



ALGORITHMIC TRADING MODEL DEVELOPMENT FOR BTC/USDT CRYPTO MARKET

PREPARED FOR

Kharagpur Data Science Hackathon 2024

PREPARED BY

Zaid Ahmed Khan, Akash Kaushik, Shishir Haldia, Bhanupriya Gupta

Team Spectacular

1. Introduction


For years, algorithmic trading has grown in financial markets, including cryptocurrency. Crypto market volatility and 24/7 nature offer lucrative opportunities for algorithmic trading strategies. Unlike traditional assets, modelling and optimising cryptocurrency algorithmic trading systems is difficult.

This report details our approach to developing algorithmic trading models tailored to the Bitcoin (BTC)/Tether (USDT) cryptocurrency market.

The problem statement for this competition was to create profitable algorithmic trading models for the BTC/USDT market that can consistently outperform benchmark returns, manage risk, and capitalise on the dynamics of this volatile market. Combining cryptocurrency expertise with statistical modelling, machine learning, and programming skills is crucial. The increased volatility requires effective risk management in the models.

This undertaking highlighted the complexity of accurately modeling and trading Bitcoin compared to traditional assets. Our team relished this challenge to unlock alpha opportunities in the growing cryptocurrency space with data-driven and rigorously backtested algorithmic trading systems. The following sections detail our methodology, results, and insights.

2. Data Acquisition and Preprocessing

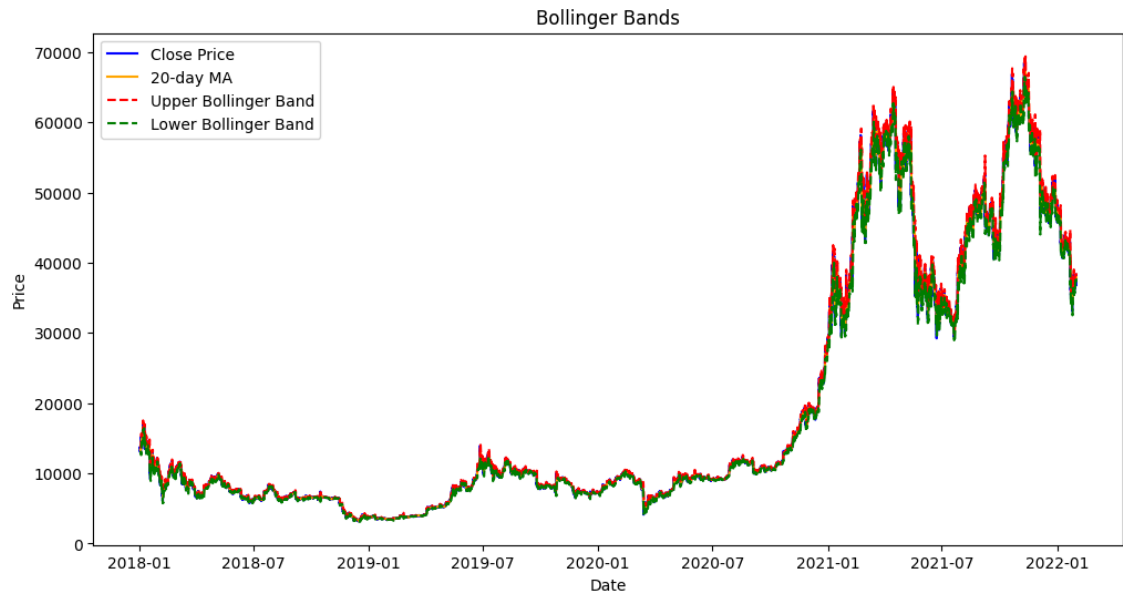
- Data Acquisition:  KDSH Data
- Data Preprocessing: The data obtained was quite clean and contained no null values.

3. Model Design

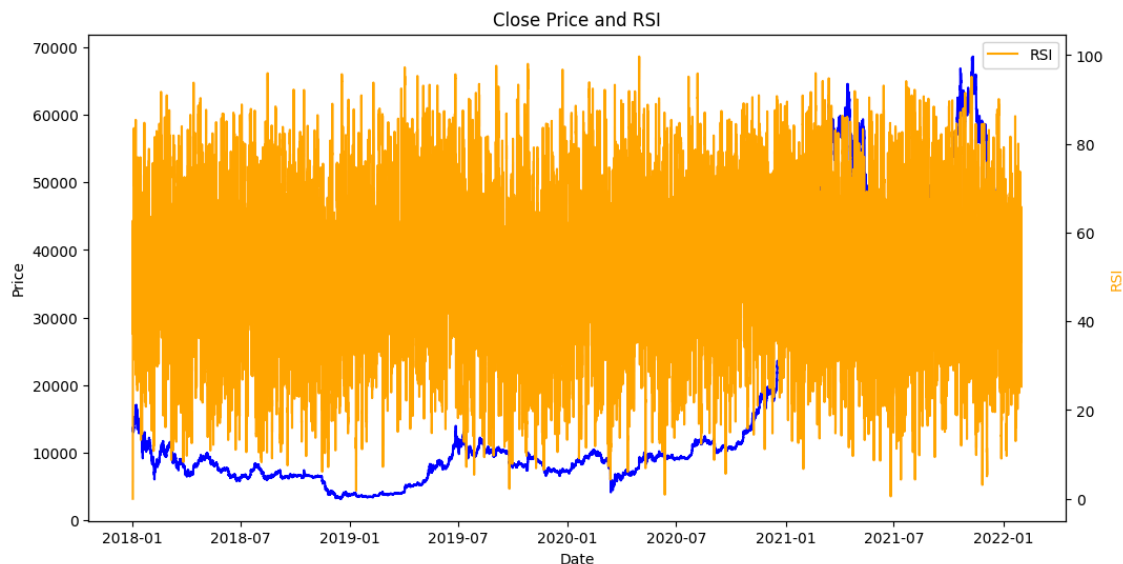
We have created a trading algorithm based on a Multi-Layer Perceptron (MLP) as a classifier with three classifications, Buy, Hold and Sell. We created a full standard analytic pipeline, which included gathering price and volume time-series data, preprocessing it with feature extraction and labelling, and then training and testing the machine learning model. The dataset labelling method, which is based on two thresholds to intercept important market changes and two temporal windows—one in the past and one in the future—for the purpose of projecting price trends. We have used a LSTM model for predicting future stocks prizes . Then we have combined both the indicators data and LSTM data using MLP classifier to predict whether to Buy, Sell or Hold and we have monte carlo simulation to predict stable guesses.

The following technical indicators are used -

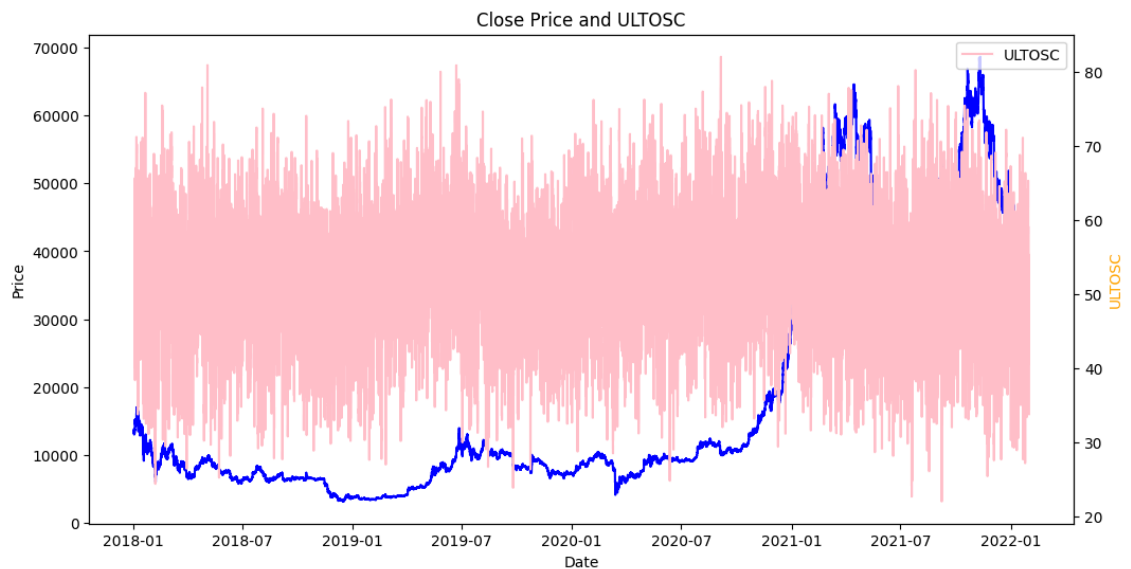
- Bollinger bands. These are used to check whether prices are high or low on a relative basis. Given n periods, the price calculated using the n -period moving average is used as a reference price. Two lines are plotted above and below one standard deviation away from the reference price. The standard deviation of the price in the previous n -periods is used as a measure of volatility.



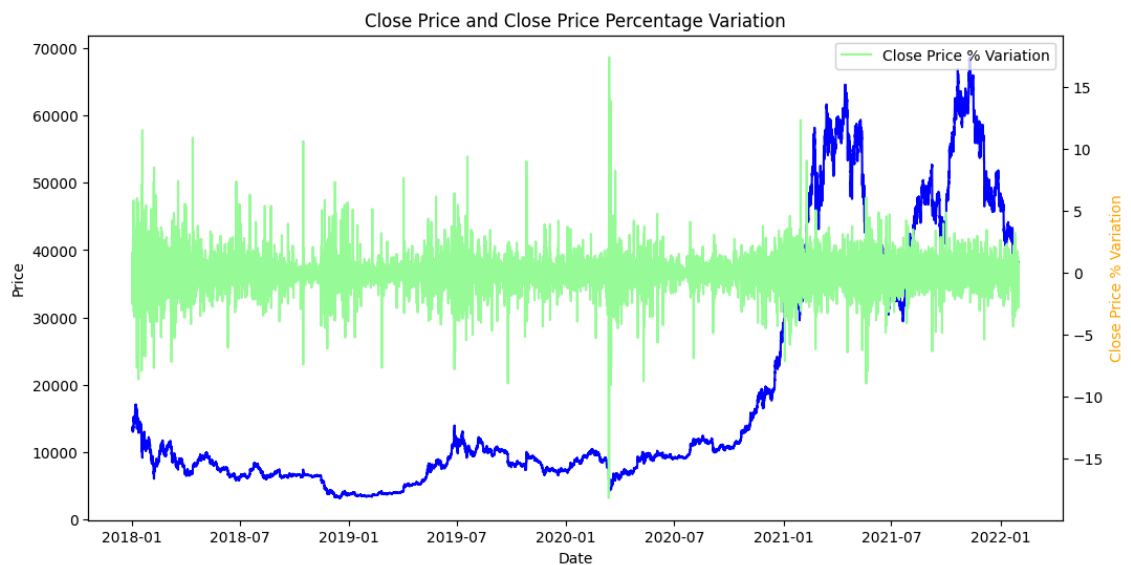
- RSI. The Relative Strength Index is a momentum indicator that measures the magnitude of recent price changes in order to evaluate overbought and oversold conditions. A lower RSI value indicates that the asset is oversold, while a higher value means that the asset is overbought.



- ULTOSC. The Ultimate Oscillator uses the values of three different moving averages with multiple time periods (or cycles), to identify overbought and oversold conditions in the market, thus improving the accuracy of the signals generated by the indicator.

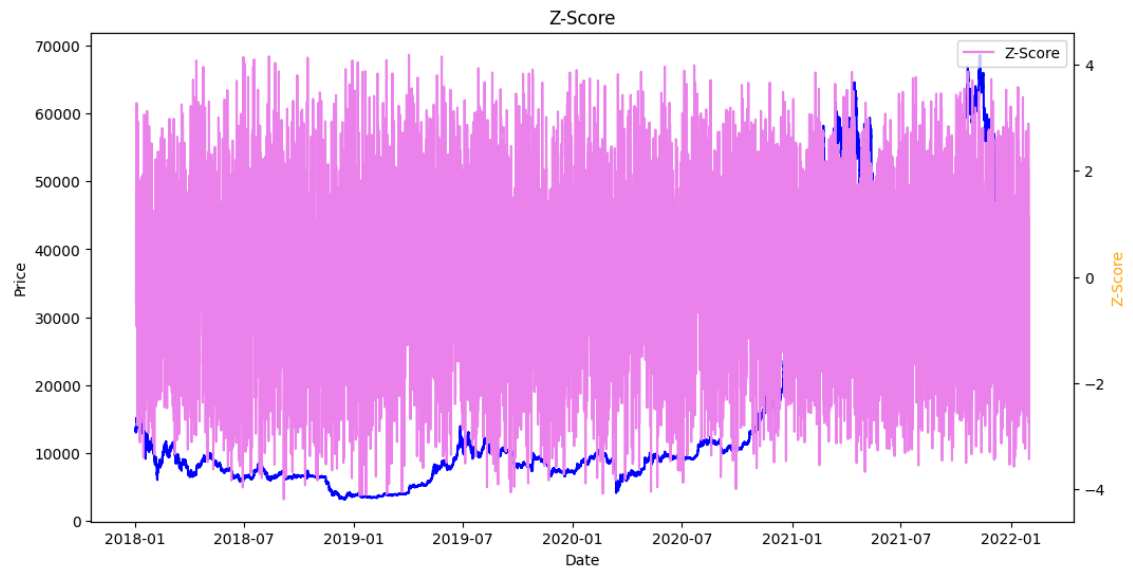


- Close price percentage variation. This measures the percentage difference between the current price and the previous close price.

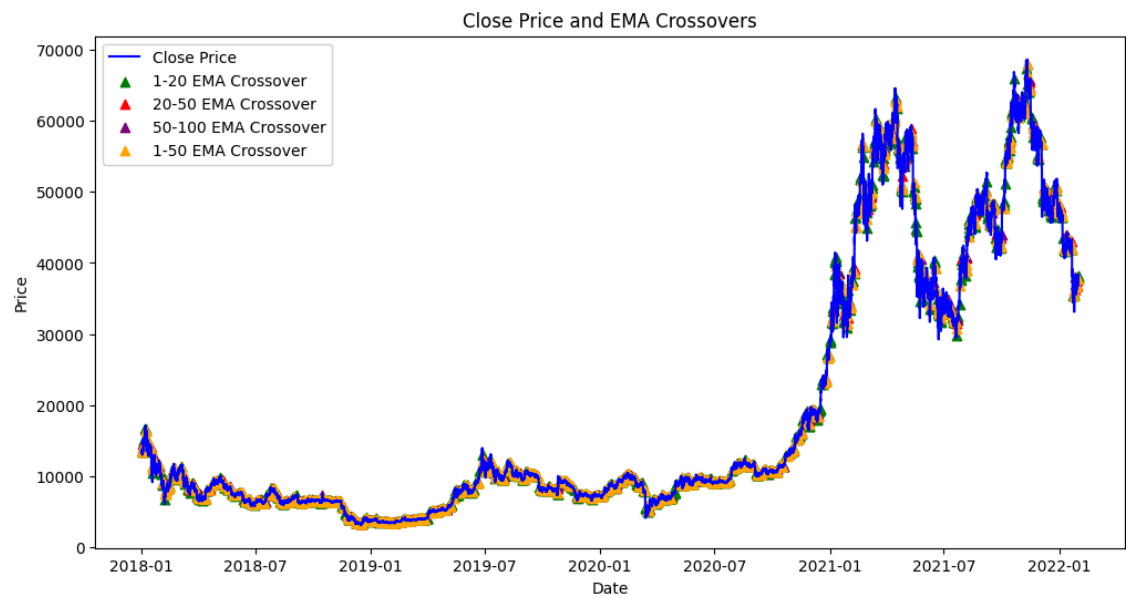


- Z-Score. This uses the z-score of the close price in a given number of time frames. In our implementation we used 30 close price past samples to compute the Z-Score of

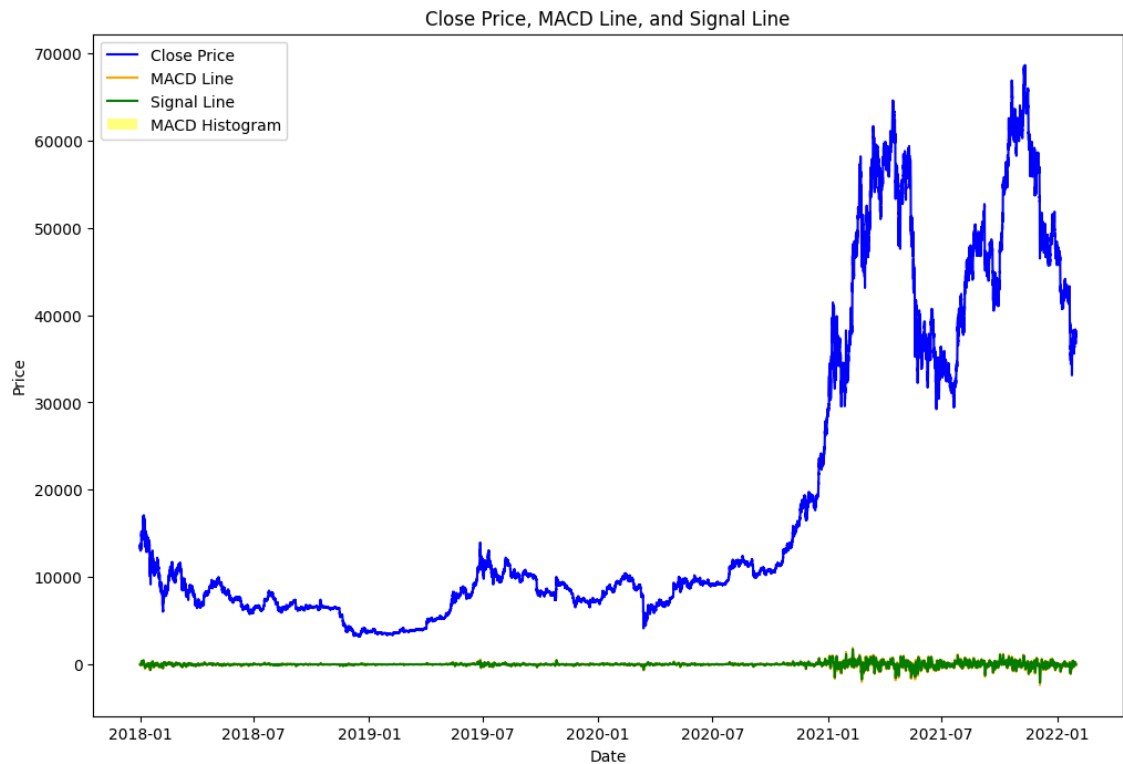
the actual time frame.



- EMA crossovers. The Exponential Moving Average crossovers are an established source of information for trend following and inversion. 4 EMA crossovers are based on 1 and 20, 20 and 50, 50 and 100, and 1 and 50 periods.



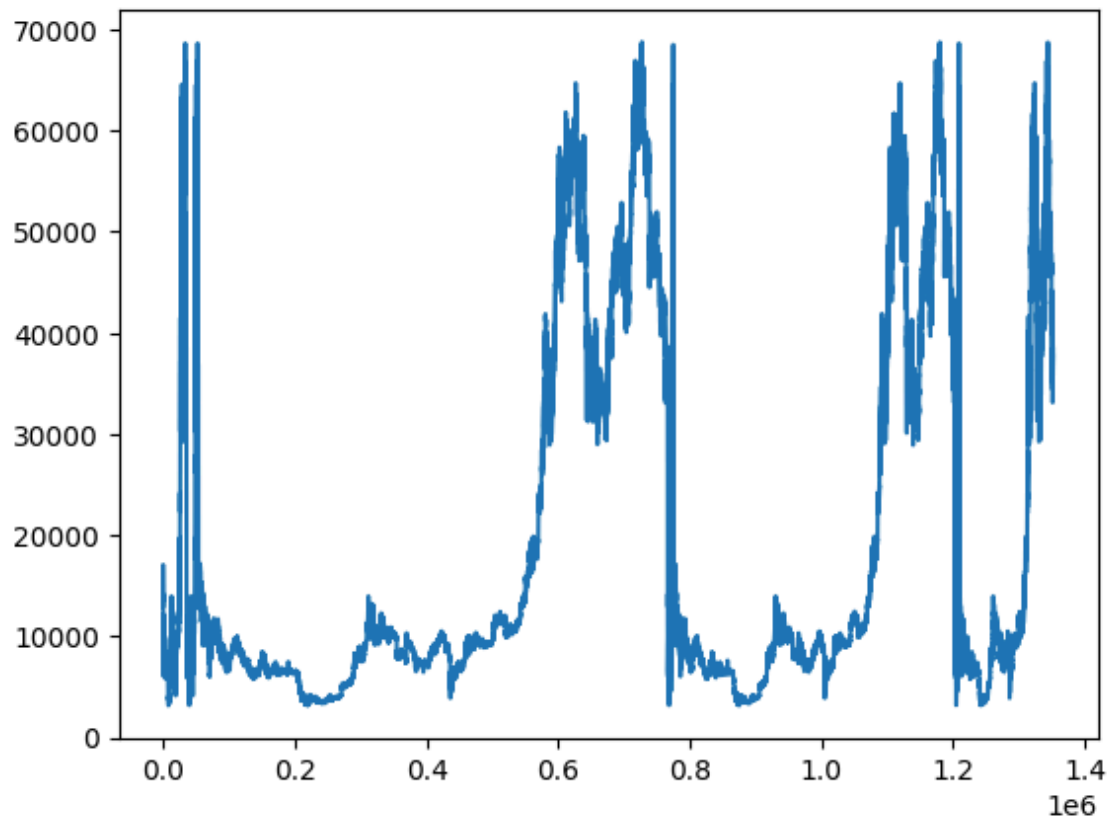
- MACD- The MACD indicator measures the relationship between two EMAs and can be employed alongside the EMA to confirm trend direction and strength.



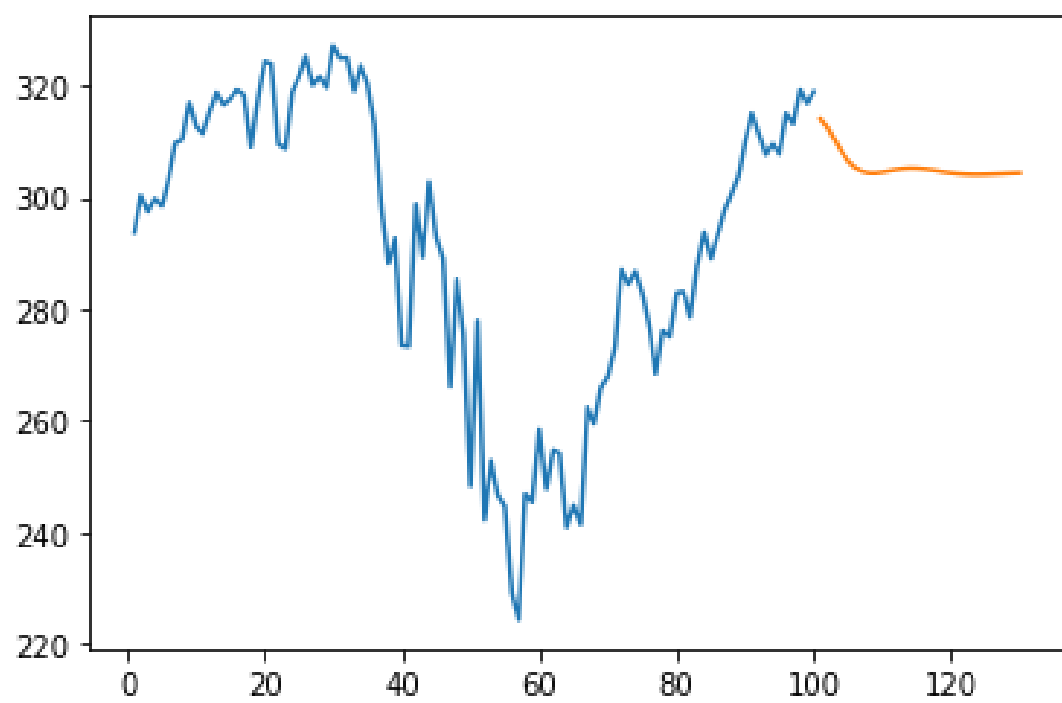
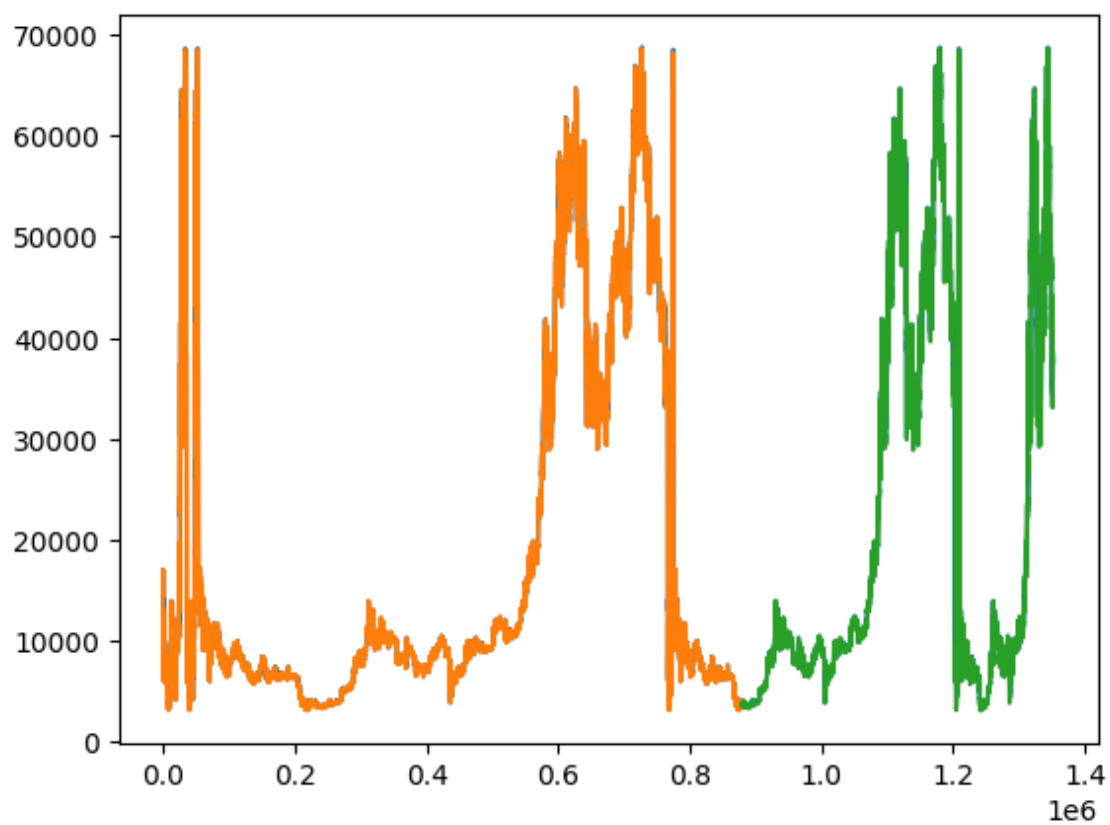
Using three labels addresses potential ambiguities that may occur when deciding whether to open or close a position, which should improve classification accuracy compared to using only two labels. We attempt to minimise any ambiguity that might occur when determining whether to hold off on initiating a buy position or whether to hold off on initiating additional buy orders once the position has been opened by taking the Hold label into account.

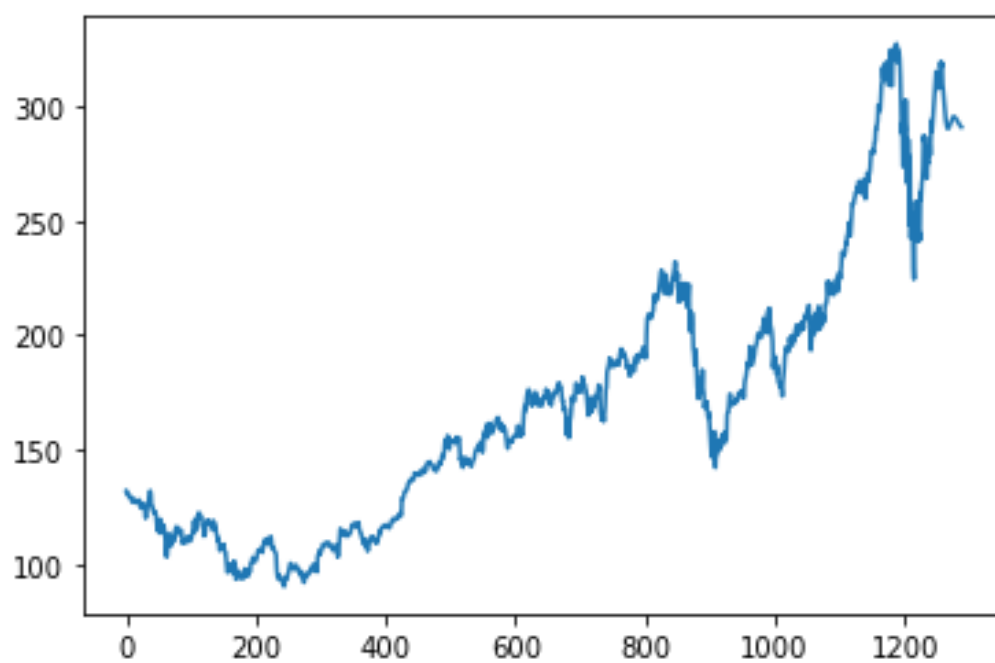
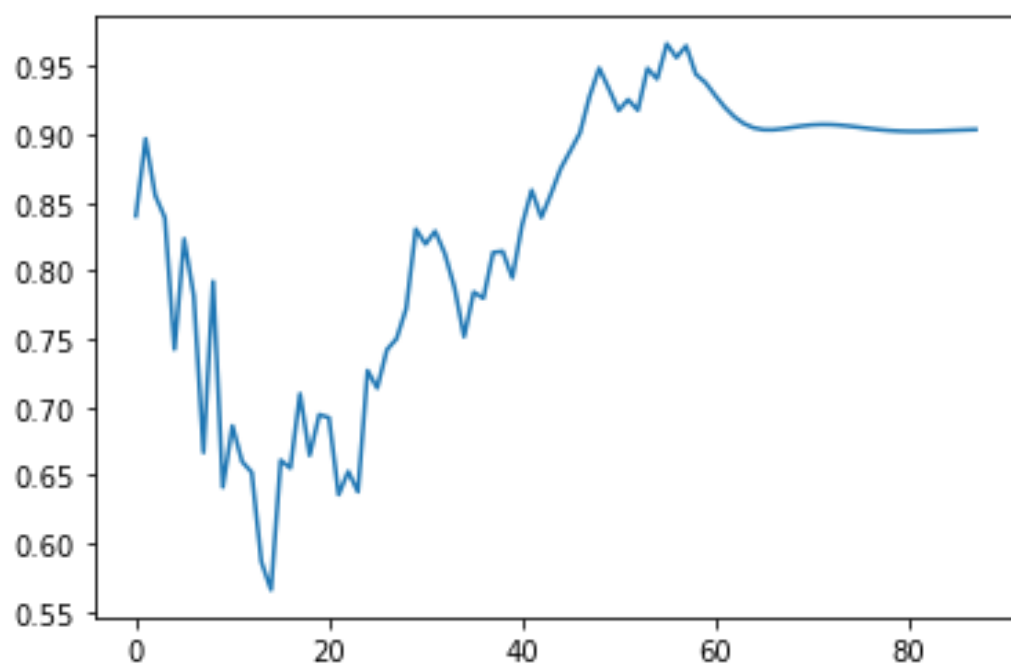
LSTM Time Series Forecasting

The data is normalized using MinMaxScaler from sklearn. Following are the stock predictions result that we got using this model plotted using matplotlib.



The model architecture consists of three LSTM layers followed by a Dense layer. The model is compiled with the mean squared error loss function and the Adam optimizer.

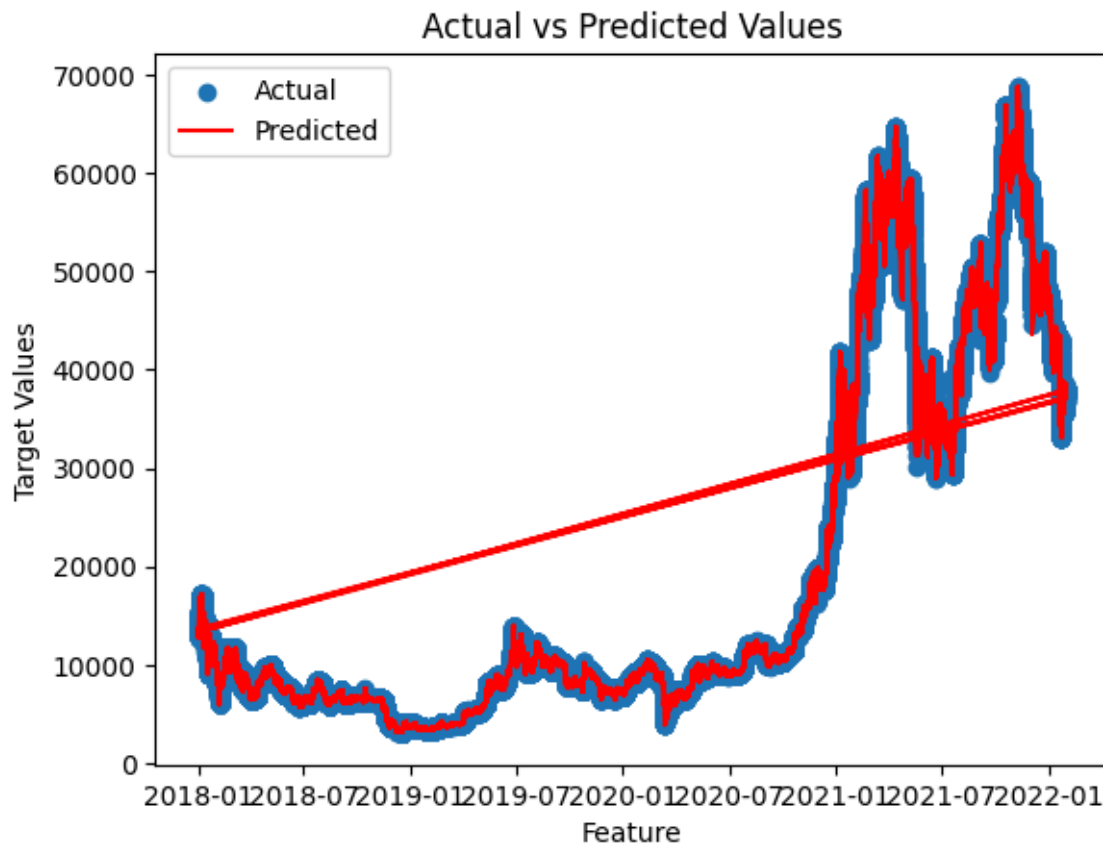




Linear Regression Model

We also tried for Linear Regression model for doing the task of LSTM model.

StandardScaler is applied to the feature variables. A linear regression model is instantiated and trained using the training data.



Monte Carlo Simulation

We used Monte Carlo simulation to more stable predictions by simulating our strategy more time. This strengthens the process of evaluating strategies by adding another layer.

4. Backtesting

Backtesting involves evaluating the strategy's performance on historical data. The calculated metrics contribute to the overall understanding of the strategy's effectiveness.

Mean Net Profit: 132485.48

Sharpe Ratio: 1.93

Sortino Ratio : 2.96

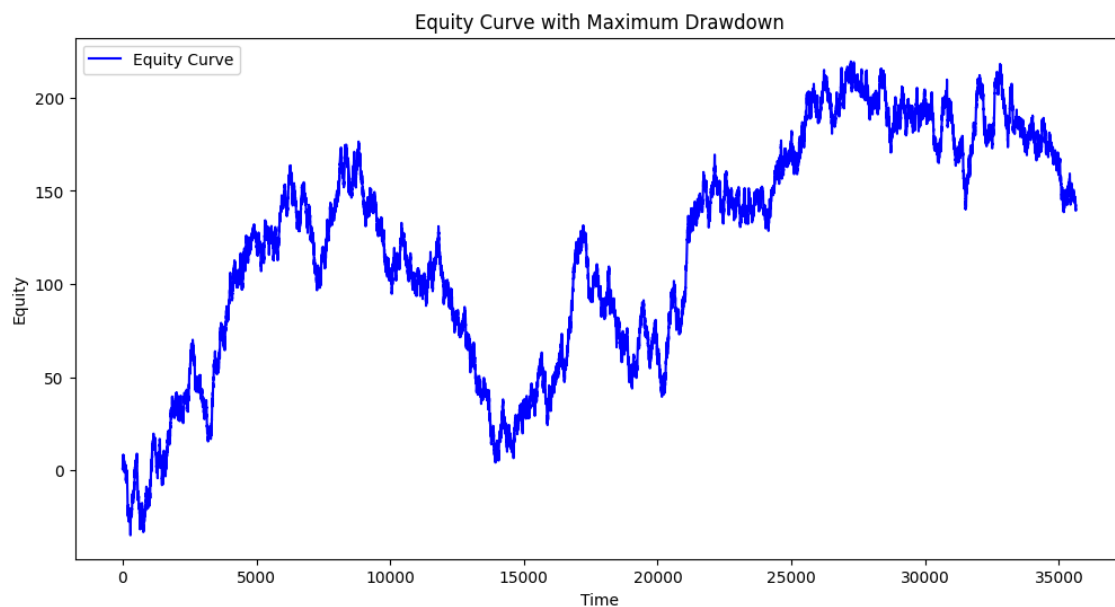
Win Rate (Profitability %) : 84.01616343023908

Maximum Drawdown : 0.08946176345812168

Risk-Reward Ratio: 1.9484456417265248

Net Profit Exceeding Benchmark: 255632.80754503317

The plotted equity curve with Maximum Drawdown provides a visual representation of the strategy's performance over time.



5. Risk Management

Introduction

We propose a risk management strategy for algorithmic trading using a Multi-Layer Perceptron (MLP) model. The MLP classifies various trading instances into buy, sell, or hold actions. A unique feature of our approach is the introduction of a new column representing a stop-loss value whenever a buy action is initiated.

Stop Loss Strategy

A stop-loss order is designed to limit an investor's loss on a position in a security. It is a crucial component of risk management in trading. The idea is to set a stop loss level for each trade that will close the trade if the price changes unfavorably by a specified amount.

Several strategies can be used to determine the best stop-loss level based on the current price and past trends. Some traders use a fixed percentage, while others use technical indicators such as the Average True Range (ATR) or moving averages.

One common method is the trailing stop loss, which adjusts the stop loss level as the price changes, locking in profits as the price moves in a favorable direction. Another approach is to set the stop loss at a level that, if reached, would change your opinion about the direction of the market.

Machine Learning Model for Stop Loss Prediction

To predict the best stop loss level, we can use machine learning models. The choice of model depends on the characteristics of the data and the specific requirements of the task.

For instance, regression models such as Linear Regression, Decision Trees, Random Forest, and Gradient Boosting can be used for numerical prediction. These models can handle non-linear relationships and are less prone to overfitting.

However, as per our observations, ours is the best stop-loss method based on the data provided and after testing our algorithm against different stop-loss strategies and evaluating the performance, ensuring a robust testing method where results aren't being curve-fitted.

Conclusion

In conclusion, our proposed risk management strategy for algorithmic trading combines the predictive power of an MLP model with a dynamic stop-loss mechanism. This approach aims to balance the pursuit of profit with the need to limit potential losses, providing a robust framework for algorithmic trading.

6. Deliverables

Code: Included in the zip file

Presentation: link, Summarize the model, its rationale, backtesting results, and risk management approach tailored to the BTC/USDT market. Included in the zip file.

7. Other Approaches Employed

1. **Dilated Convolutional Neural Network (CNN) sequence-to-sequence (seq2seq) model** for time series forecasting, specifically for predicting cryptocurrency market returns in the BTC/USDT market.

The model includes an encoder with dilated convolutional layers and a self-attention mechanism. The model also has a forecasting layer for predicting the next value in the time series. The training loop uses gradient tape and an Adam optimizer to minimise the mean squared error loss.

But we discarded that model because it was overfitting the training dataset.

2. In place of MLP Classifier we also tried Logit Classifier and XG Boost but we were getting better scores for MLP Classifier , So we discarded them.