Bitcoin Price Prediction
with Machine learning algorithms

ریحانه زرین - یاسمین عالم زاده

Abstract

After cryptocurrencies made significant waves in the investment industry in recent years, Bitcoin has increasingly gained recognition as a valuable investment asset. This paper addresses the challenging task of predicting Bitcoin prices, a critical aspect in the volatile cryptocurrency market. Employing a comprehensive approach, we explore the effectiveness of three classification models—Logistic Regression, Decision Tree Classifier, and K-Neighbors Classifier—in forecasting the directional movement of Bitcoin prices for the following day, deciding whether the price will go up or down. Furthermore, we delve into more precise predictions using advanced techniques, leveraging the power of XGBoost, a gradient boosting algorithm, and Long Short-Term Memory (LSTM) neural networks.

Keywords: **bitcoin, machine learning, LSTM, XGBoost, classification models.**

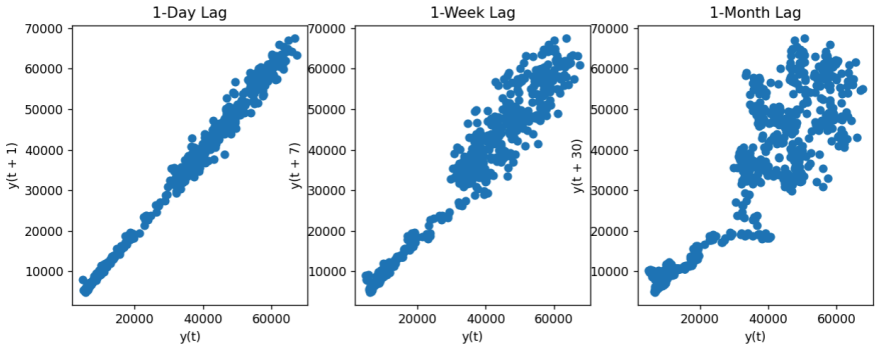


Fig 1 - the randomness is visible by the month lag, further showing the instability of prices.

1. Introduction

Cryptocurrencies, particularly Bitcoin, have transformed the landscape of financial investments, with the constant fluctuations that have happened since the beginning of the bitcoin journey in 2009, which has only escalated in recent years. With bitcoin gaining more popularity each year, the task of predicting becomes crucial for decision making by many investors. This study tries to forecast the bitcoin prices, addressing the challenges posed by the dynamic and often unpredictable nature of the cryptocurrency market. So to keep that into consideration, we haven't used any data prior to the year of 2020.

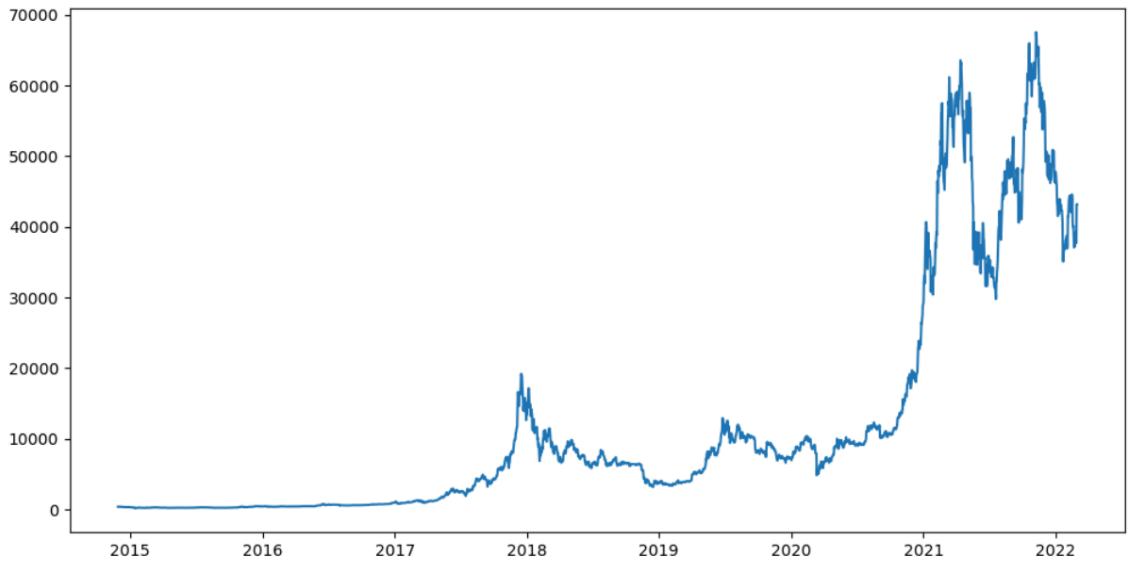


Fig 2 - the price based on time from 2015 until around 2022

The goal is to use advanced machine learning models to recognize patterns and relationships within the dataset, ultimately predicting (1) whether Bitcoin prices will rise or fall and which is easier for traders to make decisions and follow. (2) what will the price of bitcoin exactly be in the future.
The dataset we are going to use overall contains some important detail in the price of bitcoin each day throughout the years.

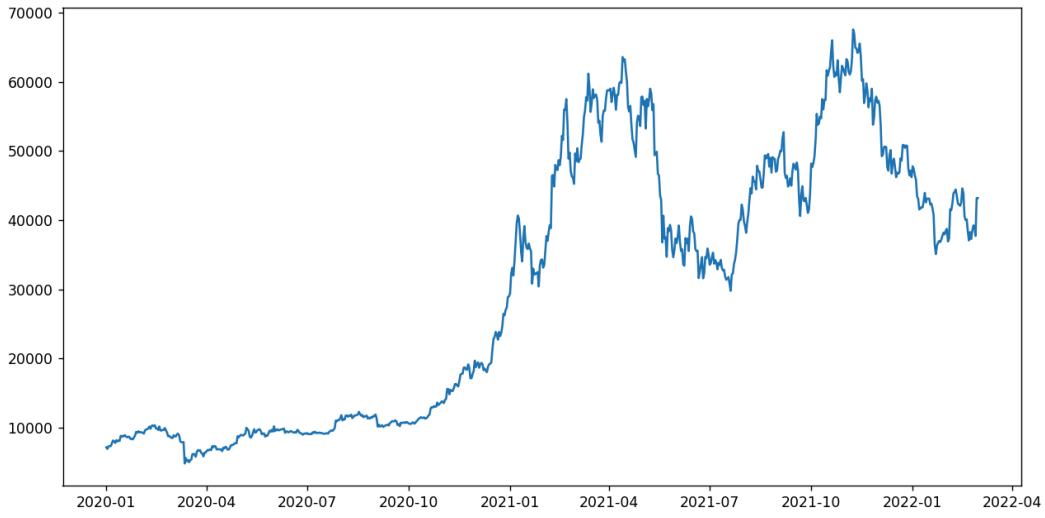


Fig 3 - the portion of data that we have used.

2. Related work/Background

This topic has actually been rather popular to work on. So there are many papers and notebooks available

(1) The paper "Predicting Bitcoin Prices Using Machine Learning" compares logistic regression, support vector machines, and random forest algorithms. Analyzing a dataset encompassing various economic variables and spanning from December 2014 to July 2019, the study concludes that the traditional logistic regression model outperforms machine-learning techniques, achieving a 66% accuracy. Findings suggest that Bitcoin returns are independent of other cryptocurrencies and macroeconomic factors, positioning Bitcoin as a unique asset for investors seeking a hedge against regulatory frameworks and inflation. (so, as said here, we only used historical bitcoin data and didn't track other factors in our prediction)
The study contributes to understanding Bitcoin's dynamics and emphasizes the significance of traditional modeling approaches in predicting its price movements.

(2) The paper "Bitcoin Price Prediction: A Machine Learning Sample Dimension Approach" aims to predict Bitcoin prices using various machine learning techniques, addressing the need for accurate predictions due to the cryptocurrency's high volatility. The study categorizes Bitcoin prices into daily and high-frequency (5-minute interval) predictions, employing high-dimensional features for daily prices and fundamental trading features for 5-minute interval prices. Logistic Regression is found to predict daily prices with 64.84% accuracy, while XGBoost achieves 59.4% accuracy for 5-minute interval prices. The research highlights the importance of sample dimensions in machine learning algorithms and suggests that prudently selected high-dimensional features compensate for model simplicity in Bitcoin's daily price prediction. The findings contribute to understanding the dynamics of Bitcoin prices, emphasizing the significance of feature selection in accurate predictions.

3. Proposed method

The purpose of this paper is to predict future Bitcoin prices, and also predict the rise and fall of the next day, so we divide our models into two sections, classification models and regression models to predict the exact price.

3.1. classification models

These classification models are trained on features such as:

- Open-close: the difference between opening and closing prices on one day.
- Low-high: the difference between daily low and high prices
- is_quarter_end: indicating if it's the end of a quarter.

The target variable 'target' is binary, denoting whether the closing price will increase (1) or decrease (0) on the next day.

3.1.1. Logistic Regression is a linear model that uses the logistic function to model binary outcomes. It estimates the probability that a given instance belongs to a particular category. It calculates a weighted sum of input features and applies a logistic function to produce a probability score. A threshold is set, and if the probability surpasses this threshold, the model

predicts a positive outcome (increase in closing price); otherwise, it predicts a negative outcome (decrease).

3.1.2. Decision Tree is a tree-like model where each node represents a decision based on a feature, leading to subsequent nodes (branches) until a final decision is reached at the leaves. The tree recursively splits the dataset based on features, choosing the feature that best separates the data at each step. It predicts the target variable by traversing the tree from the root to a leaf, where the leaf's decision represents the model's prediction.

3.1.3. KNN is a non-parametric model that classifies an instance based on the majority class of its k-nearest neighbors in the feature space. For a given instance, the model identifies its k-nearest neighbors based on a distance metric. The majority class among these neighbors determines the prediction for the instance. KNN does not explicitly learn a model; instead, it classifies new data points based on their proximity to existing data.

For all three models, the training accuracy is determined by calculating the Area Under the Receiver Operating Characteristic (ROC) curve (AUC-ROC) score on the training set, measuring the models' ability to distinguish positive and negative instances within the data used for training. Similarly, the validation accuracy assesses the generalization capability of the models on a separate validation set. A higher AUC-ROC score (ranging from 0 to 1) signifies better predictive performance, with 1 indicating perfect discrimination.

3.2. Regression Models (Predictive)

In this section, we are aiming to exactly predict the closing value of bitcoin, we implemented two different models, one with a neural network and another model named XGBoost. For both of them, we particularly splitted the train and test sections as shown in the graph below.
Even though we could've used train-test split functions, for the sake of comparing the outputs, we sat the portions by hand.

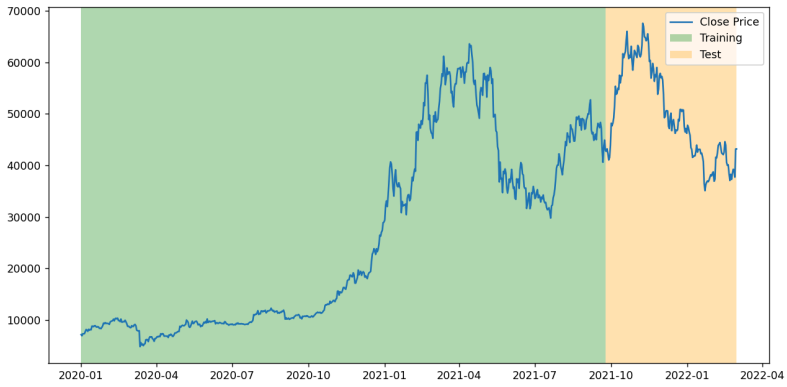


Fig 4 - train-test splitting. Green part being train and yellow part being test.

3.2.1. Neural network

The model is designed for a time series forecasting task using a Long Short-Term Memory (LSTM) neural network.

Data Preparation

Data Selection: as said before, we selected the 'close' price from the dataset starting from January 1, 2020. Since we're interested in predicting future 'close' prices based on historical data.

Normalization**:** The data was normalized using MinMaxScaler to scale the 'close' prices to a range of [0, 1]. Which is known to help in speeding up the training process and improving model convergence.

Sequence Creation: Then we transformed the data into sequences of 10 time steps. Each sequence is used to predict the 'close' price at the next time step. We chose the sequence length of 10 but it is an arbitrary choice.

Model Architecture

LSTM Layer: The model starts with an LSTM layer with 10 units. LSTMs are a type of recurrent neural network (RNN) capable of learning long-term dependencies. Again, The choice of 10 units is a model hyperparameter and can be changed. The ReLU activation function is used, which allows the model to learn non-linear relationships.

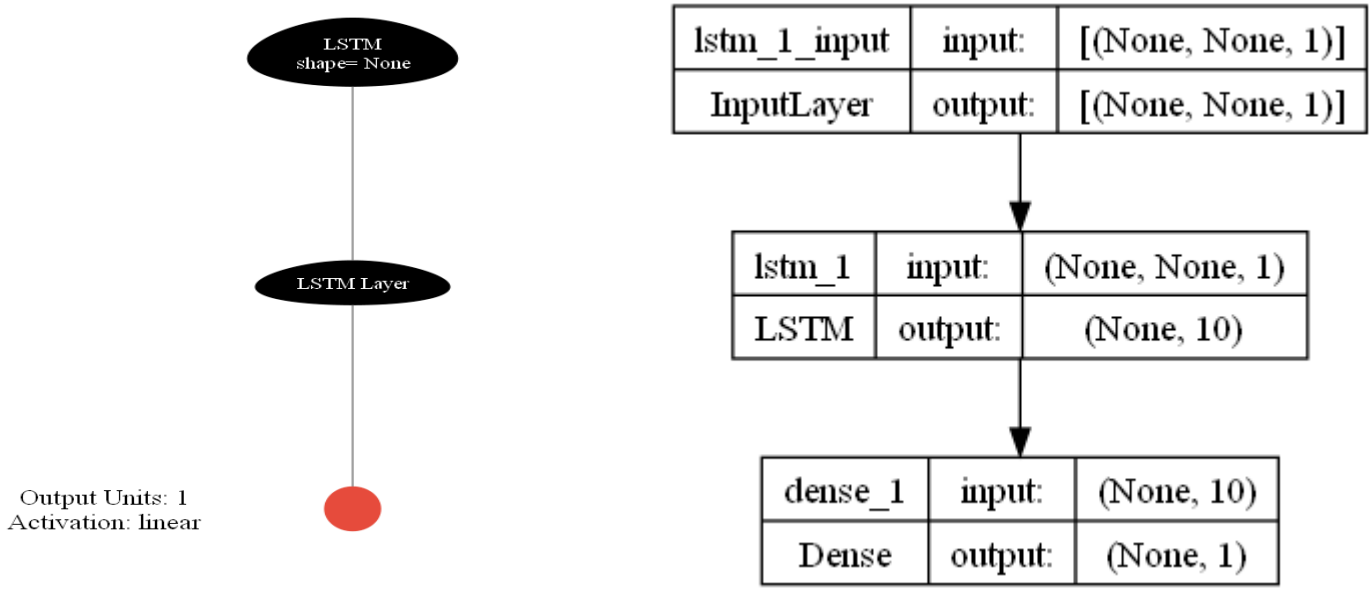
Dense Layer: Following the LSTM layer, there's a Dense layer with a single unit. This layer is responsible for producing the final prediction. Since it has only one unit, the output is a single scalar value, which corresponds to the predicted 'close' price at the next time step.

Compilation

Optimizer: Adam optimizer is used, which is an adaptive learning rate optimization algorithm designed specifically for training deep neural networks.

Training Data and Test Data

The data is split into training and test sets, with exactly the first 632 data points going to the training set and the rest to the test set. Both training and test sets are reshaped to fit the model's expected input shape, which is (samples, time steps, features). Here, each sample is a sequence of 10 time steps with 1 feature (the 'close' price).



3.2.2. XGBoost

XGBoost is a model that was introduced in 2014 and then caught the Machine Learning world by fire. It is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework (But it optimizes the gradient boost fig-5).

In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured data, Decision tree based algorithms are doing a good job for now. And Regardless of the type of prediction task at hand; regression, or classification. XGBoost is well known to provide better solutions than other machine learning algorithms.

To fully understand the structure of the XGboost algorithm, it is necessary to know how Ada boost, gradient boost, and regularization works, so the whole explanation cannot be put into this paper. For now, it's enough to now that XGBoost combines the predictions of multiple weak learners (decision trees) to form a strong learner. This helps in reducing overfitting and improving generalization performance.

In the model we implemented, which counts as supervised learning, the feature matrix (X) excludes the 'date' and 'close' columns and includes all the other columns, while the target variable (y) is set as the 'close' column. The XGBoost regression model is instantiated with specified hyperparameters, including the number of trees (n_estimators), learning rate of 0.2 (that can be changed), and maximum depth of each tree being 5. The accuracy metric is calculated (which might not be the best indicator for the model's predictive performance). To have a more comprehensive evaluation of the model's effectiveness in predicting future Bitcoin prices, the Root Mean Squared Error (RMSE) is computed to quantify the overall prediction error between the predicted and actual closing prices. And also MAE (Mean Absolute Error).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

'n' : number of observations in the dataset.
y_i : the actual value of the target variable for observation i.
$\hat{y}_i$: the predicted value of the target variable for observation i.

4. Results

Our dataset was a bitcoin historical price dataset, originally consisting of 2651 rows and 9 columns. Each row contains information about the bitcoin price, such as opening price, which is the first price registered for the day, then highest price, lowest price, and closing price. Also the Volume BTC and Volume USD. It contained the data from 2014-11-28 until 2022-03-01.

The classification models we tested with two different parts of the dataset, first, they used the whole dataset (the data from 2014 until 2022) and the second time, only the data after 2020 (like the regression models).

	Training Accuracy (2014 - 2022)	Validation Accuracy (2014 - 2022)	Training Accuracy (2020 - 2022)	Validation Accuracy (2020 - 2022)
Decision Tree	100%	96.97%	100.00%	99.21%
Logistic Regression	99.74%	99.45%	99.96%	99.93%
K-Neighbors	99.89%	98.99%	99.91%	99.82%

Overall, the models performed exceptionally well in both time periods. The output on the decision tree however, might suggest overfitting.

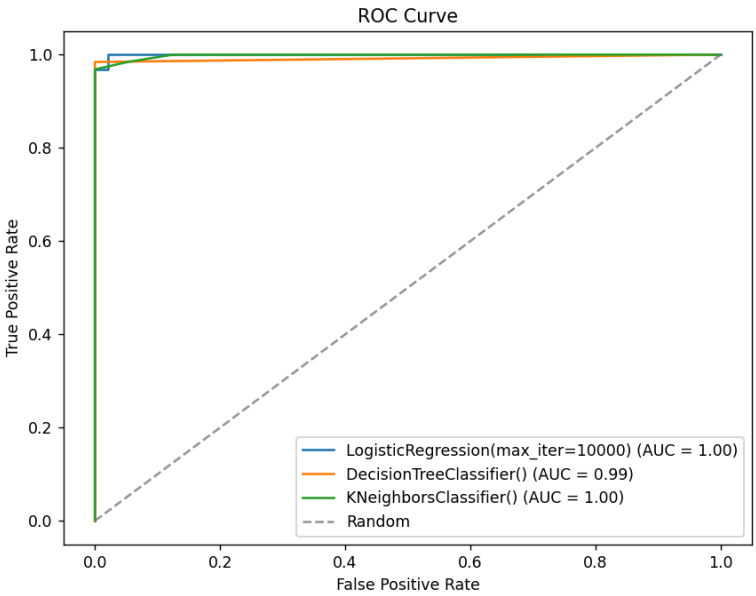


Fig 6 -
The ROC curve visually captures how well the classification models worked. (this is 2020 - 2022 data) .

For predicting the exact price, XGBoost actually did a good job, with the RMSE being 1304.8541, considering the range of closing prices (as we can see in previously shown figures). And MAE being 1009.5633. We can see that these two parameters are close in amount which suggests that the errors are somewhat evenly distributed and not skewed by a few large errors. And there might not be a significant presence of very large errors that would disproportionately impact the RMSE, since this metric gives more weight to larger errors because it involves taking the square root of the average of squared errors.

LSTM output was impressive as shown below with the graphs. It already works really well on the train set, but its performance on the test set showcases how good it generalizes on unseen data, and is also an indicator against overfitting.

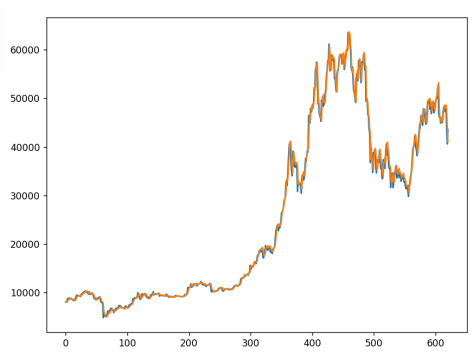


Fig 7- train set prediction vs actual data

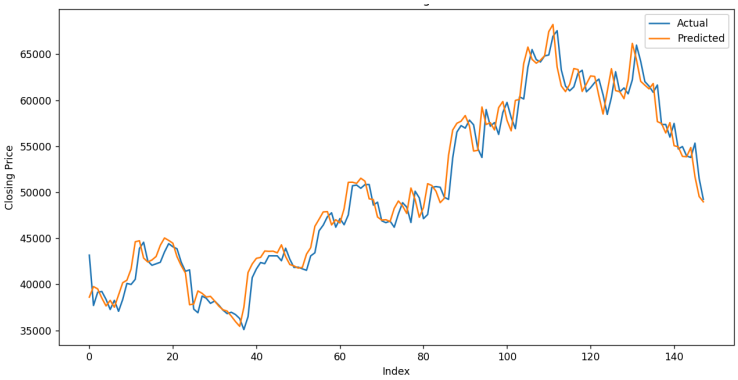


fig 8- test set prediction vs actual data

This success of the LSTM model in predicting Bitcoin prices for both the training and testing sets can be attributed to its adeptness in capturing temporal patterns and long-term dependencies inherent in financial time series data.

5. Discussion

Considering how good the classification models worked, we can come to the conclusion that with both limited training set data and broader data, the bitcoin traders won't be having such a difficult job predicting the direction of the bitcoin price. However, It's worth noting that the models trained on the data from 2020 to 2022 achieved slightly higher accuracy, possibly due to the models adapting to more recent market dynamics and trends.

In the regression section, we had both XGBoost and LSTM predict the price from October 2021 until March 2022 (as the test set) so their output is comparable.
As we can see in the figures below, LSTM worked better than XGBoost in the comparison graph because LSTM is good at understanding patterns over time, especially in financial data like bitcoin prices. In very simple words, LSTM is better at connecting the dots in a sequence, while XGBoost is good with other types of data but might miss some of the time-related patterns. LSTM's special structure helps it catch the ups and downs in the data, making it great for predicting prices. So, in the graph, LSTM outperformed XGBoost because it's more flexible and skilled at handling the complexities of predicting prices over time.

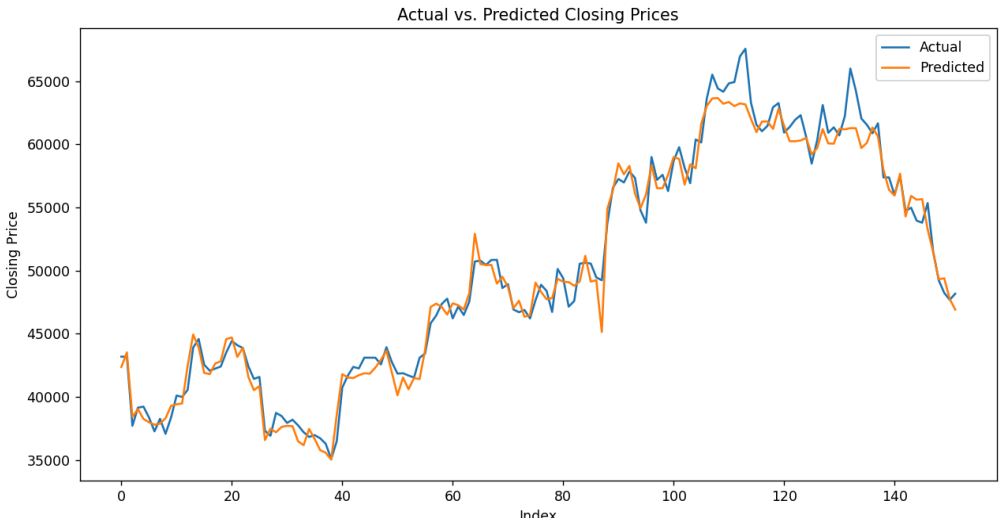


Fig 9 - XGboost model

Fig 10 - LSTM model

We only used these two models but there are other models which are popular for the task of price prediction, however, we chose not to use them for various reasons. For example **ARIMA** (AutoRegressive Integrated Moving Average) is a popular time series forecasting model. However, it might struggle with the non-linear and complex patterns often observed in cryptocurrency price data and is considered to be better at capturing linear trends.
And **Random Forests** which is model similar to XGBoost in the sense of building multiple decision trees and combines their predictions but this model may not work as well as XGBoost in our task and could be computationally expensive.

6. References

[1] XGBoost Algorithm: Long May She Reign!
https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
[2] Bitcoin Price Prediction: A Machine Learning Sample Dimension Approachhttps://www.researchgate.net/publication/360096794_Bitcoin_Price_Prediction_A_Machine_Learning_Sample_Dimension_Approach
[3] Predicting Bitcoin Prices Using Machine Learning
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10216962/