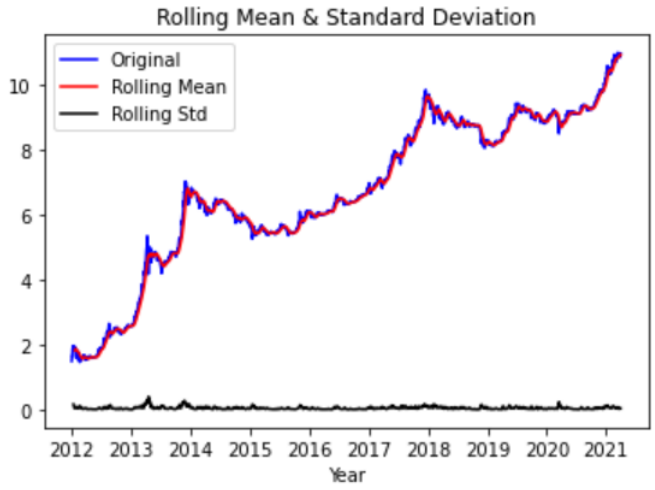

Fig. 8. After applying Log Transformation

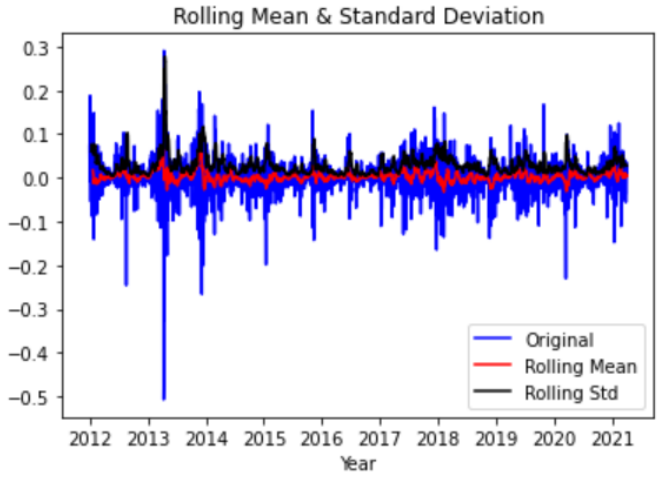

Fig. 9. After Differencing Operation

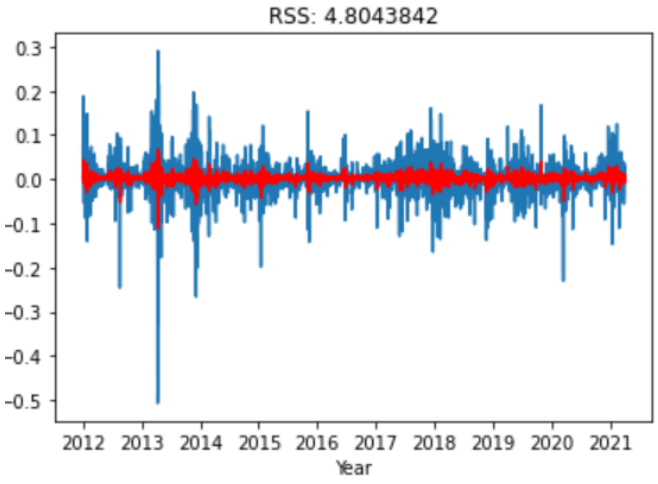

Fig. 10. RSS of Auto-Regressive Model

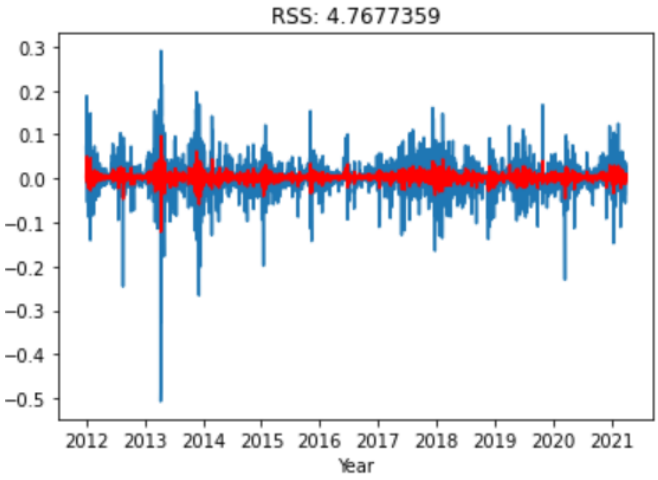

Fig. 11. RSS of Moving Average Model

As our time series is now stationary (p value is less than 0.05) we can apply time series forecasting models. We compare the performance of three models and then choose the model which has the minimum Residual Sum of Squares (RSS) as our final predictive model. First we start with applying a simple Auto regressive model which is a time series forecasting model where the current values are dependent on past values. Here, the (p,d,q) values are (1,1,0) respectively.

Next, we apply the Moving Average Model wherein the series is dependent on past error terms. In this case, the (p,d,q) values are (0,1,1) respectively.

The last model we try out is the Auto Regressive Integrated Moving Average (ARIMA) Model. It is a combination of both AR and MA models which makes the time series stationary by itself through the process of differencing. Therefore differencing need not be done explicitly for ARIMA model. From the three models, we observe that the RSS value is the least in case of the ARIMA model as can be seen from Fig.11.

Thus as the RSS (Residual Sum of Squares) error is minimum for ARIMA model, it is the best among the three models because of use of dependence on both lagged values and error terms. Therefore it is further used to calculate the mean square error.

We divided the dataset into training and testing datasets. The model was trained on data from 2011-12-31 to 2020-12-21 and it was tested on the last 100 days data, this is, from 2020-12-22 to 2021-03-31. For every value in the test test we apply an ARIMA model and then the error is calculated and then after iterating over all values in the test set the mean error between predicted

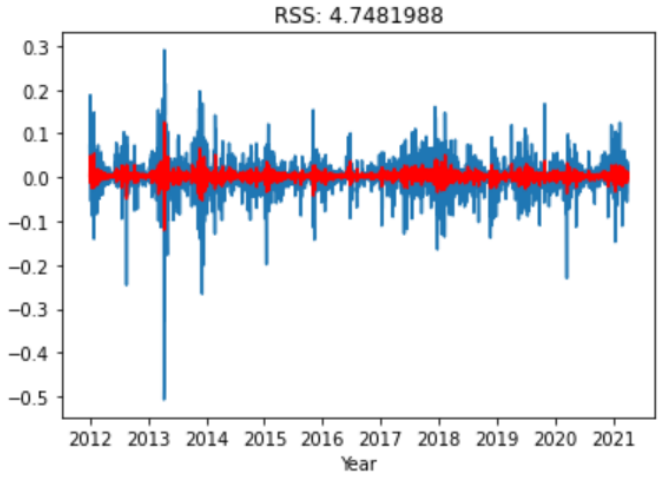

Fig. 12. RSS of ARIMA Model

and expected value is calculated.

D. LSTM

The long short term memory (LSTM) networks are special type of recurrent neural Networks having gated mechanism to solve the notorious problem of gradient vanishing and exploding.Deep learning has shown great success in Time Series forecasting Problems given their capability of dealing with huge amount of multivariate data. A limitation of RNNs becomes clear when we begin using large amounts of data. Since in our work we have considered over 9 years of Bitcoin prices data which boils to more than 3,500 timesteps of input and above all we also considered data input on a rolling basis, the dimensionality of the input even increased.Therefore, every time the model is updated, derivatives are calculated, which causes the weights to drop close to zero (vanishing gradient) or explode (exploding gradient) , meaning the model is learns very slowly. Because RNNs have difficulty learning early inputs on large datasets, often described as short-term memory.

Bidirectional LSTM (BiLSTM), and gated recurrent units (GRU) are some other type of RNNs using internal mechanisms, called gates, that can regulate how information flows through the network. Ultimately, they decide which information is important, and which is irrelevant

In our experiments, we have used Bidirectional LSTMs, which is just putting two independent RNNs together. This structure allows the networks to have both backward and forward information about the sequence at each step. Using bidirectional will run the inputs in two ways- one from past to future and one from future to past.Where, In the LSTM that runs backward we could gather information from the future and using the two hidden states combined we are able to preserve information from both past and future

IV. Results and Analysis

A. Support Vector Machine

Our model gives an R2 score of above 0.995 on test data. Though this looks like a really good prediction score, it turns out to be not so useful as evident from the following graphs.

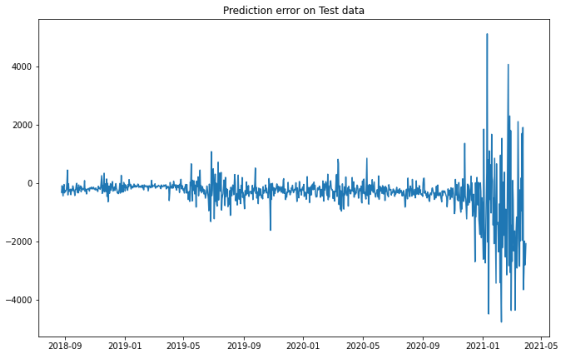

Fig. 13. Prediction error on Test data

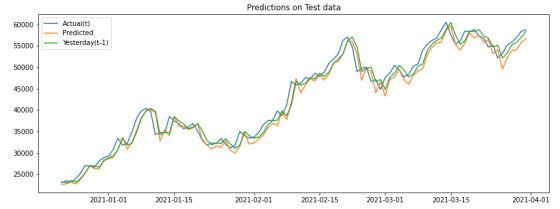

Fig. 14. Prediction on Test data

As we can see that our prediction closely follows the prediction on the day before, and therefore our model does not have a very good predicting power.

B. Random Forests

A grid-search over hyperparameters gave a model with R2 score of 0.999 on training set. But the model does not perform well on the test dataset.

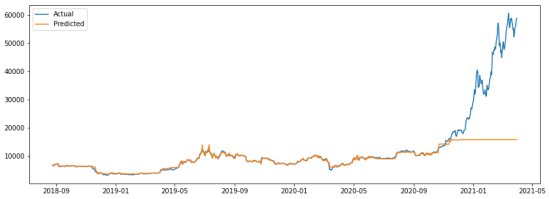

Fig. 15. Test predictions

As seen from the graph, the Random forest model fails to generalize over the test dataset. This is because the range of values of Weighted price changes when the price soars high and there is not enough representation of high prices in our training set.

C. ARIMA

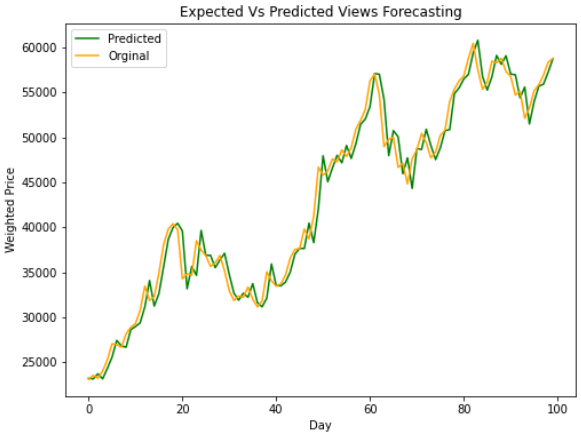

Fig. 16. Prediction of ARIMA Model

The mean absolute error (MAE) in predicting the test case data achieved was 3.32% Fig.17 shows the original and predicted time series. R2 score achieved was 0.97 and the RMSE value achieved was roughly 1780.69. The RMSE is quite high which can be expected as the data is quite volatile and there are sudden jumps in some time intervals. But on the overall, we were able to use different transformations and models to predict the weighted price of bitcoin with a decent level of accuracy.

D. Experiments using LSTM

Our LSTM architecture consists of a 2-layer deep version of Bidirectional LSTM. The output units of each layer is 100 where the input of 30 timesteps by 5 features matrix was fed into this model.The first BiLSTM layer is then followed by a dense layer acting as a hidden connection with tanh activation function, which gives a value between -1 or 1 and this is followed by another BiLSTM layer. Next, a Dropout factor was added as a regularizer for our architecture to prevent overfitting.

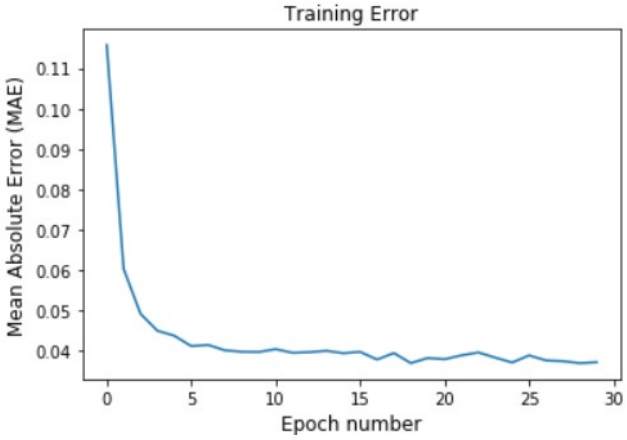

Fig. 17. Mean Squared Error Plot vs Number of Epochs

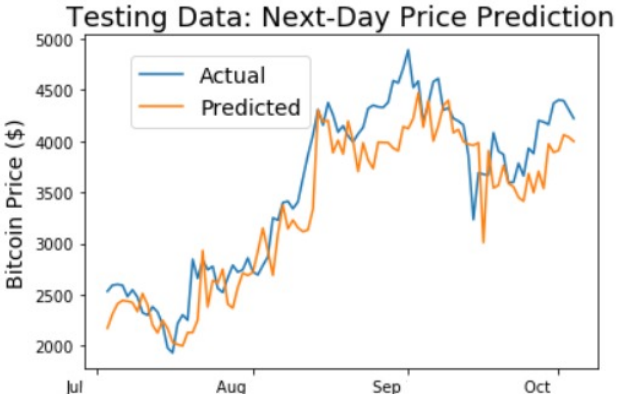

Fig. 18. Next Day Prediction for the year 2020

For our model, we narrowed down the features since some of them like the Volume (Currency) or number of bitcoins in circulation simply increase with time regardless of changes in price. Further considerations relied on a sliding window of memory of an optimal length (we found 30 days to be optimal. There lies a large degree of variance in the accuracy of our model to predict future price trends based on the size of the sliding window of memory, number of neurons, epochs used to train. Optimally, we found that using a window of memory of 30 days, 20 neurons and training for 40-50 epochs led to the best results. However, there is still plenty of room for optimization that can produce

even better performance.

Although the Mean Absolute Error for training decreased sharply, the testing error initially decreased but remain constant after some epochs.This concluded sign of overfitting. Lot of Experimentation was done to tune the hyperparameters such as size of window,horizon and fixing number of epochs and batchsize, but the results didn't improved quite a lot. One of the potential reason could be the lack of high frequency data and less number of features.

V. Conclusion & Further Work

As directly conclusive from our Exploratory data analysis that bitcoin prices are highly volatile which we also know from domain knowledge as bitcoin price highly depends on market news such as tweets or policies from government authorities with regards to cryptocurrency and sentiments of successful investors. Although from our autocorelation plot we know that there is some dependence of day's price on price of previous days, the model developed to predict the prices still were not able to optimally learn the data distribution.

The SVR model achieved a high R2 score of about 0.995 on both test and training set. It also achieved a low MAE(Mean absolute error) score of 0.0367 and MSE(Mean Squared Error) score of 0.0021. The Random Forests model performed better on training set but failed to generalise over the test set. This makes Random Forests model not helpful when the test set is not in the range of values of the train set. The ARIMA model was able to achieve a good mean absolute error value and a good correlation between the actual and predicted values in the test dataset but the RMSE value was quite high even though the R2 score was quite good.

The Lstm model rapidly overfitted the data, where one of the possible way to solve this problem is to have high frequency data with less granularity. Along with this, having more related features like blockchain related predictors which act as technical indicators.For Further improvement in results, We can add additional features which can respond to market news impact on price such as positive and negative and could use it to identify sudden increase or decrease.

VI. Contribution

Sanskar- Researched about Multivariate Sequential Modelling and Experimented with different RNN based models such as LSTM,GRU models, Implemented hyperparameter tuning and Ablation study of different features and designing custom features.

Anish- Analysed and studied the data using traditional statistical Methods like ARIMA and its variants, performed data transformations and differencing and experimented with several evaluation metrics such as RSS, Rolling Mean and Rolling standard deviation.

Khush- Implemented Traditional Machine Learning Approaches such as Random Forest(Ensemble Models), Support Vector Regression(SVMR) and did hyperparameter training to reduce variance in the results.

Rishab- Data cleaning, Data pre-processing, Feature selection and Exploratory data analysis of Bitcoin price over years. Autocorrelation and partial coorelation plots. Analysing new features like volatility and daily returns. Comparing Bitcoin data with Etherium data.

References

[1] Andrianto, Y. The Effect of Cryptocurrency on Investment Portfolio Effectiveness. J. Financ. Account. 2017, 5, 229.

[2] Derbentsev, V.; Babenko, V.; Khrustalev, K.; Obruch, H.; Khrustalova, S. Comparative Performance of Machine Learning Ensemble Algorithms for Forecasting Cryptocurrency Prices. Int. J. Eng. Trans. A Basics 2021, 34, 140–148.

[3] Patel, M.M.; Tanwar, S.; Gupta, R.; Kumar, N. A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. J. Inf. Secur. Appl. 2020, 55, 102583.

[4] Miura, R.; Pichl, L.; Kaizoji, T. Artificial Neural Networks for Realized Volatility Prediction in Cryptocurrency Time Series. In Advances in Neural Networks—ISNN 2019; Lu, H., Tang, H.,Wang, Z., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11554.

[5] Karasu, S.; Altan, A.; Sarac, Z.; Hacioglu, R. Prediction of Bitcoin prices with machine learning methods using time series data. In Proceedings of the 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018

[6] Saad, M.; Mohaisen, A. Towards characterizing blockchain-based cryptocurrencies for highly-accurate predictions. In Proceedings of the IEEE INFOCOM—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018.

[7] Yiying, W.; Yeze, Z. Cryptocurrency Price Analysis with Artificial Intelligence. In Proceedings of the 5th International Conference on Information Management (ICIM), Cambridge, UK, 24–27 March 2019; pp. 97–101.

[8] Chen, Z.; Li, C.; Sun, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. J. Comput. Appl. Math. 2019, 365, 112395.

[9] Valencia, F.; Gómez-Espinosa, A.; Valdés-Aguirre, B. Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. Entropy 2019, 21, 589.

[10] Ferdiansyah, F.; Othman, S.H.; Radzi, R.Z.R.M.; Stiawan, D.; Sazaki, Y.; Ependi, U. A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market. In Proceedings of the ICECOS—3rd International Conference on

Electrical Engineering and Computer Science, Batam, Indonesia, 2–3 October 2019; pp. 206–210.
[11] Zhao, Y.; Chen, Z. Forecasting stock price movement: New evidence from a novel hybrid deep learning model. J. Asian Bus. Econ. Studies 2021. ahead-of-print.
[12] Zhao, Y.; Chen, Z. Forecasting stock price movement: New evidence from a novel hybrid deep learning model. J. Asian Bus. Econ. Studies 2021. ahead-of-print.