PROJECT 2

ALGO-TRADING

THOMAS GOSPER
AMY BOOKER
PARVINDER RAKHRA

HYPOTHESIS

We are interested in stock market trading algorithms.

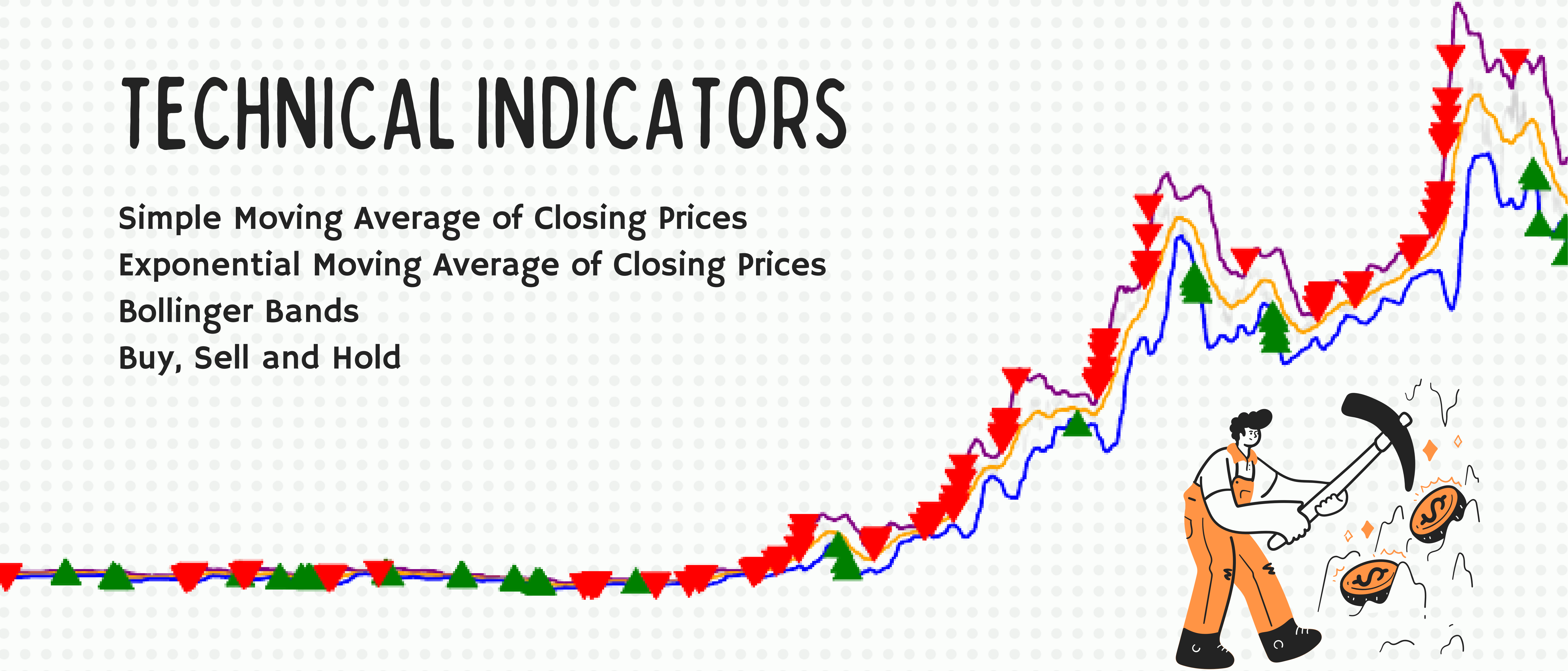
We wanted to create a trading algorithm, using Tesla (TSLA) stock as our baseline.

We created predictive models using 3 technical indicators over 5 years of TSLA stock market data.

We then evaluated the performance of the models to determine which had the best predictive power to maximise potential portfolio value.

TECHNICAL INDICATORS

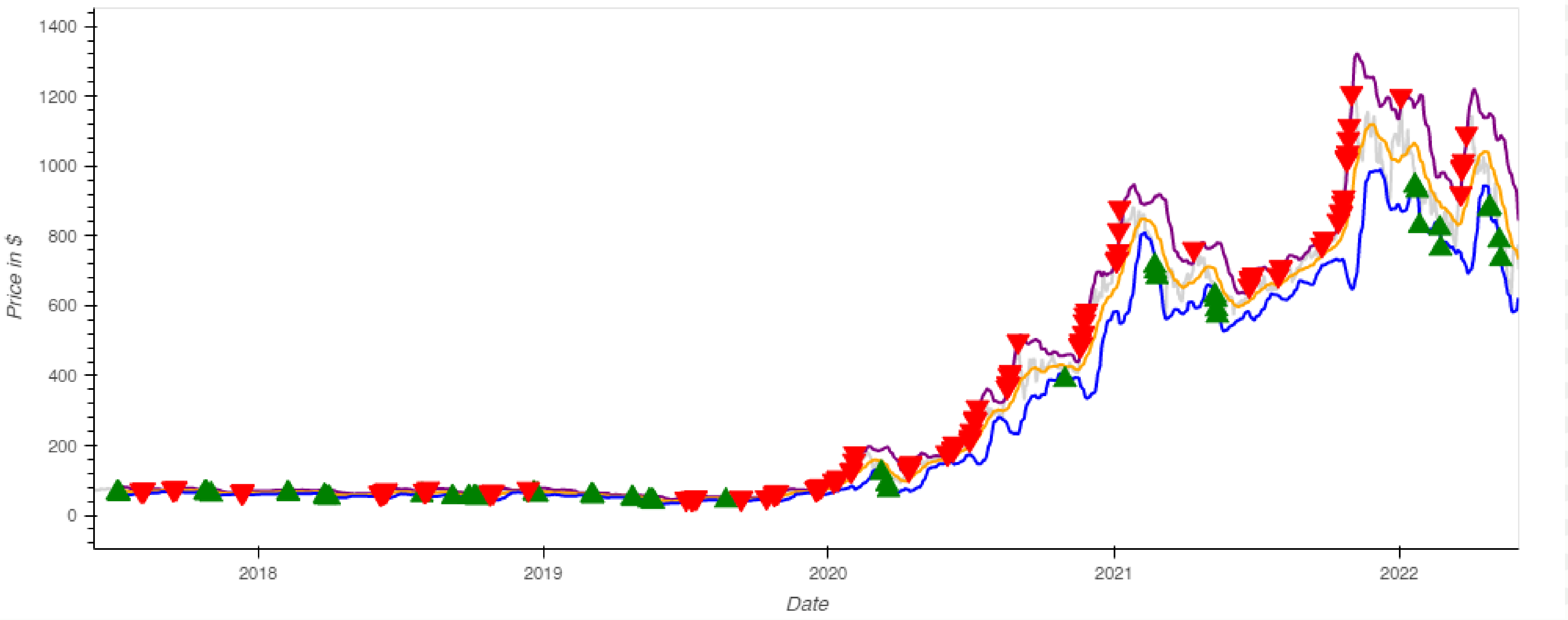
Simple Moving Average of Closing Prices
Exponential Moving Average of Closing Prices
Bollinger Bands
Buy, Sell and Hold



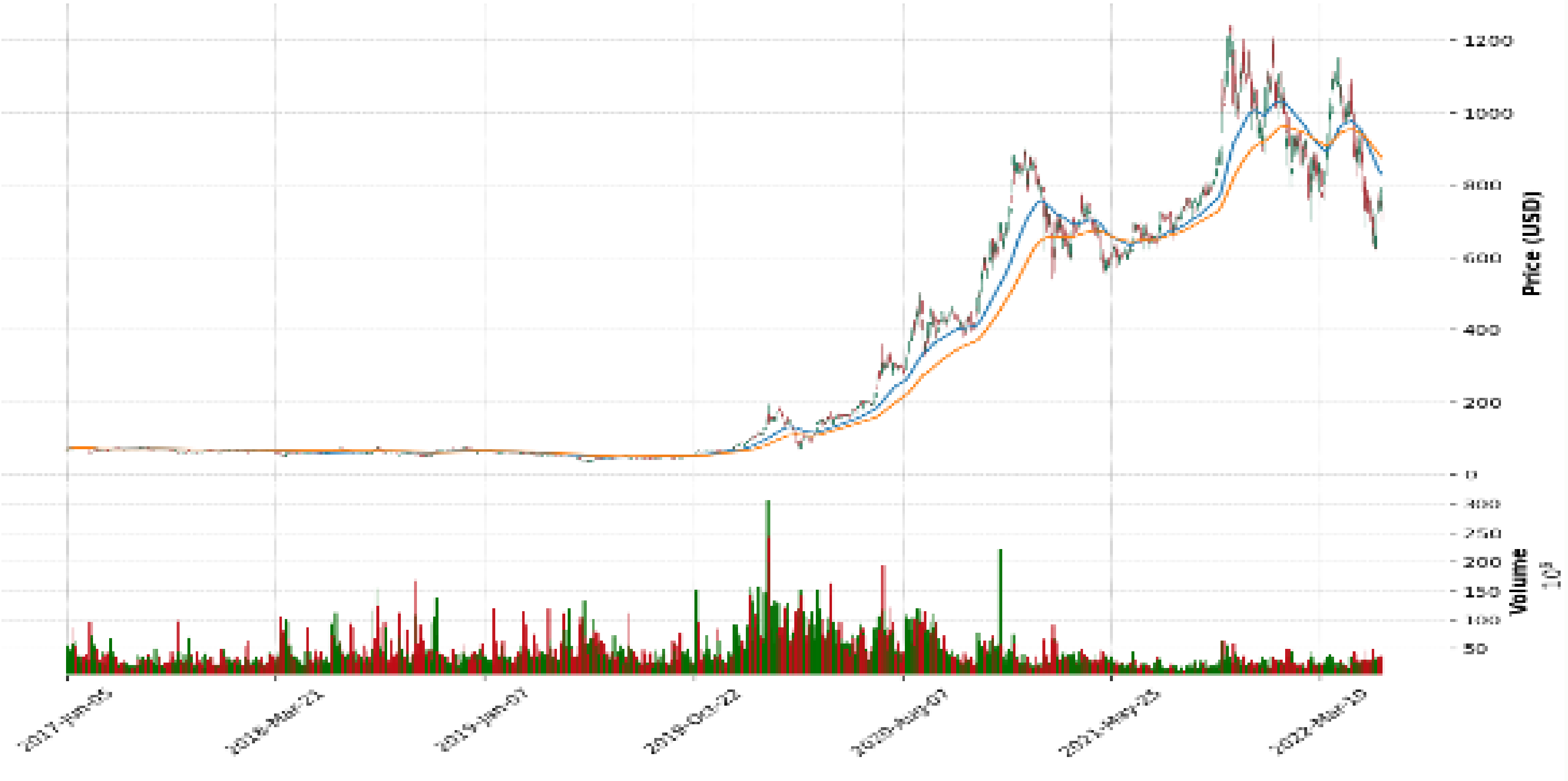
MPL FINANCE

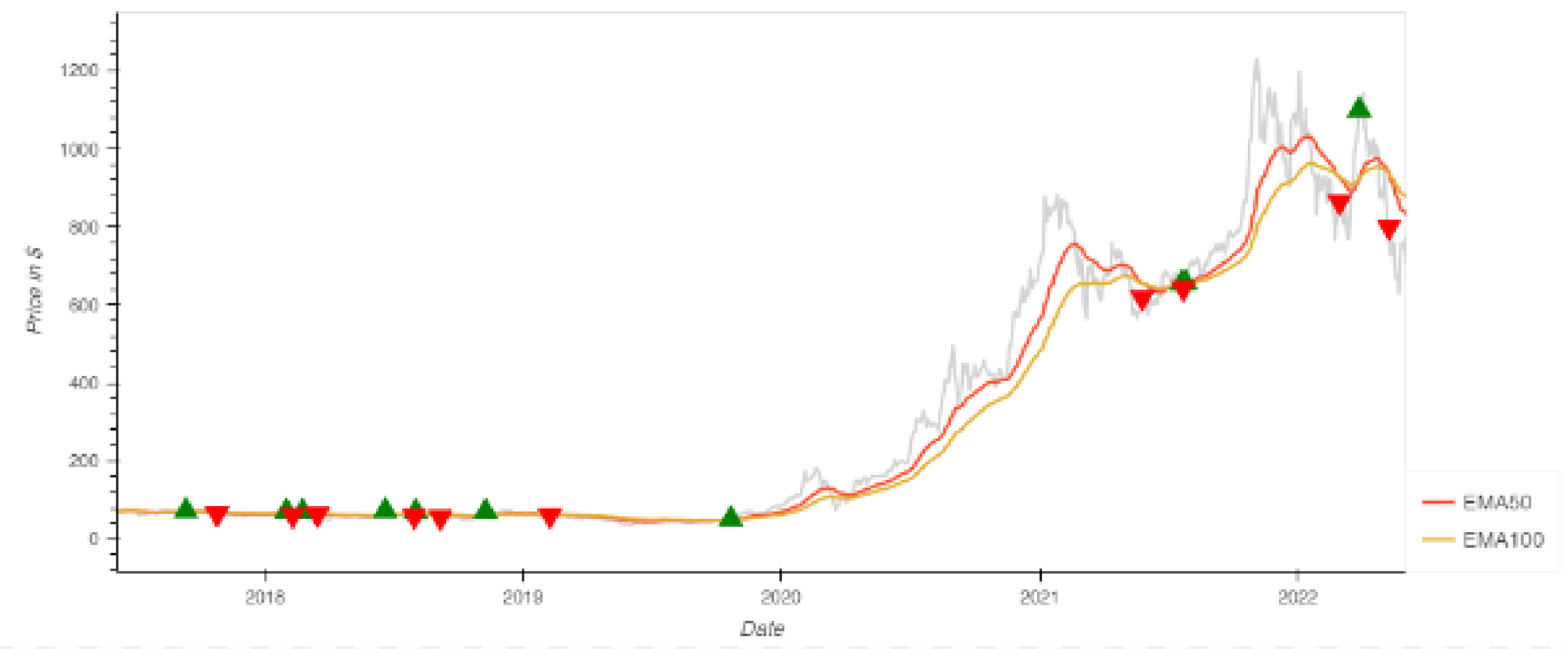
```
sma_df = pd.concat([sma_df.sma50, sma_df.sma100], axis=1)

apd = mpf.make_addplot(sma_df)

mpf.plot(chart_data, type='candle', style='charles',
        title='TSLA SMA 50 v SMA100',
        ylabel='Price (USD)',
        ylabel_lower='Volume',
        volume=True,
        figscale=1.7,
        addplot=apd,
        savefig='tsla_sma_chart.png'
        )
```

BOLLINGER BANDS

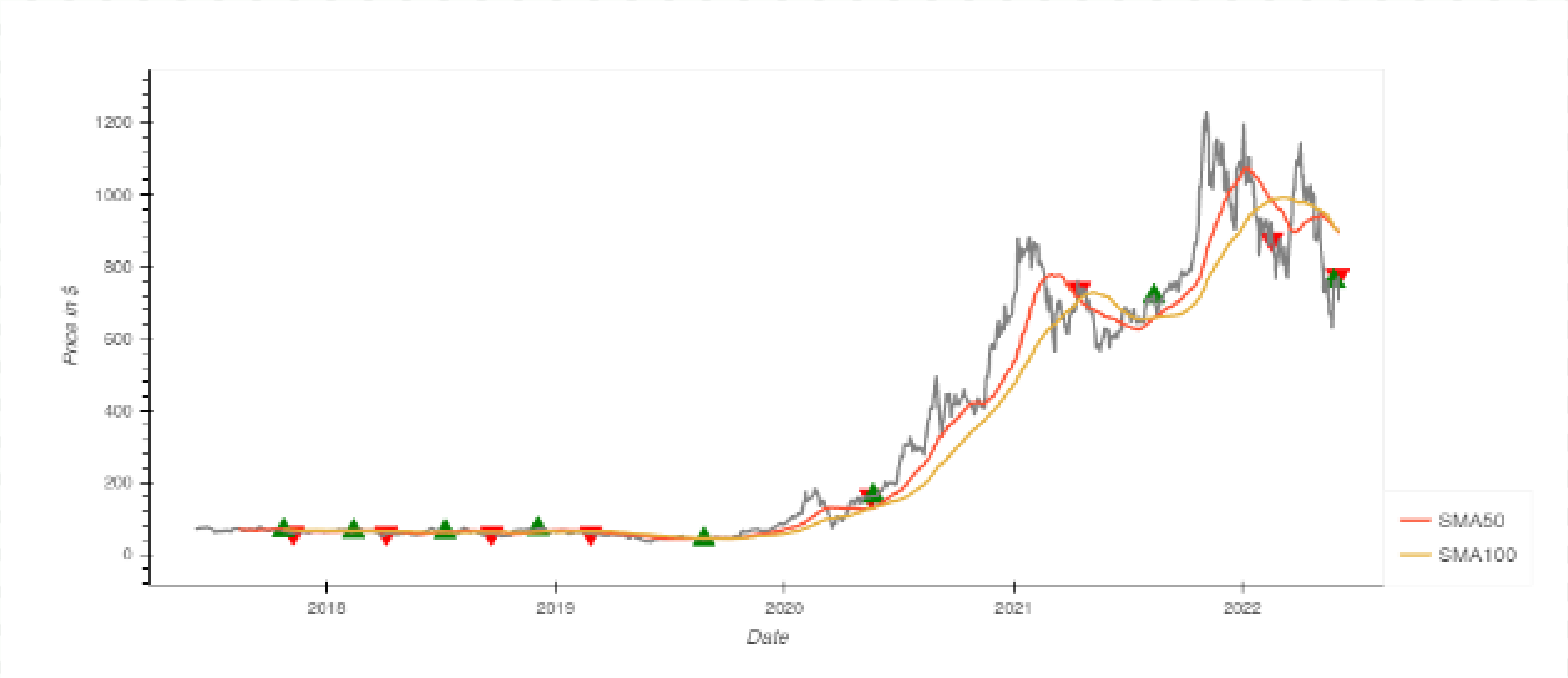


EXPONENTIAL MOVING AVERAGE (EMA) OF CLOSING PRICES

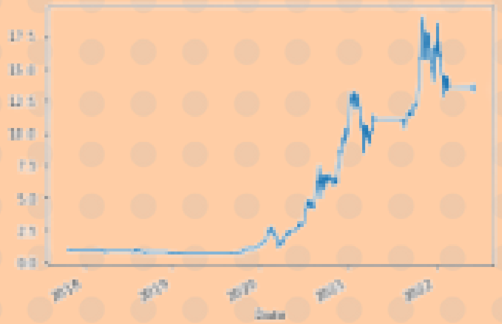




SIMPLE MOVING AVERAGE (SMA) OF CLOSING PRICES



EVAULATION OF MODEL PERFORMANCE



SUPPORT VECTOR CLASSIFIER

DESCRIBE PREDICTIVE MODEL

The objective of SVC is to fit the data you provide, returning a 'best fit' hyperplane that divides or categorises your data - from there you can feed some features to your classifier to see what the predicted class is.

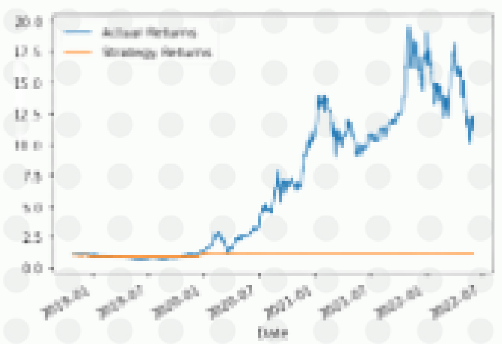
Date	Predicted	Actual Returns	Strategy Returns
2018-10-26	0.0	0.050943	0.0
2018-10-29	0.0	0.011937	0.0
2018-10-30	0.0	-0.014783	-0.0
2018-10-31	0.0	0.022492	0.0
2018-11-01	0.0	0.020633	0.0

DATA PREP + MODEL TRAINING PROCESS

Bollinger Band Signals

SMA

EMA



Bollinger Band Signals

	precision	recall	f1-score	support
-1.0	0.00	0.00	0.00	34
0.0	0.89	1.00	0.94	[illegible]
1.0	0.00	0.00	0.00	35
accuracy			0.88	255
macro avg	0.30	0.33	0.31	255
weighted avg	0.78	0.88	0.83	255

SMA

	precision	recall	f1-score	support
0.0	0.88	0.97	0.92	179
1.0	0.89	0.69	0.78	74
accuracy			0.88	253
macro avg	0.89	0.83	0.85	253
weighted avg	0.89	0.89	0.88	253

EMA

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	153
1.0	0.98	0.96	0.97	100
accuracy			0.98	253
macro avg	0.98	0.97	0.98	253
weighted avg	0.98	0.98	0.98	253

RANDOM FOREST

```
RandomForestClassifier
RandomForestClassifier(n_estimators=10, random_state=0)
```

Bollinger Band Signals

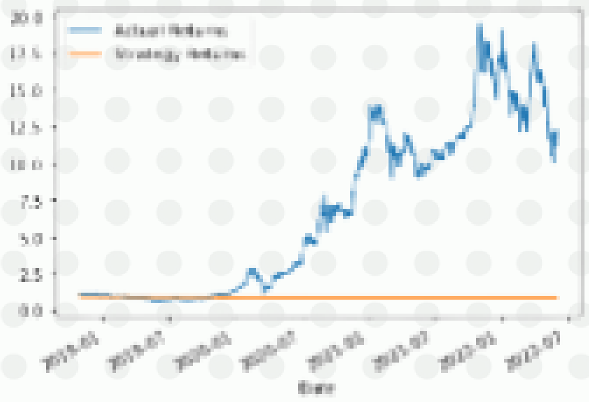
SMA

EMA

ADA BOOST

```
AdaBoostClassifier
AdaBoostClassifier(n_estimators=100)
```

Bollinger Band Signals



SMA

EMA

LSTM

LSTM is an artificial recurrent neural network (RNN) architecture and is well-suited to classifying, processing, and making predictions based on time series data.

- We used LSTM neural network to analyse short term trade scenarios i.e. to decide whether to stay in the market or not. The LSTM data was captured in a particular shape involving "windows" and at each step we predicted the closing price of the day.

- **DATA PREP + MODEL TRAINING PROCESS**

```python
tsla_data['Prev_Close']=tsla_data['Close'].shift(1)
tsla_data.head()
```

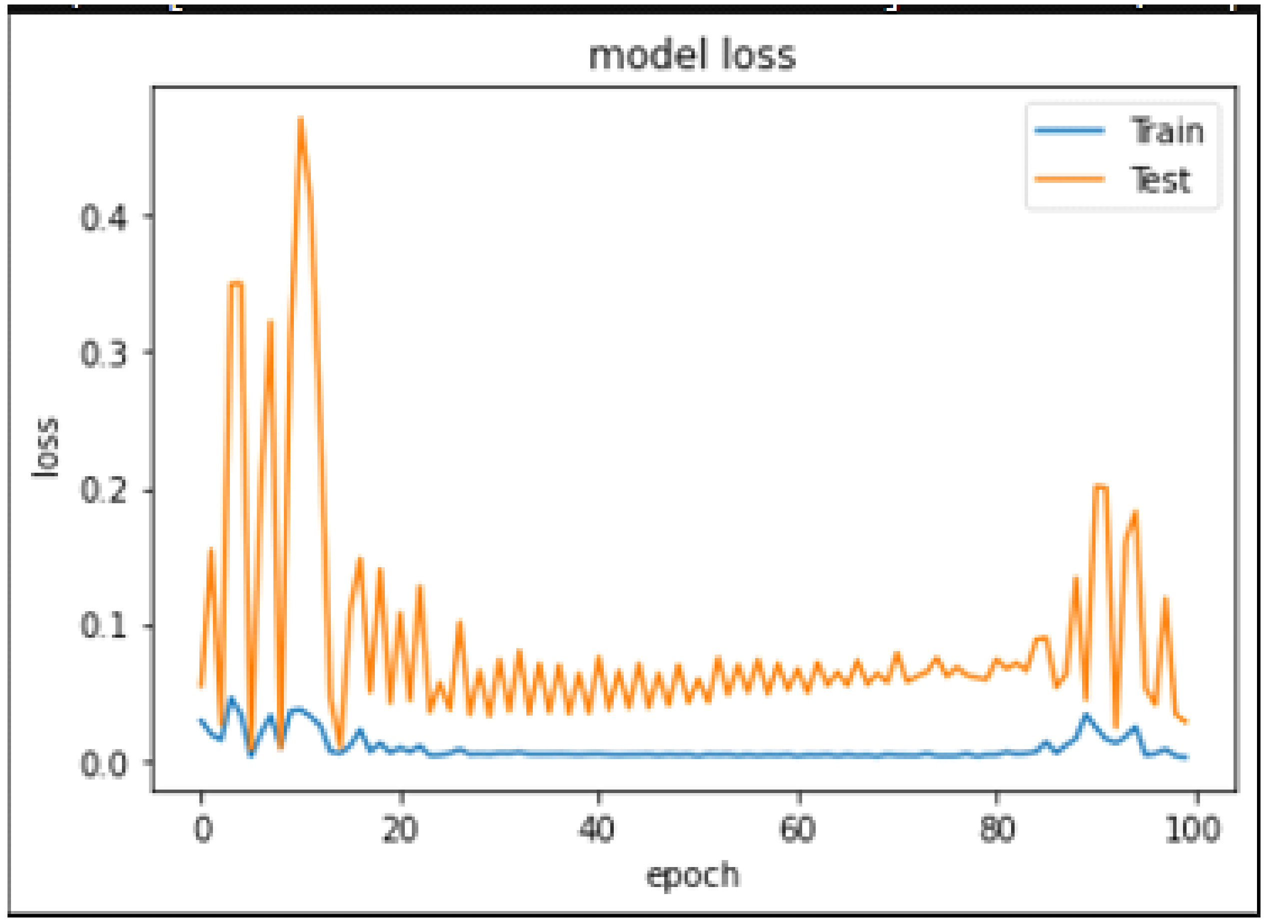
Date	High	Low	Open	Close	Volume	Prev_Close
2017-01-06	46.062000	45.090000	45.386002	45.801998	27639500.0	NaN
2017-01-09	46.383999	45.599998	45.793999	46.256001	19897500.0	45.801998
2017-01-10	46.400002	45.377998	46.400002	45.973999	18300000.0	46.256001
2017-01-11	45.995998	45.335999	45.813999	45.945999	18254000.0	45.973999
2017-01-12	46.139999	45.116001	45.812000	45.917999	18951000.0	45.945999

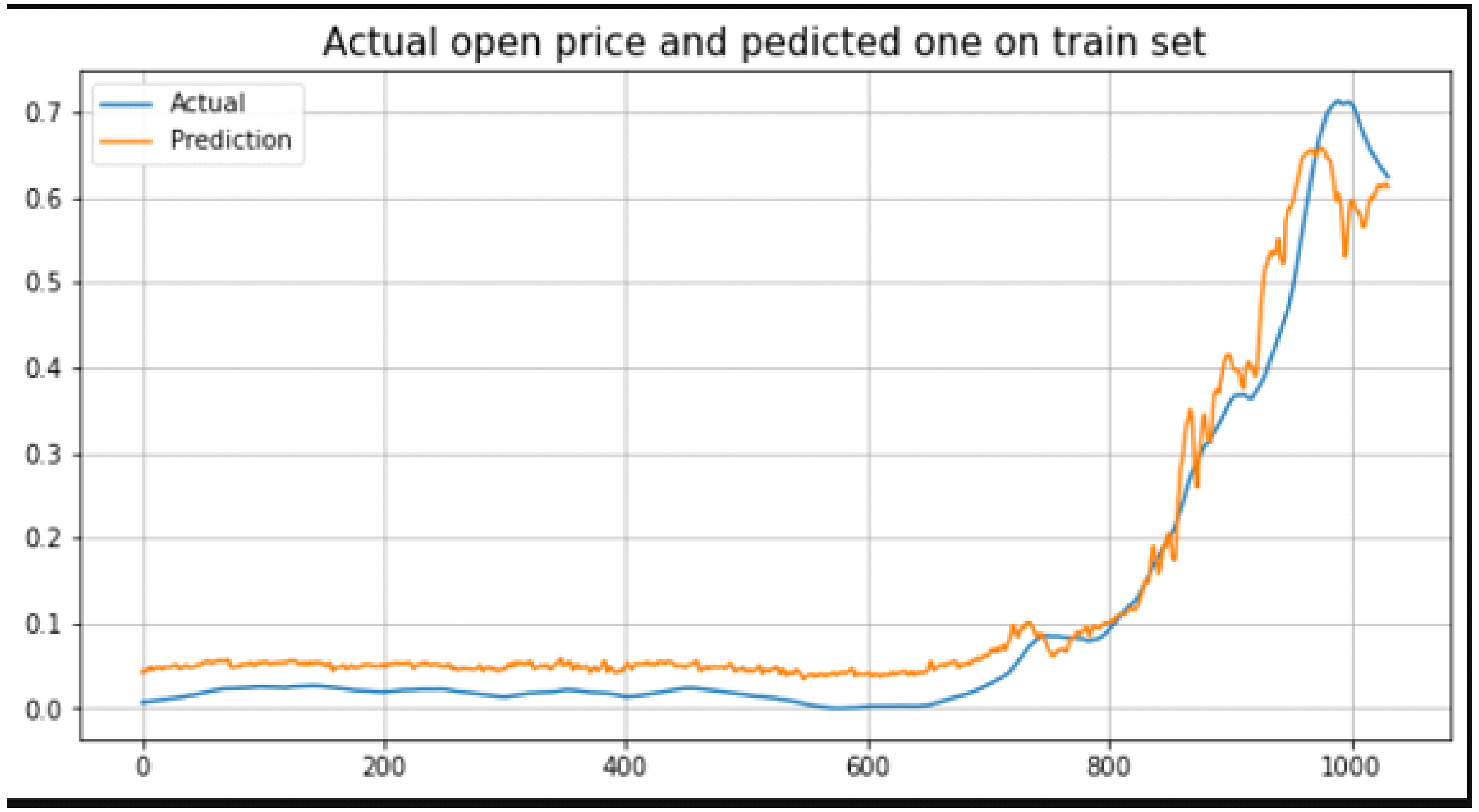
```python
tsla_data["SMA5"]= tsla_data["Close"].rolling(window=5).mean()
tsla_data["SMA50"]= tsla_data["Close"].rolling(window=50).mean()
tsla_data.head()
```

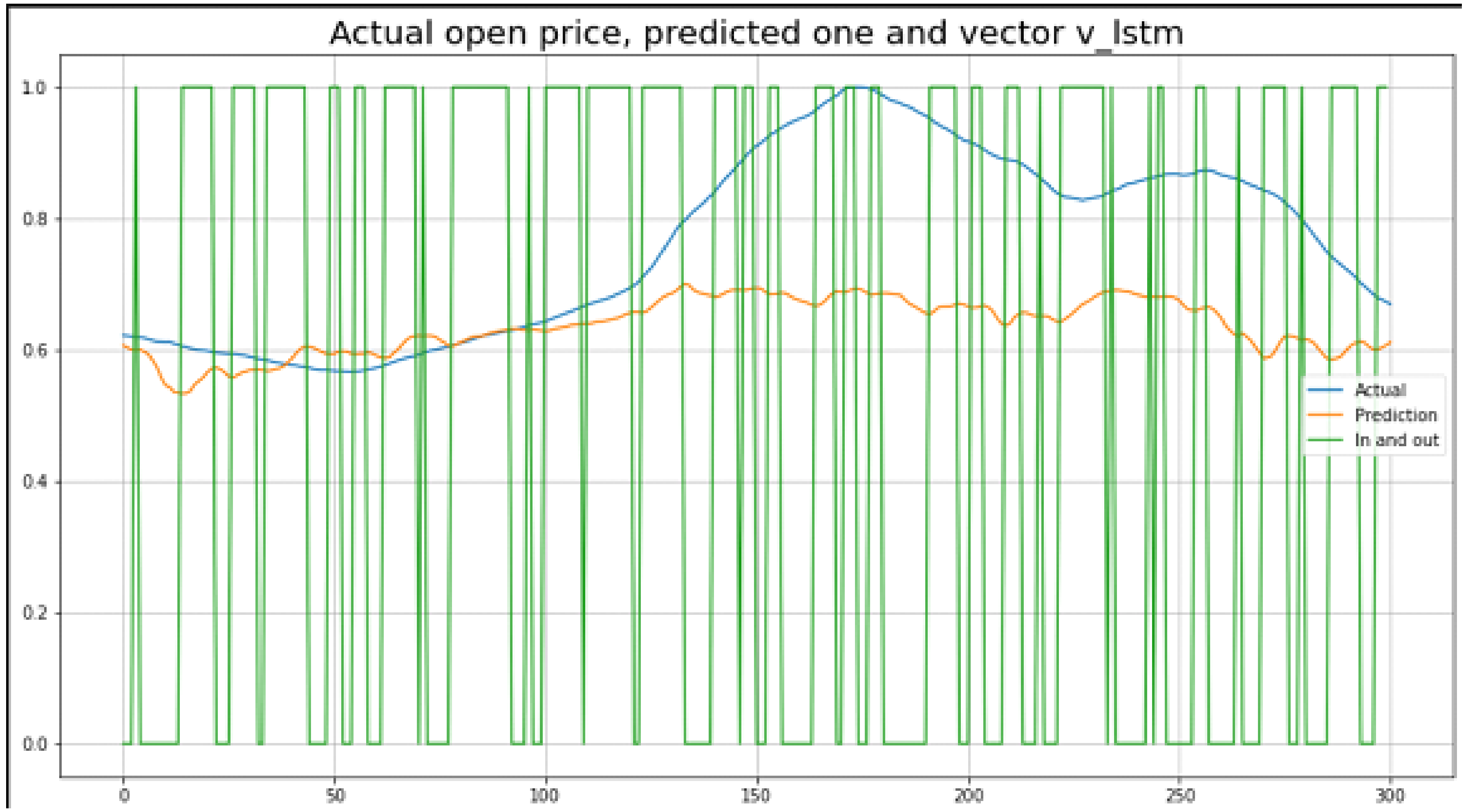
Date	High	Low	Open	Close	Volume	Prev_Close	rapp	SMA5	SMA50
2017-01-06	46.062000	45.090000	45.386002	45.801998	27639500.0	NaN	NaN	NaN	NaN
2017-01-09	46.383999	45.599998	45.793999	46.256001	19897500.0	45.801998	1.009912	NaN	NaN
2017-01-10	46.400002	45.377998	46.400002	45.973999	18300000.0	46.256001	0.993903	NaN	NaN
2017-01-11	45.995998	45.335999	45.813999	45.945999	18254000.0	45.973999	0.999391	NaN	NaN
2017-01-12	46.139999	45.116001	45.812000	45.917999	18951000.0	45.945999	0.999391	45.979199	NaN

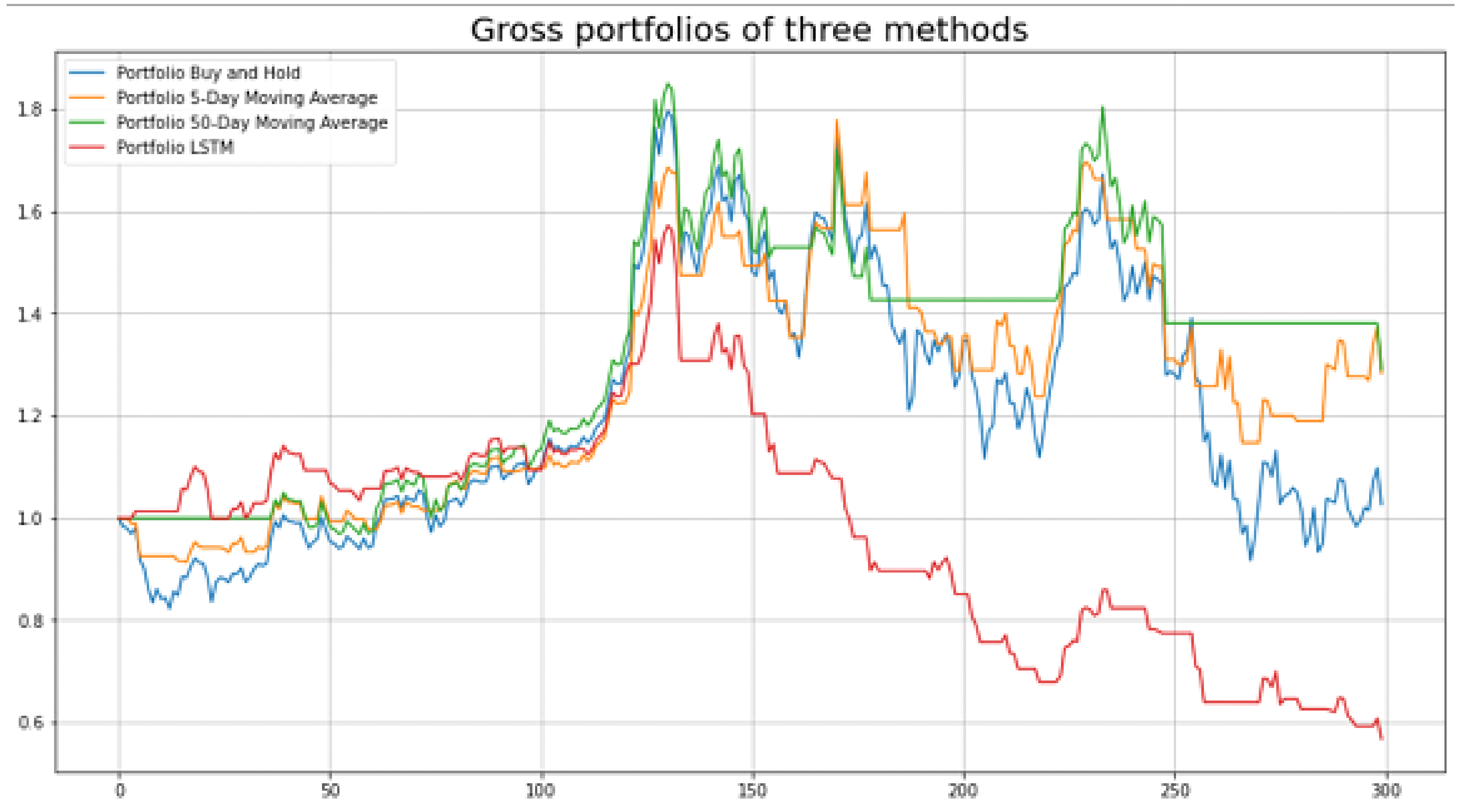

```python
tsla_data['Prev_Close']=tsla_data['Close'].shift(1)
tsla_data.head()
```

Date	High	Low	Open	Close	Volume	Prev_Close
2017-01-06	46.062000	45.090000	45.386002	45.801998	27639500.0	NaN
2017-01-09	46.383999	45.599998	45.793999	46.256001	19897500.0	45.801998
2017-01-10	46.400002	45.377998	46.400002	45.973999	18300000.0	46.256001
2017-01-11	45.995998	45.335999	45.813999	45.945999	18254000.0	45.973999
2017-01-12	46.139999	45.116001	45.812000	45.917999	18951000.0	45.945999






Actual open price, predicted one and vector v_lstm
Actual
Prediction
In and out
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250
300


Gross portfolios of three methods
Portfolio Buy and Hold
Portfolio 5-Day Moving Average
Portfolio 50-Day Moving Average
Portfolio LSTM
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
50
100
150
200
250
300

IMPLICATIONS OF FINDINGS

CONCLUSION