ChihChin Yang
CSC 680
12/06/2022

Final Project

Bitcoin Stock Price Prediction

Introduction

The goal of this study is to predict prices for Bitcoin using Machine Learning Techniques for the next day and prepare a strategy to maximize gains for investors. One of the most unpredictable datasets for machine learning is the stock market. Any success in this subject would practically allow people to create money because analysts have long tried to predict the stock market. In particular, Bitcoin, has seen a significant increase in popularity among investors worldwide in recent years. Due to the novelty of the cryptocurrency transaction system, there is a great deal of uncertainty among investors, and it has been asserted that any rumors or news on social media would have a big impact on the pricing of cryptocurrencies. Still, in a market where investors and experts are always looking for the best time trend to invest in. To achieve this goal, it is necessary to improve the prediction model's ability to predict cryptocurrency trends accurately and consistently. In this report, I will present a detailed description of the machine learning techniques that employ to address the persistent problem of enhancing the ability to successfully predict whether Bitcoin stocks will increase or decrease on a given day.

Dataset

Kaggle https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data
CSV files for select bitcoin exchanges for the time period of Jan 2012 to December March 2021, with minute to minute updates of OHLC (Open, High, Low, Close), Volume in BTC and indicated currency, and weighted bitcoin price. Timestamps are in Unix time. Timestamps without any trades or activity have their data fields filled with NaNs. If a timestamp is missing, or if there are jumps, this may be because the exchange (or its API) was down, the exchange (or its API) did not exist, or some other unforeseen technical error in data reporting or gathering.

Task addressed

1. Time series analysis:

Time Series is a series of observations taken at specified time intervals usually equal intervals. Analysis of the series helps us to predict future values based on previous observed values. In Time series, we have only 2 variables, time & the variable we want to forecast.

2. Components of Time Series:

There are 4 components:

Trend - Upward & downward movement of the data with time over a large period of time. Ex: Appreciation of Dollar vs rupee.
Seasonality - seasonal variances. Ex: Icecream sales increases in Summer only
Noise or Irregularity - Spikes & troughs at random intervals
Cyclicity - behavior that repeats itself after large interval of time, like months, years etc

Method and Model

To solve the problem using Machine learning, I first tried to categorize the situation and tried to find previous solutions on how they solved it. I quickly learned that, since the issue involves prices that are changing with time, this could be modeled as a Series prediction problem. Also, as a machine learning problem with the features being the previous prices and the output being the price predicted for that day.

1. ARIMA

Autoregressive integrated moving average (ARIMA) is a statistical regression model, which can be utilized in time series forecasting applications, such as finance. ARIMA makes predictions while considering the lagged values of a time series, while accommodating for non-stationarity. The model, which is one of the most popular linear models for time series forecasting, originates from the autoregressive (AR) and moving average (MA) models, as well as their combination, also known as ARMA. For making predictions with the ARIMA model, we had to follow a step-by-step procedure to be able to feed the data to the ARIMA model. This first involved Visualizing the time series data: It is essential to analyze the trends prior to building any kind of time series model. The details we are interested in pertains to any kind of trend, seasonality or random behavior in the series.

ARIMA(Auto Regressive Integrated Moving Average) is a combination of 2 models AR(Auto Regressive) & MA(Moving Average). It has 3 hyperparameters - P(auto regressive lags),d(order of differentiation),Q(moving avg.) which respectively comes from the AR, I & MA components. The AR part is correlation between prev & current time periods. To smooth out the noise, the MA part is used. The I part binds together the AR & MA parts. In order to find the values of P and Q, We need to take help of ACF(Auto Correlation Function) & PACF(Partial Auto Correlation Function) plots. ACF & PACF graphs are used to find value of P & Q for ARIMA. We need to check, for which value in x-axis, graph line drops to 0 in y-axis for 1st time. From PACF(at y=0), get P From ACF(at y=0), get Q.

2. Random Forest

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 and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For b = 1, …, B:

Sample, with replacement, n training examples from X, Y; call these Xb, Yb. Train a classification or regression tree fb on Xb, Yb. After training, predictions for unseen samples x’ can be made by averaging the predictions from all the individual regression trees on x’ or by taking the majority vote in the case of classification trees.

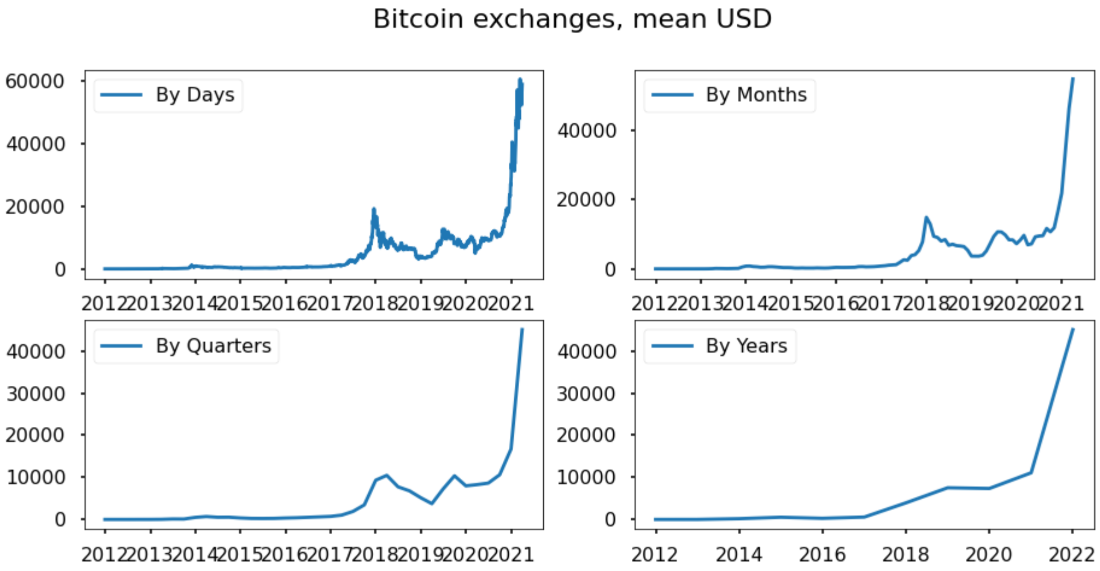
This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets.

3. Recurrent Neural Net

RNNs provide a generalization of the feed forward network model for dealing with sequential data, with the addition of an ongoing internal state the serving as memory buffer for processing sequences.

Experiments and Evolutions

Data Exploration:



Stationarity:

Time Series(TS) need to be stationary because,

1. If a TS has a particular behavior over a time interval, then there's a high probability that over a different interval, it will have same behavior, provided TS is stationary. This helps in forecasting accurately.
2. Theories & Mathematical formulas ae more mature & easier to apply for as TS which is stationary.

Before applying any statistical model on a Time Series, the series has to be stationary, which means that, over different time periods,

1. It should have constant mean.
2. It should have constant variance or standard deviation.
3. Auto-covariance should not depend on time.

Trend & Seasonality are two reasons why a Time Series is not stationary & hence need to be corrected.

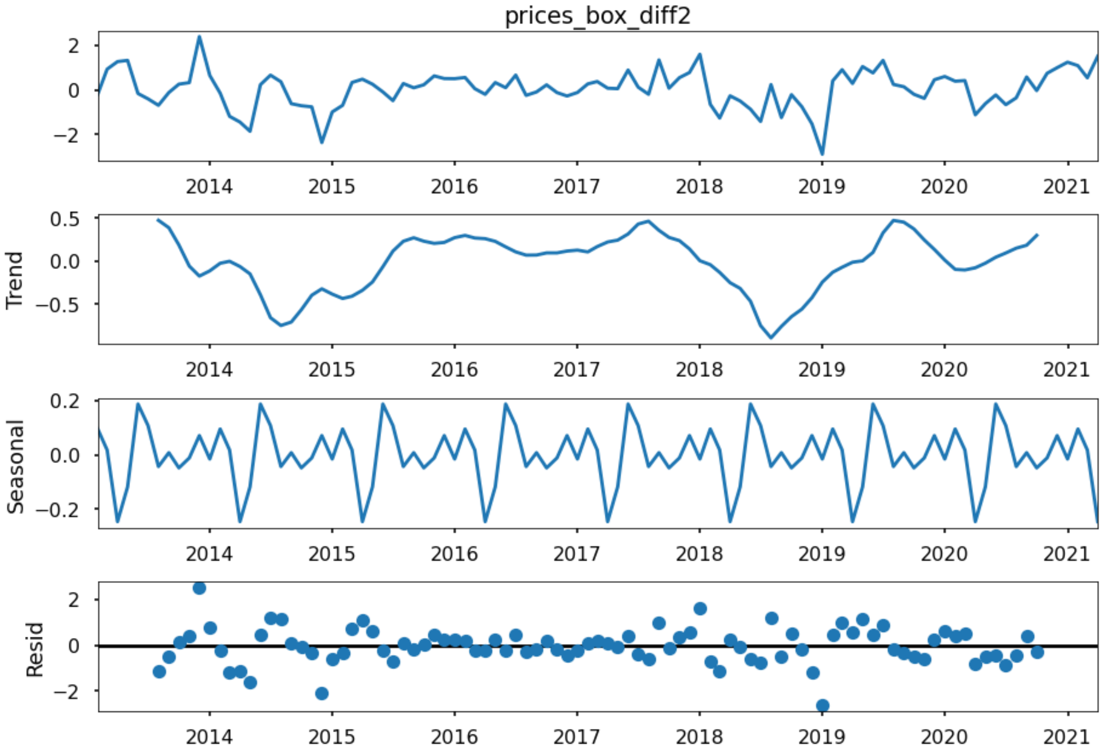
There are 2 ways to check for Stationarity of a TS:

1. Rolling Statistics - Plot the moving avg or moving standard deviation to see if it varies with time. Its a visual technique.
2. ADCF Test - Augmented Dickey–Fuller test is used to gives us various values that can help in identifying stationarity. The Null hypothesis says that a TS is

non-stationary. It comprises of a Test Statistics & some critical values for some confidence levels. If the Test statistics is less than the critical values, we can reject the null hypothesis & say that the series is stationary. THE ADCF test also gives us a p-value. Acc to the null hypothesis, lower values of p is better.

Stationarity check and seasonal decomposition:

Decompose seasonal component of the time series.

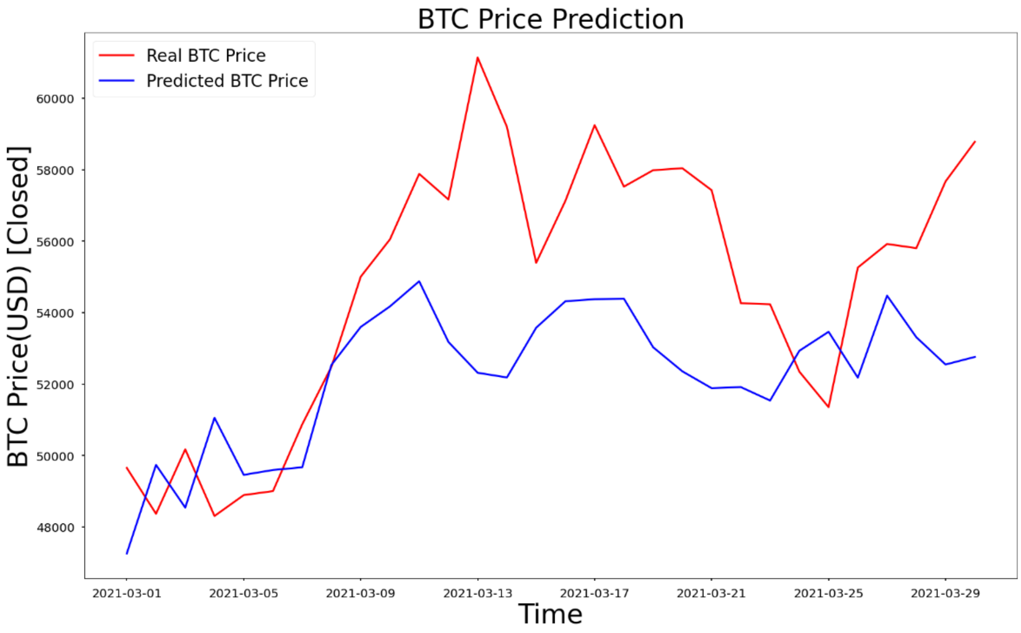


The series are not stationary.

After reviewing the dataset, I think I need to recode the datetime since it is better to use the dataset sorted by date not by minutes. Then I group the dataset by date and take the average price of all minutes in the day as the price of the day.

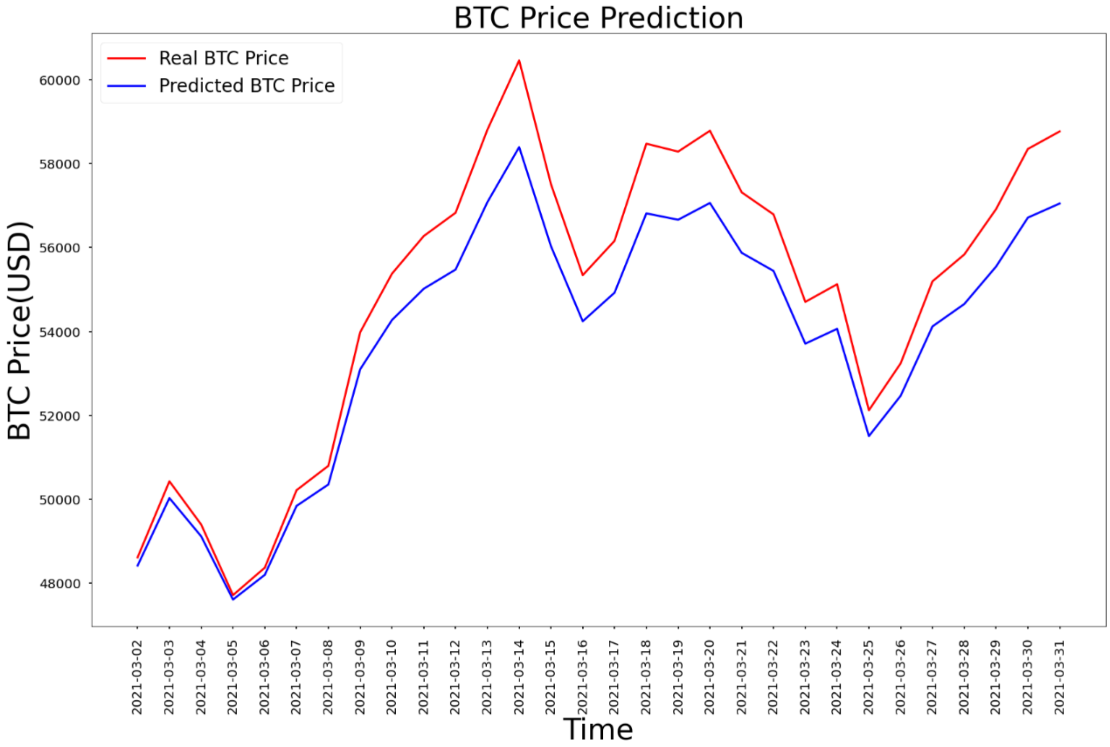
1. **Random Forest**

   The result of random forest model trained on data set from 2012 to 2021 tested on March 2021. Model use window size 5 and 1000 epochs during training and testing. The result is predict the next 30 days.



2. RNN

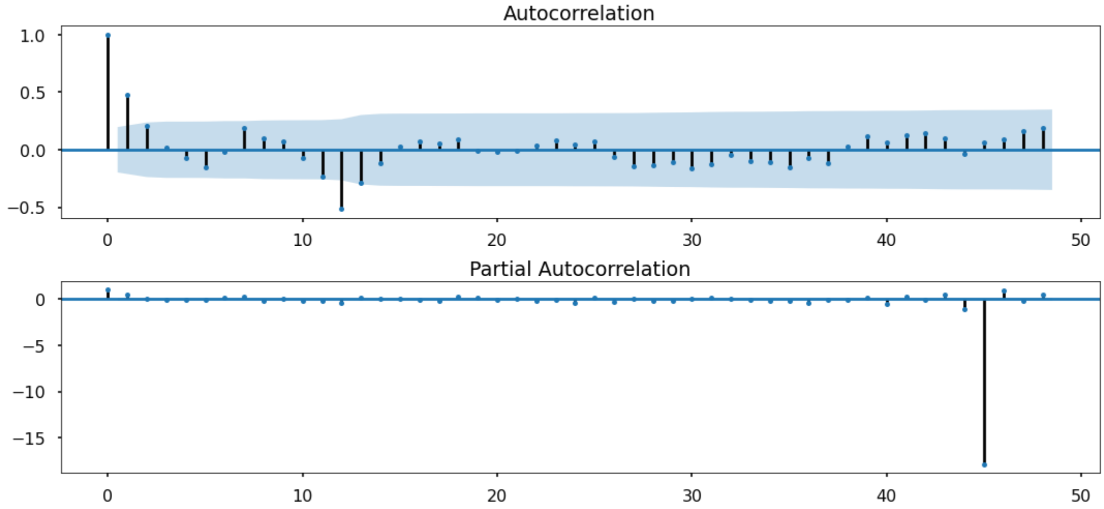
Compare the difference of predicted price and the real price. The difference is larger when the time is further to the training set. That is why I only want to predict the price of one month.



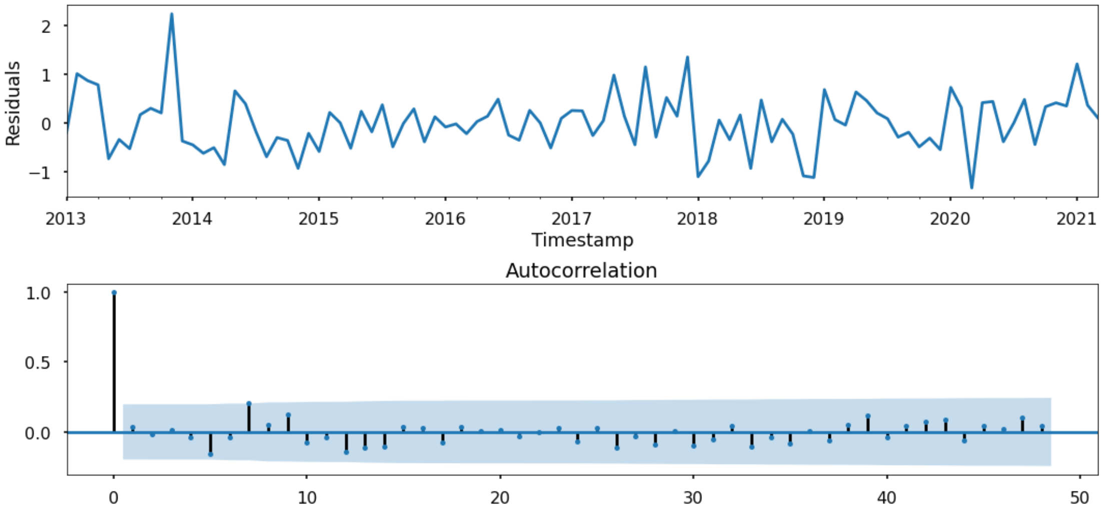
3. ARIMA

According to the smallest AIC, the optimal parameters are determined to be p=1, q=0, so the model is ARIMA(1,1,0)

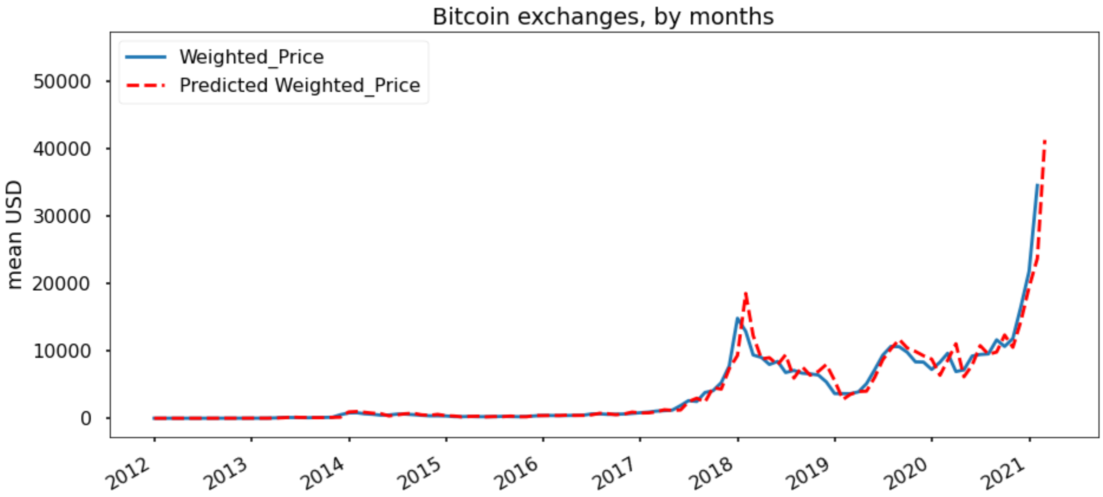
Model Selection:

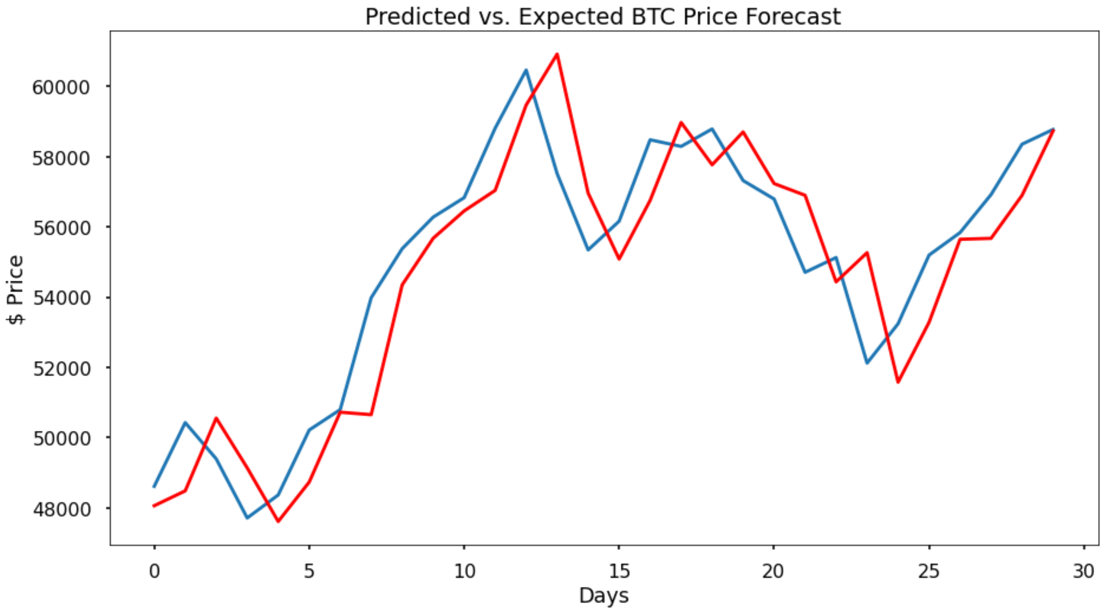


Analysis of residues



Predictions:





Compare the three model, ARIMA has the best result. Because Bitcoin has drastic changes from 2020 to 2021, the error value will increase. If you only look at the error value before 2020, the three methods can be controlled within 1000.

Model	RMSE
Random Forest	**$3,684**
RNN	**$1,205**
ARIMA	**$626**

Future work

Now, to investigate the claim of bitcoin related news and tweets affecting the bitcoin prices, so for the future work should add News data (Text to Vectors) or other factors (mining cost, cryptocurrency).

Reference

McNally, S., Roche, J., & Caton, S. (2018, March). Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)* (pp. 339-343). IEEE.

T. Phaladisailoed and T. Numnonda, "Machine Learning Models Comparison for Bitcoin Price Prediction," *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2018, pp. 506-511, doi: 10.1109/ICITEED.2018.8534911.