Bitcoin Price Forecasting: A Comparative Study of Machine Learning, Statistical and Deep Learning Models

Neelam Urooj[1,*], Laiba Asif [2], Zohra Jabin[1]
[1] Institute of Business Management and Administrative Sciences (IBM & AS), The Islamia University of Bahawalpur, Bahawalpur, Pakistan
[2] School of Systems and Technology, University of the Management and Technology, Lahore, Pakistan
*Correspondence: ellahineelam786@gmail.com
Citation | Urooj. N, Asif. L, Jabin. Z, "Bitcoin Price Forecasting: A Comparative Study of Machine Learning, Statistical and Deep Learning Models", IJIST, Vol. 6 Issue. 2 pp 396-412, April 2024
Received | Mar 25, 2024 **Revised |** April 20, 2024 **Accepted |** April 22, 2024 **Published |** April 25, 2024.

Introduction/Importance of Study:
Cryptocurrency price prediction is crucial for investors and researchers, given the market's nonlinear nature and the potential for significant financial implications.
Novelty:
This study offers a novel approach to cryptocurrency price prediction, leveraging a range of machine learning and deep learning models to address the challenges of predicting Bitcoin's exchange rate.
Materials & Methods:
The study employs various machine learning and deep learning models, including Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), along with traditional models like Linear Regressor, Random Regressor, ExtraTreesClassifier, XGBoost Regressor, ARIMA, Prophet, and CNN.
Results & Discussion:
The ExtraTreesClassifier model emerged as the top performer, achieving a Test MAPE of 0.0689. This model outperformed deep learning models like RNNs, indicating its effectiveness in cryptocurrency price prediction.
Conclusion:
The findings suggest that the proposed models, particularly the ExtraTreesClassifier, can provide valuable insights for investors and traders in the cryptocurrency market.
Keywords: Bitcoin, Forecasting, Statistical Analysis, Machine Learning, Deep Learning

1. Introduction

Bitcoin, a decentralized digital currency created in 2008 under the pseudonym of Satoshi Nakamoto, operates independently of financial institutions and governments, utilizing encryption for security [1]. Its decentralized nature enables rapid transfer of money around the world without reliance on central banks. It has grown in acceptance as a medium of trade and a store of value [2]. Despite experiencing fluctuations over the previous ten years, in November 2021 it exceeded USD 68,000 per coin, and at one point the total current price surpassed USD 1.2 trillion. However, bitcoin is the world's largest published computing project [3]; that was initially acknowledged by some segments of society, is currently recognized by most nations, competing with other international currencies [4].

However, one issue with Bitcoin as a commodity is its extreme volatility. The standard deviation of Bitcoin's daily return rate over the seven years from April 2015 to April 2022 was 3.85%, which was 2.68 times greater than the standard deviation of gold's return rate over the same period and 3.36 times greater than the S&P 500.This volatility raises concerns about Bitcoin's suitability as a stable currency for transactions and a reliable store of value.

In recent studies, Artificial Intelligence (AI) algorithms and strong computer processing capacity were used to forecast the price of Bitcoin in the future. The advancements in hardware performance in 21st century has propelled machine-learning technology into various domains, including financial markets. Machine learning has been applied in many fields, including the stock market [5]. Time series and machine learning are the two approaches used in Bitcoin price prediction research. Numerous studies [6], and [7] have found that machine learning outperforms ARIMA in terms of prediction accuracy.

Aggarwal et al. [8] included the price of gold as one of the explanatory factors along the bitcoin price. In comparison to CNN and GRU, the LSTM approach has a reduced Root Mean Square Error (RMSE) of 47.91 based on the testing data. McNally et al. [6] expanded their investigation by including the variable's difficulty and the hash rate of Bitcoin characteristics; with the prediction accuracy of 52.78%, LSTM performs better than that of RNN and ARIMA.

While macroeconomic factors were not included in a similar experiment, the self-adaptive LSTM approach in the study still showed promising prediction performance. García-Medina and Duc Huynh [9] conducted a creative analysis of variables, such as social media and the Tesla shares cost, to determine their impact on the price of Bitcoin. Despite initial attention, these factors were found to be insufficient in explaining the price fluctuations observed in the latter half of 2020. Akyildirim et al. [10] used machine-learning classification algorithms, such as Support Vector Machine (SVM), logistic regression models, artificial neural networks, and random forest for predicting cryptocurrency returns.

They found that while the average classification accuracy across these methods was close to 50%, SVM consistently produced better and more reliable results compared to other classification algorithms like logistic regression and artificial neural networks.

Objectives:

- Examining, with an emphasis on the exchange rate of Bitcoin, various machine learning and deep learning models for predicting cryptocurrency prices.
- Evaluation of the degree to which these models can accurately represent the intricate and rapidly changing bitcoin markets.
- Using the model's performance and projections as a basis, offer traders and investors recommendations.
- Contribute to developing the field of study on machine learning applications in finance and cryptocurrency price prediction.

Novelty Statement:

Using a wide range of machine learning and deep learning models, including neural network models like LSTM and GRU, as well as conventional models like Linear Regressor, Random Regressor, and XGBoost Regressor, ARIMA and many more models, this study offers a novel approach to cryptocurrency price prediction. We aim to shed light on the dynamics of cryptocurrency markets and providing insights into the market by contrasting and assessing these models.

Methodology

Machine learning is a significant field of study closely related to artificial intelligence, categorized into three main branches: reinforcement learning, unsupervised learning, or supervised learning, contingent upon the presence of a goal variable. This research aims to use a regression function in a supervised learning framework to forecast future Bitcoin values. This section presents the description of the dataset as well as machine learning models and Deep Learning models that are considered in this study.

Dataset:

The daily data spanning from November 28, 2014, to January 1, 2022, was extracted from the open-access Kaggle [11] and used for the experiments. The targeted variable of this study is price of Bitcoin in US dollars. The response variable is the BTC closing rate, while the date is considered the explanatory variable. The descriptive statistics for the BTC closing rate is shown in Table 1.

Table 1: Statistical Measures of the Bitcoin Dataset

Statistics	Close BTC
Count	2651.0000
Mean	11709.3262
STD	16282.9087
Min	162.0000
25%	654.3700
50%	6407.7700
75%	10726.4250
Max	67559.0000

The history of the Bitcoin (BTC) close rate is depicted in Figure 1, exhibiting the changes in the BTC closing price from November 28, 2014, to January 1, 2022. It is important to examine this pattern for the following reasons.

- **Finding Patterns:** Analyzing the trend helps identify patterns such as increasing or declining trends, cyclical oscillations, and anomalies. These patterns can provide insights into how the market functions and support well-informed decision-making.
- **Predicting Future pricing:** Historical data serves as a common method for predicting future pricing. By studying the past trends, analysts may develop models that predict future prices of Bitcoin, which can be useful for investors and traders.
- **Comprehending Market Behavior:** Moreover, the trend can be utilized to gain a deeper understanding of market dynamics and the factors influencing Bitcoin prices. Sudden changes in the closing rate may be attributed to news events, regulatory changes, or shifts in market sentiment.
- **Risk management:** Awareness of this trend is valuable for risk assessment. Traders and investors can use historical data to assess the risk associated with Bitcoin investments and develop strategies to lower such risks.

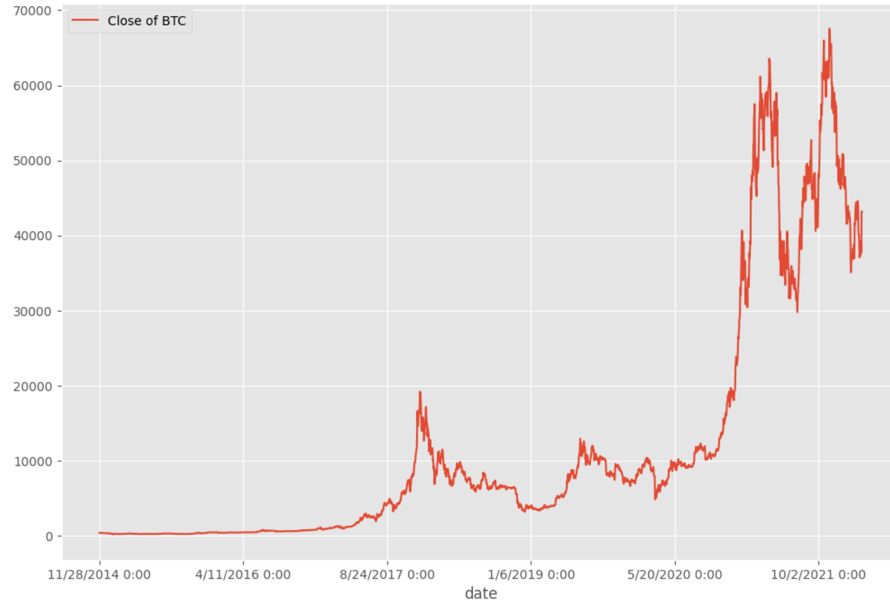

Figure 1: BTC Closing Value Trend from November 28, 2014, to January 1, 2022

In terms of preprocessing, dataset did not contain any null values, simplifying preprocessing becomes easier. The absence of null values suggested that the dataset is free of missing values, if handled improperly, missing data might result in biased analysis and incorrect predictions.

Machine Learning Models:

This study explored various machine-learning models that can serve as regressors [12]. These models, which are frequently employed for regression tasks, consist of:

Linear Regressor:

The statistical method of linear regression was used to model one independent variable (x) and one dependent variable (y), typically expressed as y [13]. It is predicated that the independent and dependent variable having a linear relationship, as shown by the equation:

$$y = \alpha + \beta x \quad (1)$$

The study's dependent variable, which is the BTC closing rate, is denoted by y, while the independent variable, or characteristics, is represented by x. The coefficients that need to be estimated are β and the error term α.

Random Forest Regressor

A random forest is an ensemble technique composed of multiple decision trees, each trained on a different subset of the data, known for its high explainability and predictive power[14]. However, the training samples of a random forest model could have an impact on the anticipated outcomes. The fundamental component of a random forest, a regression tree, divides the data recursively into smaller groups according to the values of a particular variable, to minimize the sum of squared residuals within each group. This method continues until a predetermined endpoint is met, either a minimum node size or a maximum tree depth.

Regression trees usually use a splitting criterion that is focused on reducing the Sum of Squared Residuals (SSR) within each group. For any given node m and dataset Dm comprising Nm observations, the SSR can be computed as follows: $SSR = \sum_{i=D_m}(y_i - \hat{y}_m)$ (2)

A splitting criterion, such as the information gain (for entropy-based criteria), the Gini impurity (for classification trees), or the decrease in SSR (or rise in R-squared), controls the splitting process.

XGB Regressor

XGBoost Regressor [15] is a powerful implementation of gradient boosting machine learning algorithm designed for speed and high-performing. It is an acronym for eXtreme Gradient Boosting.

- **Gradient Boosting:** XGBoost follows the concept of gradient boosting which builds an ensemble of weak learners (usually decision trees) one at a time. Each new tree corrects errors made by the previous ones.
- **Objective Function:** There are specific objective functions used in XGBoost that need to be optimized during training. For regression tasks, the objective function typically is the MAPE and its variation.
- **Regularization:** XGBoost has regularization terms in its objective function for model complexity control and preventing overfitting. Regularization parameters include gamma, alpha, and lambda.
- **Tree Construction:** In this way, XGBoost constructs trees depth-wise rather than breadth-wise growing the tree level by level by adding nodes that minimize the objective function in the best manner and hence can lead to more overfitting compared to other methods such as lightGBM which uses leaf-wise growth strategy.
- **Handling Missing Values:** That means XGBoost can handle missing values internally without having to impute them.
- **Parallel Processing:** XGBoost works well with big datasets since it is designed for parallel processing.
- **Cross-validation** is supported by XGBoost and is useful for adjusting hyperparameters and assessing model performance.

In this research paper, we have selected the XGBoost Regressor due to its speed, performance, and ability to handle complex datasets. A common practice of splitting the dataset into 80% for training and 20% for testing is adopted for all machine learning models, as depicted in Figure 2 for the Linear Regressor.

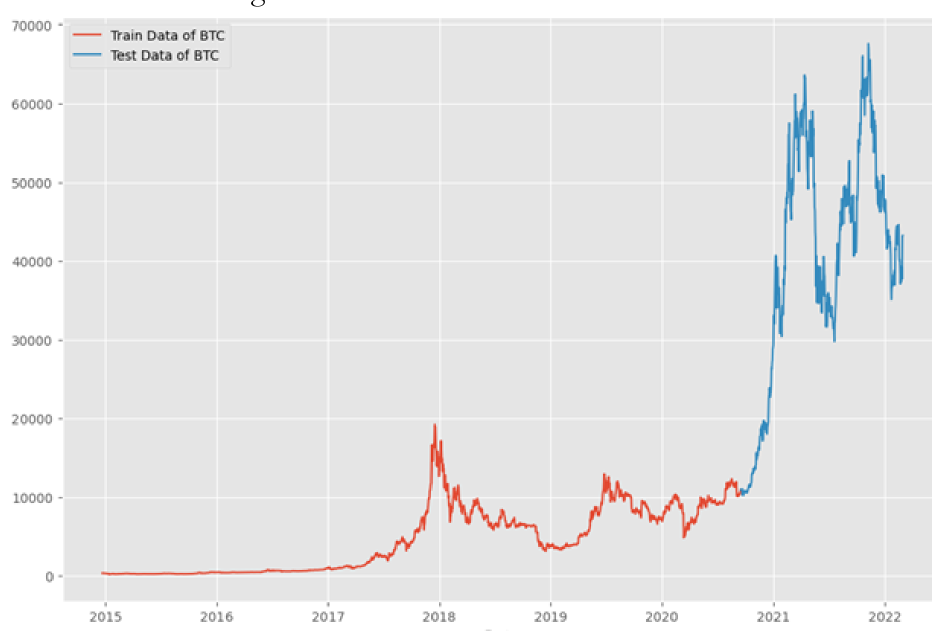

Figure 2: Dataset Splitting for Machine Learning Models (Linear Regressor, Random Forest Regressor, XGBoost Regressor)

Statistical Models:

Statistical models are mathematical representations of the relationships between variables, commonly used to explain, forecast, and characterize occurrences. These models are frequently used to test hypotheses, generate predictions, and deduce causal linkages. They are constructed using statistical tools and procedures. In our study focused on Bitcoin forecasting, we have chosen to explore two well-known statistical models: Autoregressive Integrated Moving Average (ARIMA) and Prophet.

ARIMA:

ARIMA is one of the particular statistical models used in forecasting and time series analysis [17]. It can be applied to study time series behavior comprising of characteristics such as trends or seasonality. The components of the ARIMA model are composed of three principles.

- **Auto Regression (AR):** This aspect shows how an observation is related to several prior observations (lagged terms). The current value's regression on past values is represented by it.
- **Integration (I):** This part refers to differencing raw observations to achieve stationarity in a time series. Time Series Analysis relies upon stationary data, which differencing seeks to provide.
- **Moving Average (MA):** This component depicts how an observation relates to the error from a moving average model that has been employed on lagged observations.

The parameters of an ARIMA model are traditionally notated as (p, d, q), where:

- p is the number of lag observations (AR order).
- d is the degree of differencing (I order).
- q is the size of the moving average window (MA order).

We have selected the ARIMA model because of its wide application in various fields such as finance, economics and environmental science for forecasting and time series analysis. The data splitting for the ARIMA model can be seen in Figure 3.

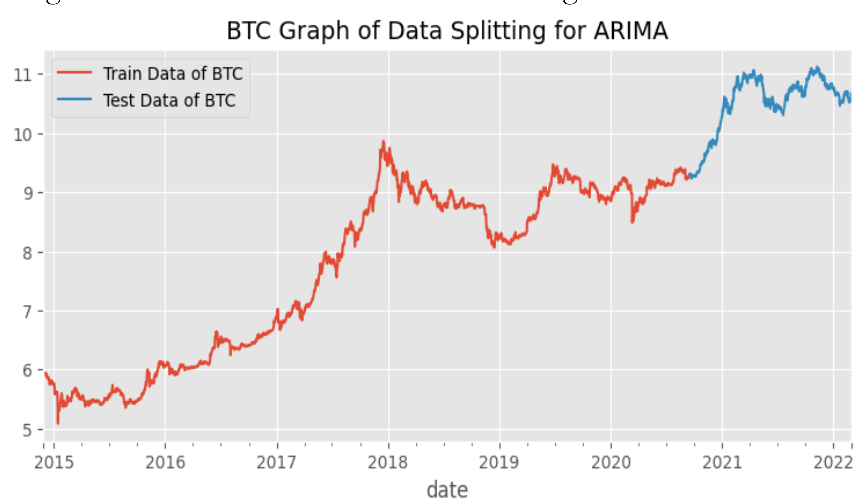

Figure 3: BTC Graph of Data Splitting for ARIMA

Prophet Model:

Various fields, including economics, finance and environmental science rely on time series forecasting. In recent years, the Prophet model has emerged as a dominant force in this field due to its simplicity, accuracy, and automation capabilities in capturing seasonality, trends, and holiday effects within time series data [16]. Unlike traditional models that need manual feature engineering, Prophet Model automates this process, making it accessible even to individuals with limited knowledge in this domain. It is an additive model composed of different components like holiday effects, trends and seasonality. The trend component for non-linear growth patterns in the data while the seasonality component captures repeating patterns at different time scales such as daily, weekly or annually. The mathematical representation of the Prophet model can be seen in Equation 3.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (3)$$

In the Equation 3, y(t) represents the observed value at time (t). The time series' overall direction is represented by the trend component, which is g(t). The seasonality component, s(t), captures periodic patterns (daily, weekly, yearly), the holiday component, h(t), accounts for the effects of holidays and special events, and the error term, t, represents the difference between the observed and predicted values. The data splitting for the Prophet model can be seen in Figure 4.

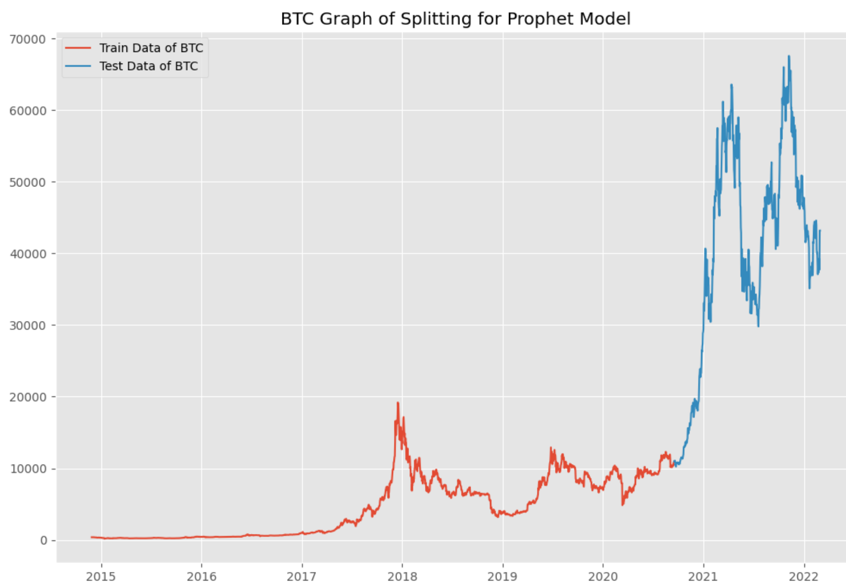

Figure 4: BTC Graph of Data Splitting for ARIMA

Deep Learning Models:

Utilized neural network architectures, such as Recurrent Neural Networks (RNNs) and their variations, such as Long Short-Term Memory (LSTM) networks and gated recurrent units (GRUs), to model and predict future values in a time series is known as deep learning for time series forecasting [17].

Convolutional Neural Network:

When forecasting with a Convolutional Neural Network (CNN), the time series data is typically viewed as an image, with the temporal dimension denoting the image's breadth and the features or channels denoting the image's height.

Recurrent Neural Network:

The RNN is one kind of neural network architecture designed to process sequential data. Unlike conventional feedforward neural networks, RNNs can maintain a memory of previous inputs through hidden states, making them suitable for handling sequences of varying lengths. Standard feedforward neural networks process inputs in a set order and lack memory. RNNs are frequently used to forecast time series data, like stock prices or, as in this case, the price of Bitcoin. The researchers can forecast future data points by using historical data analysis to spot trends. By training an RNN with historical Bitcoin price data, it can learn to predict future price fluctuations, aiding investors and traders in decision-making processes.

Nevertheless, RNNs have certain drawbacks, such as the vanishing gradient problem, which makes it challenging to learn long-term relationships. More sophisticated RNN variations, such as Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM), are frequently employed to overcome this issue because of their superior ability to recognize and retain long-term dependencies in sequential data.

Long Short-Term Memory:

Long Short-Term Memory (LSTM) [18] networks are a kind of RNN architecture that adeptly processes and predicts time series data. Unlike traditional RNNs, which often struggle with vanishing gradient problem to learn long-term dependencies. However, LSTMs were deliberately designed to remember information for longer time duration. The basic structure of an LSTM includes:

- **Cell State:** In an LSTM, the cell state runs straight down the whole length of the chain with only some linear interactions. It's like a moving walkway that adds or removes information but largely keeps it unchanged.
- **Gates:** LSTMs have three gates which control the flow of information:
- **Forget Gate:** Decides what part of memory should be let go from the cell state.
- **Input Gate:** Decides which new information to be stored in the cell state.
- **Output Gate:** Decides what output will be based on its cell state.
- **Hidden State:** This could be seen as the memory of LSTM. It performs computations and is carried over to the next time step.

Gated Recurrent Unit:

Gated Recurrent Unit (GRU) [19] comprises of a variety of RNN structures that was introduced as an easier version of the LSTM. GRUs, like LSTMs, are designed to model time series and natural language dependencies. The major components of GRU are as follows:

- **Reset Gate (r):** This determines of the extent to which the previous state should be forgotten or ignored. It considers both the current input and the previous hidden state.
- **Update Gate (z):** This decides extent to which a new state should be added to the current state. Similar to reset gate, current input and hidden states are calculated to update gate.
- **Current Memory Content (~h):** Based on the reset gate and the current input, any candidate's new state is calculated. It represents the information that is suggested to be included in the current version.
- **Hidden State (h):** The GRU unit's output is the hidden state, which is a weighted mixture of the information in memory at that moment and the previous hidden state.

There are many advantages of using GRU, some of which are listed below. These benefits are the main reasons to select this model for our study.

- GRUs are less computationally expensive and easier to train than LSTMs since they have fewer parameters.
- GRUs are a useful tool for identification of short-term dependencies in sequential data because of their effectiveness in doing so.
- Compared to LSTMs, GRUs frequently converge more quickly during training because of their simpler architecture.

Feedforward Neural Network:

The acronym for Feedforward Neural Network is FFN. Unlike recurrent neural networks, FNNs do not have cyclical connections, and information flows only in one direction: forward [20]. The major architectural components of FFN are as follows:

Input Layer: Data input is received by the input layer, which then forwards it to the hidden levels.

- **Hidden Layers:** Several neurons (nodes) make up each of the one or more hidden layers that may exist. Every neuron forwards the output to the subsequent layer after applying an activation function to the weighted sum of its inputs.
- **Output Layer:** This layer generates the network's final output. Depending on the task at hand (binary classification, multi-class classification, regression, etc.), this layer has a different number of neurons.
- **Activation Function:** To provide non-linearity to the model, each neuron in the output layer and hidden layers often applies an activation function (such as ReLU, Sigmoid, or Tanh).

There are many advantages of using FNN, some of which are listed below. These benefits are the main reasons to select this model for our study.

- If there are enough neurons in the hidden layers of an FNN, it can approximate any continuous function.
- FNNs can be effectively parallelized because information flows in a single route without loops, which qualifies them for hardware acceleration.
- When compared to more intricate neural network topologies, FNNs are comparatively simple to comprehend and use.

Results and Discussion:

In this section, we offer a thorough examination of the evaluation metrics, experimental methodology, and the results obtained utilizing various models.

Evaluation Matrices:

In the context of machine learning and deep learning, evaluation metrics were used to gauge a model's effectiveness on a certain job. These indicators are crucial for understanding the benefits and drawbacks of a model as well as for comparing different models with one another. For this study, we have chosen MAPE as the evaluation metric.

Mean Absolute Percentage Error:

Mean Absolute Percentage Error is referred to as MAPE [21]. It is a metric used to assess the forecasting method's prediction accuracy in statistics, such as trend estimation. It is determined by the formula of equation 4.

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \qquad (4)$$

The forecast value is represented by F_t in equation 4, the actual value is denoted by A_t, and the number of iterations in the summation is indicated by n. The Mean Absolute Percentage Error (MAPE) is represented by M.

Experimental Results:

The models in this study were implemented using Python packages such as Sklearn, Keras, and TensorFlow. The techniques were applied in Python 3.10 and Google Colab, making use of its GPU resources for efficient processing.

The outputs of machine learning model, showcasing the actual and anticipated values are represented graphically in Figures 5 for the Linear Regressor, Figure 6 for the Random Forest

Regressor, and Figure 7 for the XGBoost Regressor. These outputs throw light on each model's effectiveness and performance by displaying a visual comparison of the expected and actual outcomes.

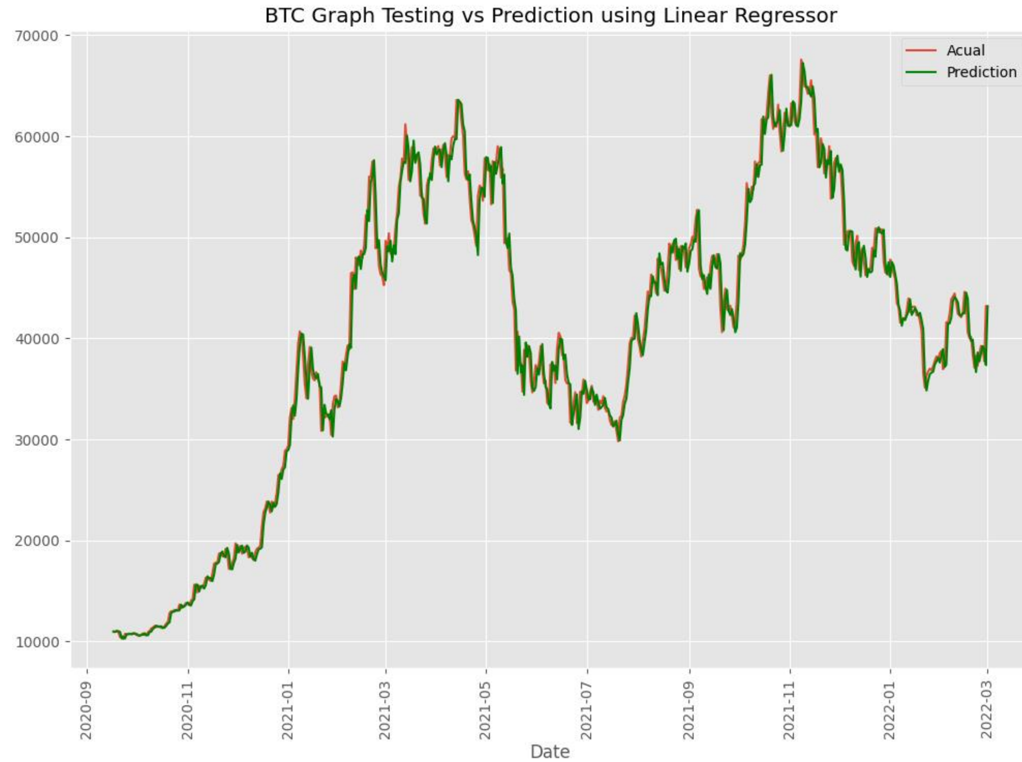

Figure 5: Actual Testing and Prediction Using Linear Regressor Model

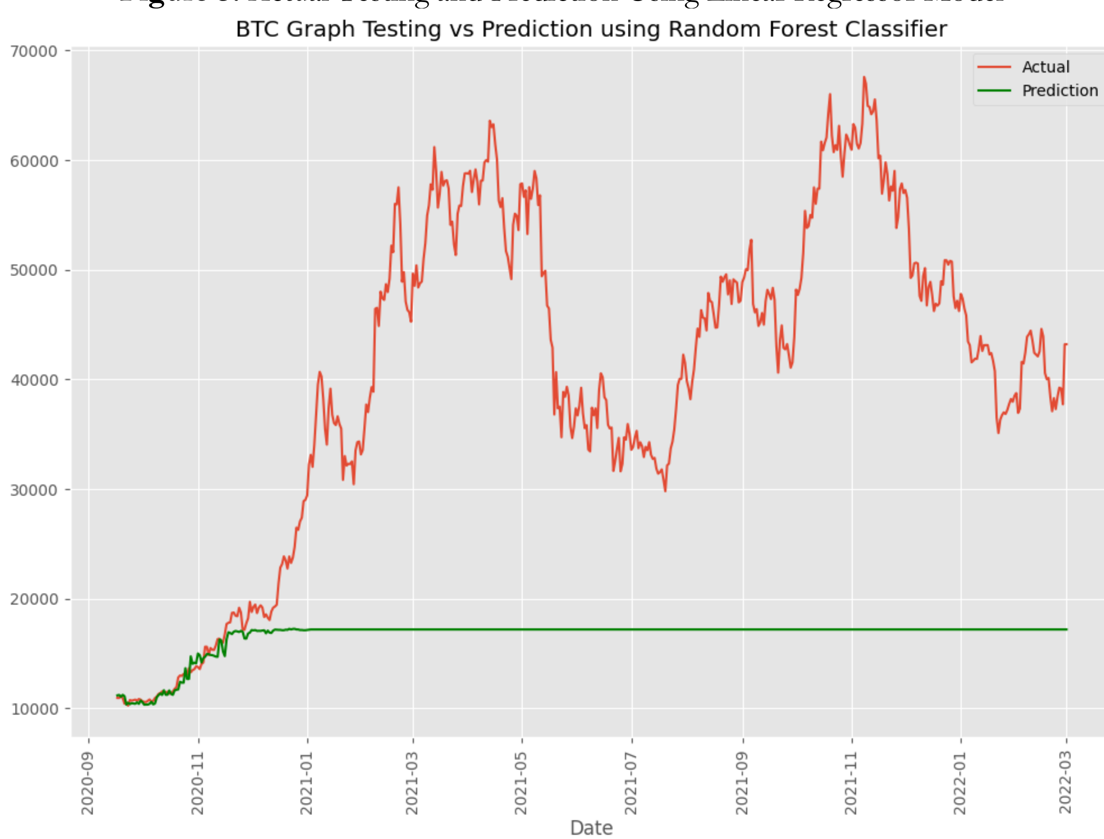

Figure 6: Actual Testing and Prediction Using Random Forest Model

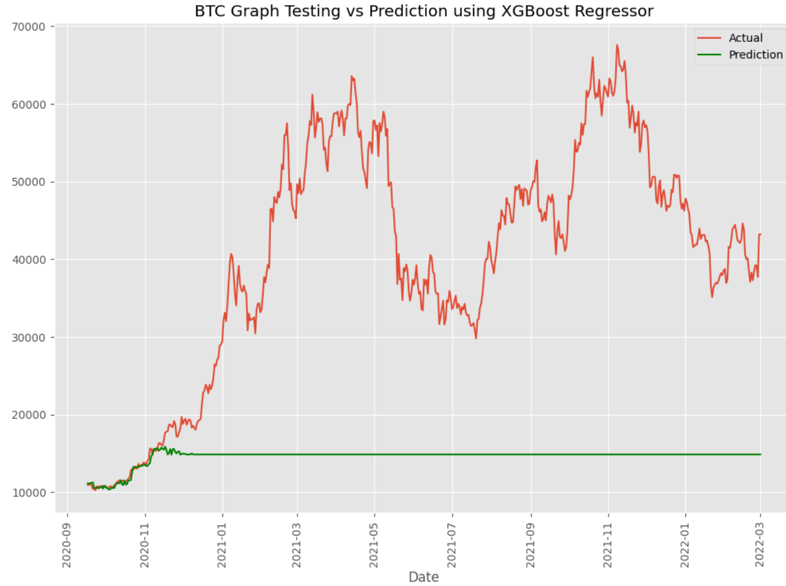

Figure 7: Actual Testing and Prediction using XGBoost Regressor Model

Figure 8 and Figure 9 depict the graphical representation of the statistical models in terms of actual vs predicted values of BTC close rate.

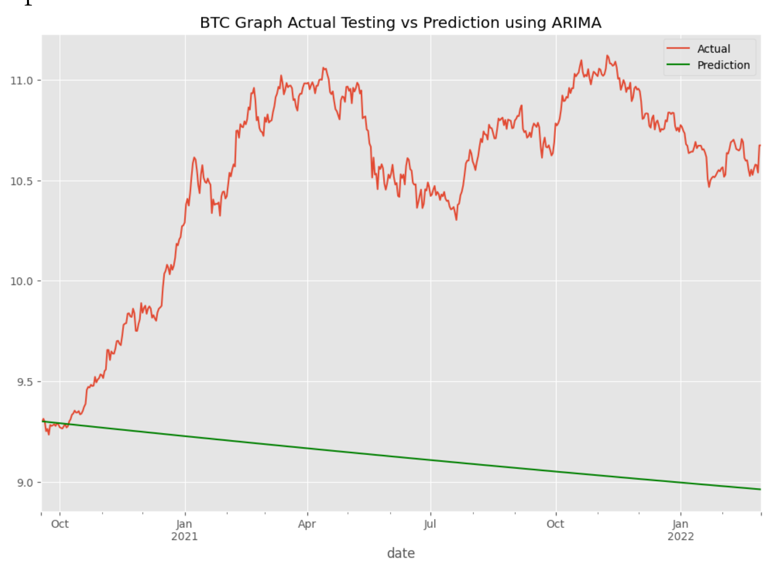

Figure 8: Actual Testing and Prediction using ARIMA Model

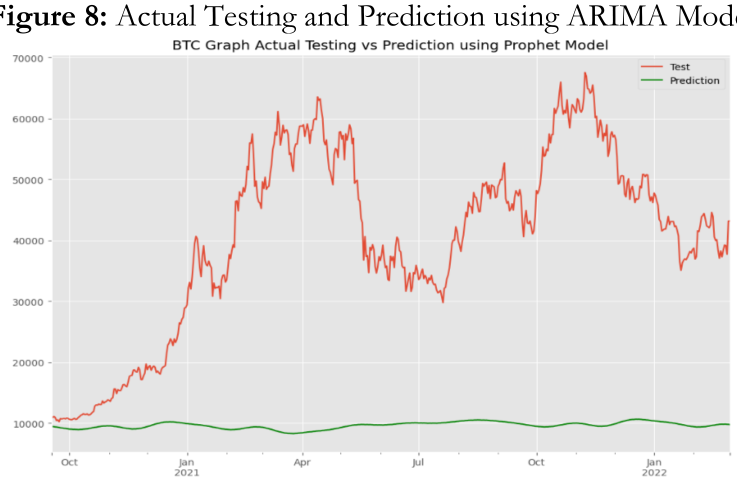

Figure 9: Actual Testing and Prediction using Prophet Model

The deep learning models were trained consistently using 30 epochs and a batch size of 16. The outcomes of these trained models are visually presented through multiple figures, ranging from Figure 10 to Figure 14. These graphs provide a thorough summary of each model's performance and predictive power, enabling comparisons and insights into how well they work with various tasks and datasets.

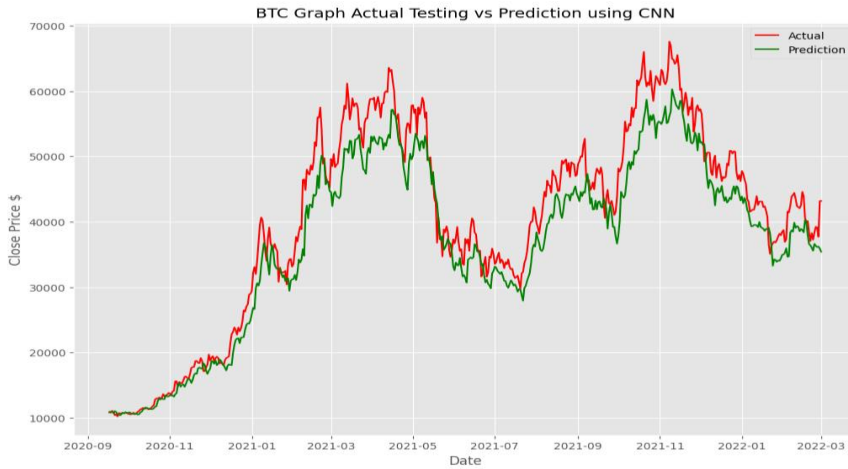

Figure 10: Actual Testing and Prediction using CNN Model

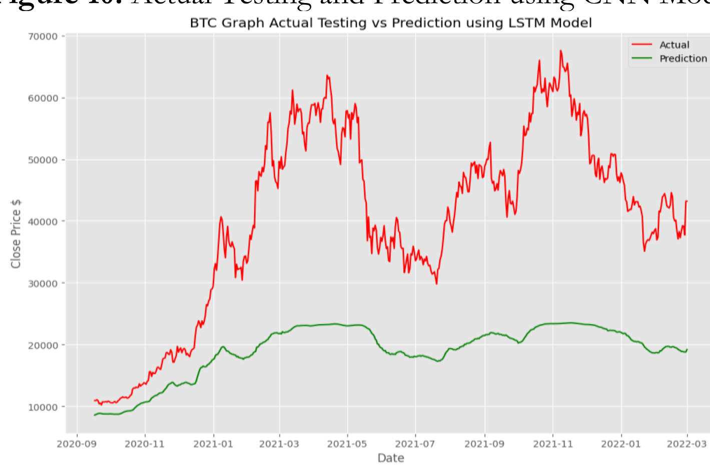

Figure 11: Actual Testing and Prediction using the LSTM Model

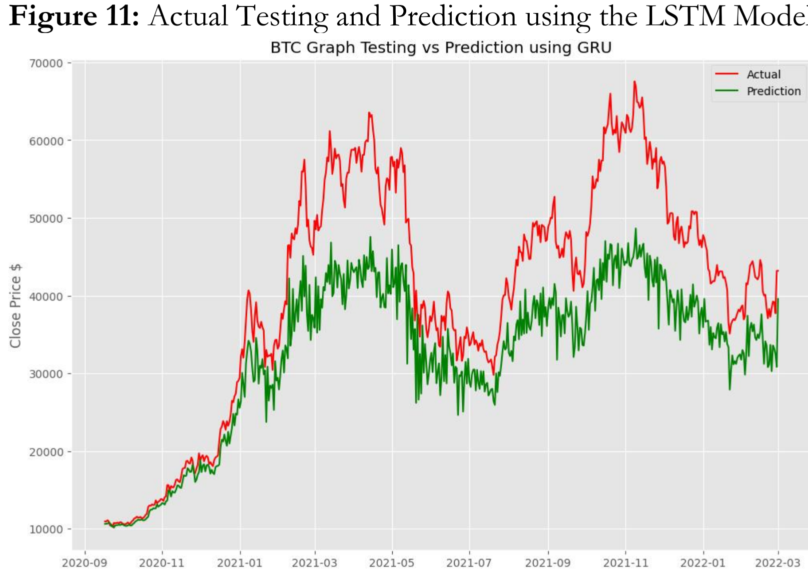

Figure 12: Actual Testing and Prediction using GRU Model

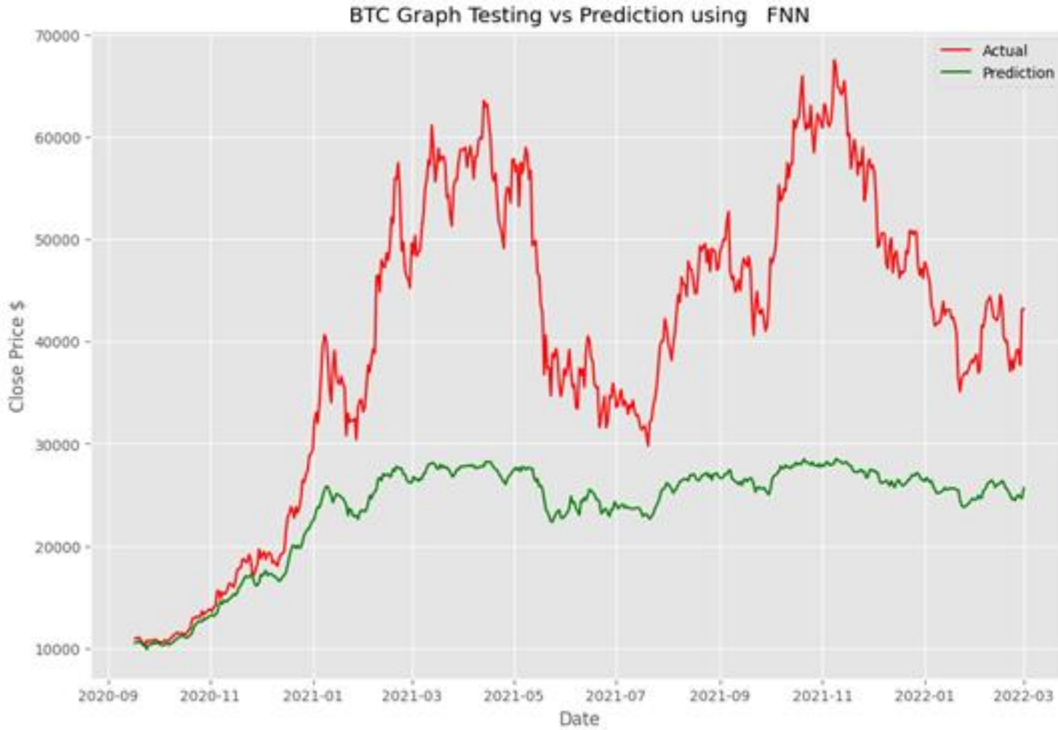

Figure 13: Actual Testing and Prediction using FNN Model

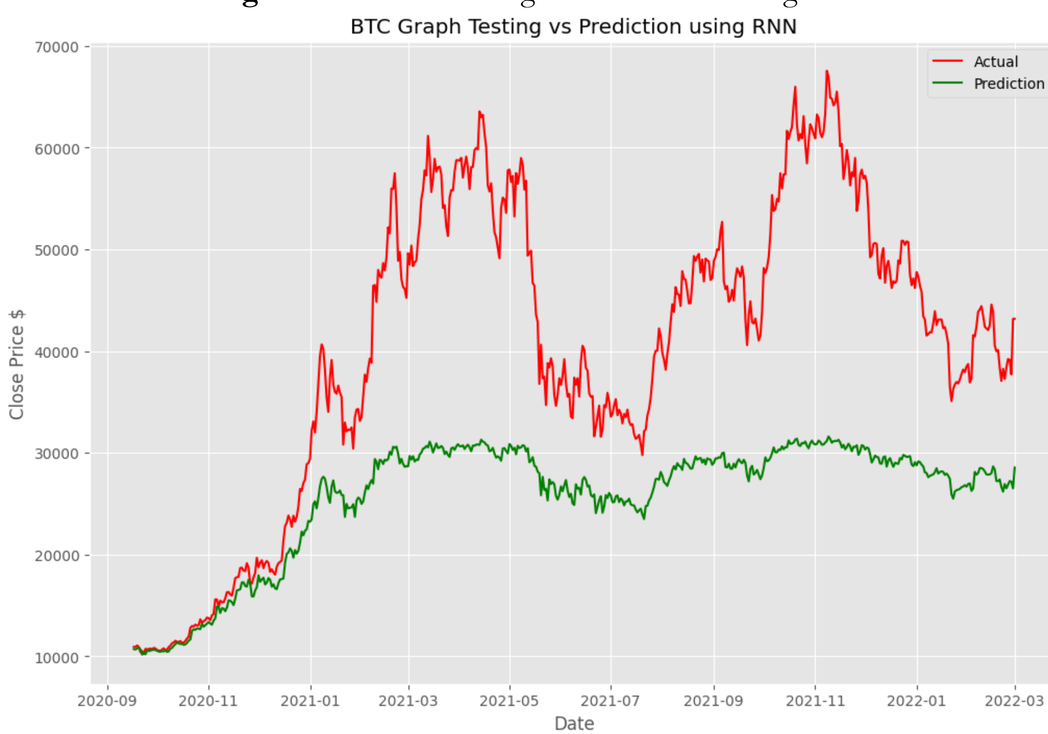

Figure 14: Actual Testing and Prediction using RNN Model

In addition to evaluating the models based on MAPE scores, the training time of each model is also taken into account to determine most robust model, as illustrated in Table 2. By considering training time, the main goal of this study is to identify a model that predicts (low MAPE) as well as trains efficiently, significant for real-world applications in which computational resources and time are limited. This approach ensures that the selected model is both accurate and practical for deployment. From Table 2, it is observable that Linear Regressor is more effective than other models.

Table 2: Comparison of Forecasting Models concerning MAPE Score and Training Time

Model	MAPE Test	Training Time
Linear Regressor	2.9989	0.0336
Random Forest Regressor	52.7365	7.6400
XGBoost Regressor	55.9155	0.4236
ARIMA	12.96	0.1693
Prophet	69.91	1.6017
CNN	7.9999	13.9089
LSTM	48.1873	158.4398
GRU	17.2573	151.5812
RNN	31.1235	86.8800
FNN	35.5801	13.2750

To improve our forecasting performance, we included the PyCaret library [22] to our forecasting framework in addition to already stated model. Building, optimizing, and implementing machine learning models (because ML plays an important role in different fields [23][24][25][26][27][28][29][30][31][32]) is made easier with PyCaret, a flexible machine learning framework. To discover the best model for our objective of predicting cryptocurrency prices, we experimented with several machine learning algorithms and setups, making use of PyCaret's capability to speed the forecasting process. We were able to rapidly iterate through several models and hyperparameters by utilizing PyCaret, which allowed us to save a significant amount of time and money. Table 3 provides an overview of our PyCaret-enhanced forecasting approach's outcomes and demonstrates how well PyCaret has helped us increase the accuracy of our forecasts.

Table 3. Comparison of Forecasting Models by PyCaret concerning Training MAPE Score and Training Time

Model	MAPE Training	Training Time
Extra Trees w/ Cond. Deseasonalize & Detrending	0.0203	2.600
Lasso Least Angular Regressor w/ Cond. Deseasonalize & Detrending	0.0203	0.3233
Linear w/ Cond. Deseasonalize & Detrending	0.0203	0.8600
Ridge w/ Cond. Deseasonalize & Detrending	0.0203	0.3167
Bayesian Ridge w/ Cond. Deseasonalize & Detrending	0.0238	0.3267
Elastic Net w/ Cond. Deseasonalize & Detrending	0.0238	0.533
Lasso w/ Cond. Deseasonalize & Detrending	0.0238	0.3867
Random Forest w/ Cond. Deseasonalize & Detrending	0.0266	6.5233

Light Gradient Boosting w/ Cond. Deseasonalize & Detrending	0.0272	1.8133
Decision Tree w/ Cond. Deseasonalize & Detrending	0.0285	0.6133
Orthogonal Matching Pursuit w/ Cond. Deseasonalize & Detrending	0.0281	0.3200
K Neighbors w/ Cond. Deseasonalize & Detrending	0.0283	0.4100
Gradient Boosting w/ Cond. Deseasonalize & Detrending	0.0290	3.4800
Naive Forecaster	0.0306	3.3467
ETS	0.0308	1.8100
Huber w/ Cond. Deseasonalize & Detrending	0.0310	0.3767
Extreme Gradient Boosting w/ Cond. Deseasonalize & Detrending	0.0316	0.9033
Exponential Smoothing	0.0326	5.1433
Theta Forecaster	0.0334	0.0767
AdaBoost w/ Cond. Deseasonalize & Detrending	0.0367	0.8200
STLF	0.0369	0.1900
ARIMA	0.0448	2.6467
Croston	0.0645	0.0400
Seasonal Naive Forecaster	0.0742	0.1600
Polynomial Trend Forecaster	0.1430	0.0433
Grand Means Forecaster	0.6993	0.0533

The ExtraTreesRegressor is the most effective model in your study for predicting bitcoin prices. It obtained a test MAPE score of 0.0609 when examined for out-of-sample (testing) with a forecast horizon of 36.

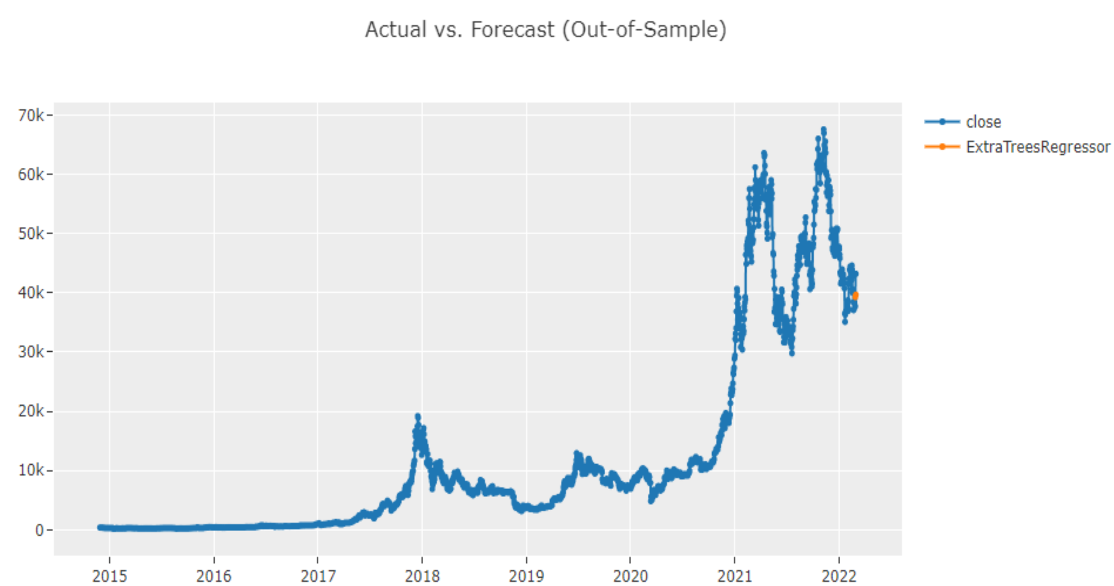

Figure 15: Actual Testing and Prediction using ExtraTreesRegressor Model

Conclusion:

We examined the performance of Deep Learning (DL) and conventional models in predicting the prices of cryptocurrencies. With a MAPE test score of 0.0609, we discovered that the Extra Trees Regressor outperforms all other models, including DL models like LSTM, ARIMA, and Prophet, after using PyCaret with all forecasting models and other well-known models. This demonstrates accuracy of models for predicting bitcoin prices. Subsequent investigations may examine supplementary variables impacting prices, and ensemble models may further improve forecast precision. Our research highlights significance of model selection and ensemble techniques to get precise bitcoin price forecasts.

References:

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", Accessed: Apr. 05, 2024. [Online]. Available: www.bitcoin.org

[2] D. G. Baur and T. Dimpfl, "The volatility of Bitcoin and its role as a medium of exchange and a store of value," Empir. Econ., vol. 61, no. 5, pp. 2663–2683, Nov. 2021, doi: 10.1007/S00181-020-01990-5.

[3] "Bitcoin Laundering: An Analysis of Illicit Flows into Digital Currency Services", Accessed: Apr. 05, 2024. [Online]. Available: https://www.mail-archive.com/cryptography@

[4] M. Xu, X. Chen, and G. Kou, "A systematic review of blockchain," Financ. Innov., vol. 5, no. 1, Dec. 2019, doi: 10.1186/S40854-019-0147-Z.

[5] J. Y. Huang and J. H. Liu, "Using social media mining technology to improve stock price forecast accuracy," J. Forecast., vol. 39, no. 1, pp. 104–116, Jan. 2020, doi: 10.1002/FOR.2616.

[6] S. McNally, J. Roche, and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," Proc. - 26th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. PDP 2018, pp. 339–343, Jun. 2018, doi: 10.1109/PDP2018.2018.00060.

[7] Y. Chen, X. Xie, T. Zhang, J. Bai, and M. Hou, "A deep residual compensation extreme learning machine and applications," J. Forecast., vol. 39, no. 6, pp. 986–999, Sep. 2020, doi: 10.1002/FOR.2663.

[8] A. Aggarwal, I. Gupta, N. Garg, and A. Goel, "Deep Learning Approach to Determine the Impact of Socio Economic Factors on Bitcoin Price Prediction," 2019 12th Int. Conf. Contemp. Comput. IC3 2019, Aug. 2019, doi: 10.1109/IC3.2019.8844928.

[9] A. García-Medina and T. L. D. Huynh, "What drives bitcoin? An approach from continuous local transfer entropy and deep learning classification models," Entropy, vol. 23, no. 12, p. 1582, Dec. 2021, doi: 10.3390/E23121582/S1.

[10] E. Akyildirim, A. Goncu, and A. Sensoy, "Prediction of cryptocurrency returns using machine learning," Ann. Oper. Res., vol. 297, no. 1–2, pp. 3–36, Feb. 2021, doi: 10.1007/S10479-020-03575-Y/TABLES/18.

[11] "Bitcoin Historical Dataset." Accessed: Apr. 05, 2024. [Online]. Available: https://www.kaggle.com/datasets/prasoonkottarathil/btcinusd

[12] "(PDF) Forecasting accuracy of machine learning and linear regression: evidence from the secondary CAT bond market." Accessed: Apr. 05, 2024. [Online]. Available: https://www.researchgate.net/publication/369010278_Forecasting_accuracy_of_machine_learning_and_linear_regression_evidence_from_the_secondary_CAT_bond_market

[13] Shalabh, "A revisit to efficient forecasting in linear regression models," J. Multivar. Anal., vol. 114, no. 1, pp. 161–170, Feb. 2013, doi: 10.1016/J.JMVA.2012.07.017.

[14] Z. El Mrabet, N. Sugunaraj, P. Ranganathan, and S. Abhyankar, "Random Forest Regressor-Based Approach for Detecting Fault Location and Duration in Power Systems," Sensors 2022, Vol. 22, Page 458, vol. 22, no. 2, p. 458, Jan. 2022, doi: 10.3390/S22020458.

[15] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-August-2016, pp. 785–794, Mar. 2016, doi: 10.1145/2939672.2939785.

[16] M. Kumar and M. Thenmozhi, "Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models," Int. J. Banking, Account. Financ., vol. 5, no. 3, pp. 284–308, 2014, doi: 10.1504/IJBAAF.2014.064307.

[17] D. Chandola, A. Mehta, S. Singh, V. A. Tikkiwal, and H. Agrawal, "Forecasting Directional

Movement of Stock Prices using Deep Learning," Ann. Data Sci., vol. 10, no. 5, pp. 1361–1378, Oct. 2023, doi: 10.1007/S40745-022-00432-6/TABLES/3.
[18] K. A. Al-Utaibi, S. Siddiq, and S. M. Sait, "Stock Price Forecasting with LSTM: A Brief Analysis of Mathematics Behind LSTM," https://doi.org/10.1142/S1793048023410047, pp. 1–14, Aug. 2023, doi: 10.1142/S1793048023410047.
[19] J. Hu, Q. Chang, and S. Yan, "A GRU-based hybrid global stock price index forecasting model with group decision-making," Int. J. Comput. Sci. Eng., vol. 26, no. 1, pp. 12–19, 2023, doi: 10.1504/IJCSE.2023.129153.
[20] J. Khan, E. Lee, and K. Kim, "A higher prediction accuracy–based alpha–beta filter algorithm using the feedforward artificial neural network," CAAI Trans. Intell. Technol., vol. 8, no. 4, pp. 1124–1139, Dec. 2023, doi: 10.1049/CIT2.12148.
[21] "3.3. Metrics and scoring: quantifying the quality of predictions — scikit-learn 1.4.1 documentation." Accessed: Apr. 05, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html
[22] "Home - PyCaret." Accessed: Apr. 05, 2024. [Online]. Available: https://pycaret.org/
[23] M. A. Arshed, W. Qureshi, M. Rumaan, M. T. Ubaid, A. Qudoos, and M. U. G. Khan, "Comparison of Machine Learning Classifiers for Breast Cancer Diagnosis," 4th Int. Conf. Innov. Comput. ICIC 2021, 2021, doi: 10.1109/ICIC53490.2021.9692926.
[24] H. Younis, M. Asad Arshed, F. ul Hassan, M. Khurshid, H. Ghassan, and M. Haseeb-, "Tomato Disease Classification using Fine-Tuned Convolutional Neural Network," Int. J. Innov. Sci. Technol., vol. 4, no. 1, pp. 123–134, Feb. 2022, doi: 10.33411/IJIST/2022040109.
[25] A. Shahzad, M. A. Arshed, F. Liaquat, M. Tanveer, M. Hussain, and R. Alamdar, "Pneumonia Classification from Chest X-ray Images Using Pre-Trained Network Architectures," VAWKUM Trans. Comput. Sci., vol. 10, no. 2, pp. 34–44, Dec. 2022, doi: 10.21015/VTCS.V10I2.1271.
[26] M. Mubeen, M. A. Arshed, and H. A. Rehman, "DeepFireNet - A Light-Weight Neural Network for Fire-Smoke Detection," Commun. Comput. Inf. Sci., vol. 1616 CCIS, pp. 171–181, 2022, doi: 10.1007/978-3-031-10525-8_14/COVER.
[27] M. A. Arshed, S. Mumtaz, M. Hussain, R. Alamdar, M. T. Hassan, and M. Tanveer, "DeepFinancial Model for Exchange Rate Impacts Prediction of Political and Financial Statements," 3rd IEEE Int. Conf. Artif. Intell. ICAI 2023, pp. 13–19, 2023, doi: 10.1109/ICAI58407.2023.10136658.
[28] M. T. M. A. Arshed, A. Shahzad, K. Arshad, D. Karim, S. Mumtaz, "Multiclass Brain Tumor Classification from MRI Images using Pre-Trained CNN Model," VFAST trans. softw. eng., vol. 10, no. 4, pp. 22–28, 2022.
[29] M. A. Arshed and F. Riaz, "Machine Learning for High Risk Cardiovascular Patient Identification 1," J. Distrib. Comput. Syst., vol. 4, no. 2, pp. 34–39, 2021, Accessed: Dec. 26, 2023. [Online]. Available: https://ijdcs.ir/wp-content/uploads/2022/08/IJDCS-8-6.pdf
[30] H. M. H. J. Aslam, M. A. Arshed, S. Iqbal, "Deep Learning Based Multi-Class Eye Disease Classification: Enhancing Vision Health Diagnosis," Tech. J., vol. 29, no. 1, pp. 7–12.
[31] M. Hussain, A. Shahzad, F. Liaquat, M. A. Arshed, S. Mansoor, and Z. Akram, "Performance Analysis of Machine Learning Algorithms for Early Prognosis of Cardiac Vascular Disease," Tech. J., vol. 28, no. 02, pp. 31–41, Jun. 2023, Accessed: Dec. 26, 2023. [Online]. Available: https://tj.uettaxila.edu.pk/index.php/technical-journal/article/view/1778
[32] M. A. Arshed et al., "Machine Learning with Data Balancing Technique for IoT Attack and Anomalies Detection," Int. J. Innov. Sci. Technol., vol. 4, no. 2, pp. 490–498, 2022, doi: 10.33411/ijist/2022040218.