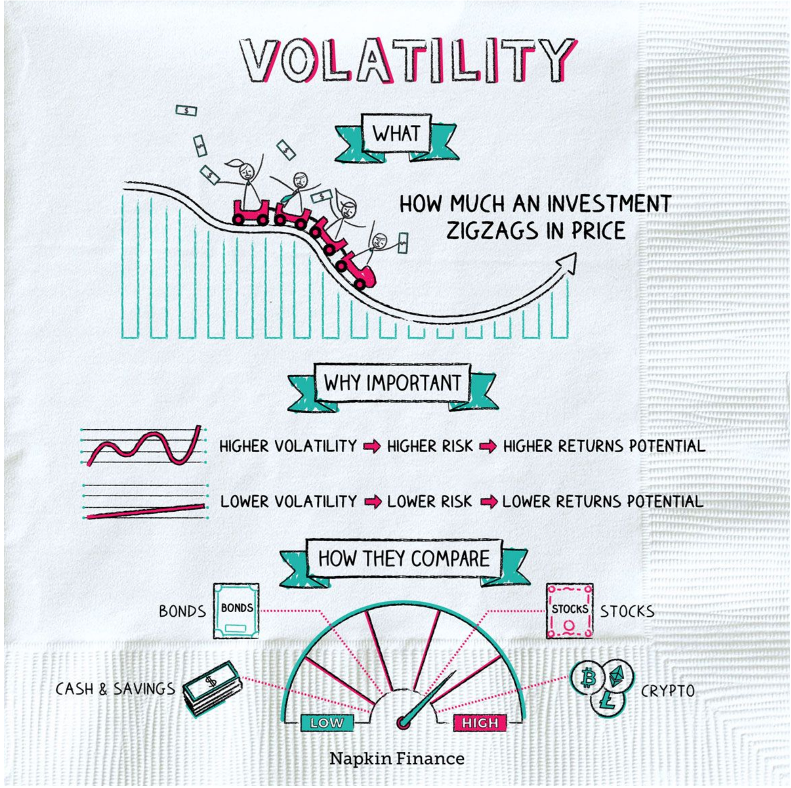
Bitcoin Volatility Prediction

Hammer Capital

The Problem

- Want money
- Know financial markets are a great place to make money
- Bitcoin volatility is a somewhat liquid yet underdeveloped market
- Stocks act like Brownian motion, realized volatility likely does not

Volatility Explanation



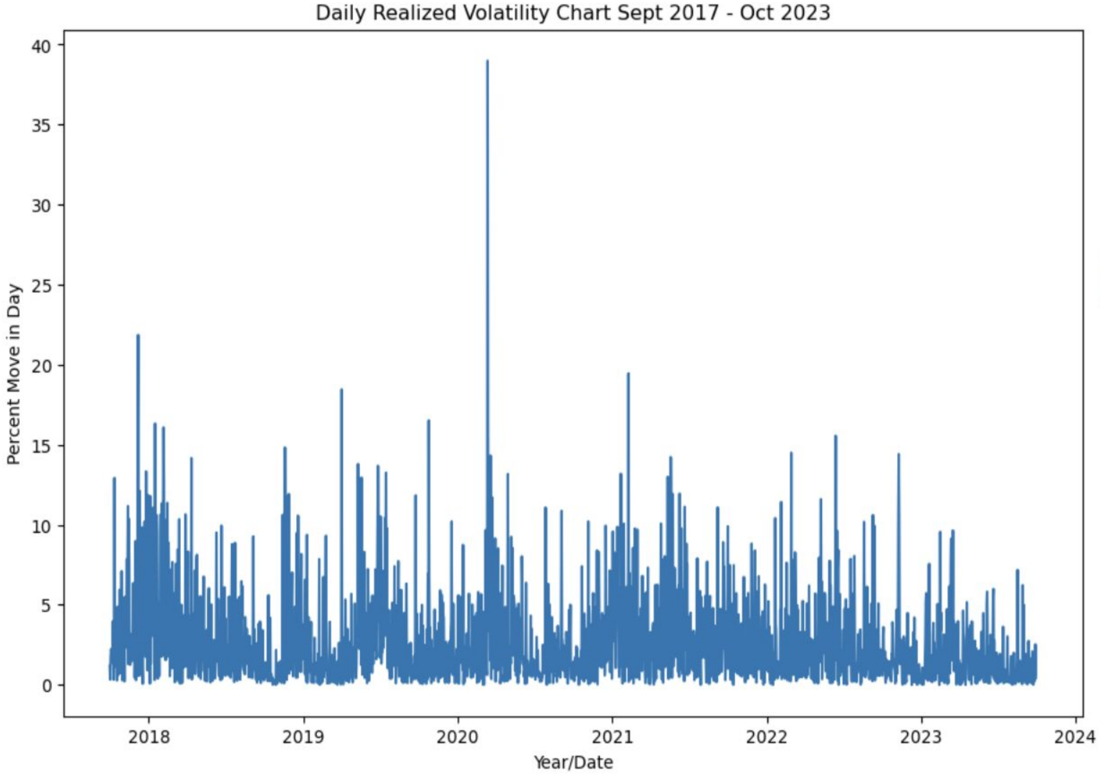
The Thesis

- A machine learning model for future volatility based on past data can add predictive value on the trading floor and assist Hammer Capital in capturing edge

The Data

- Bitstamp Bitcoin data
- Sampled once per day (for volatility this is the frequency we need as we can trade this way)
- Sept 2017 - Nov 2023

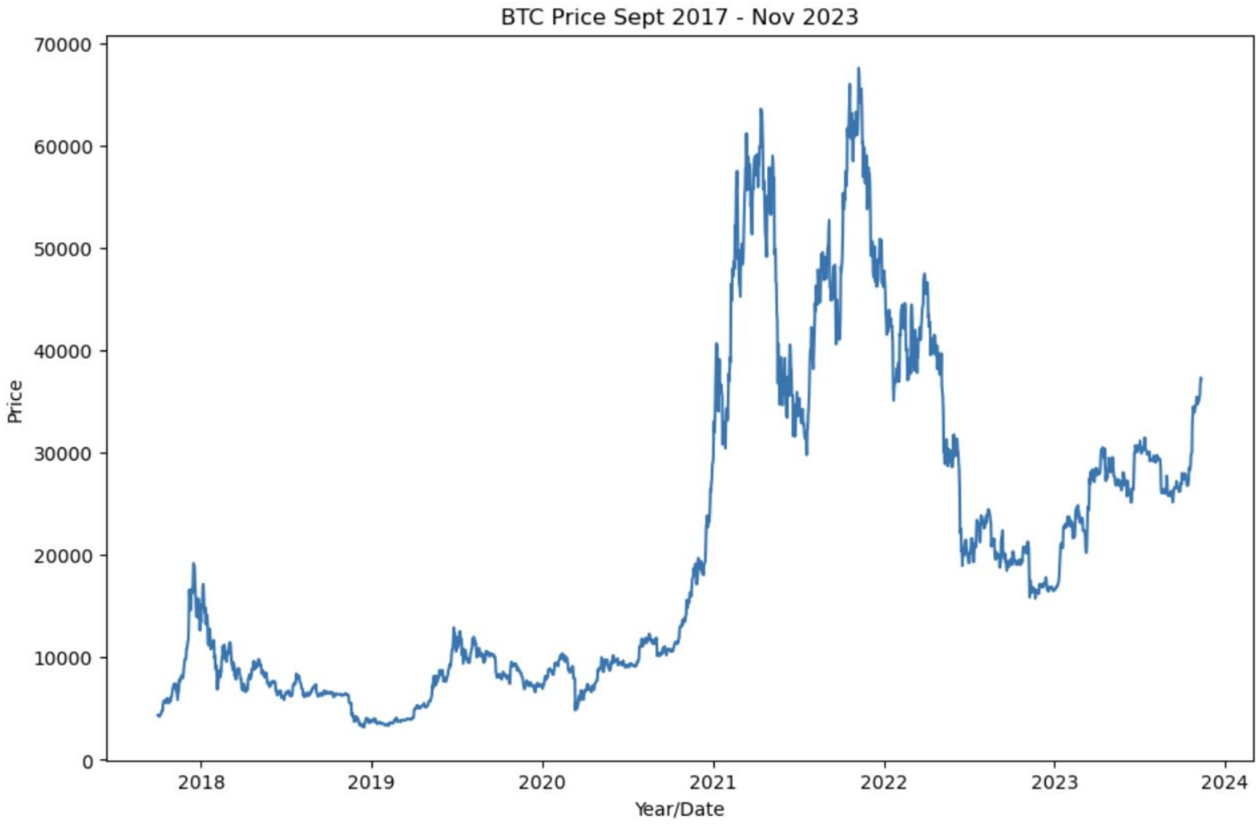
Daily Realized Volatility




```
print(df.nlargest(5, 'realized_absolute_vol'))
```

```
               close  realized_absolute_vol
datetime
2020-03-12   4841.67              38.980058
2017-12-07  16599.99              21.848203
2021-02-08  46416.45              19.450265
2019-04-02   4899.63              18.453843
2017-12-06  13623.50              16.669521
```

Bitcoin Price Since 2017

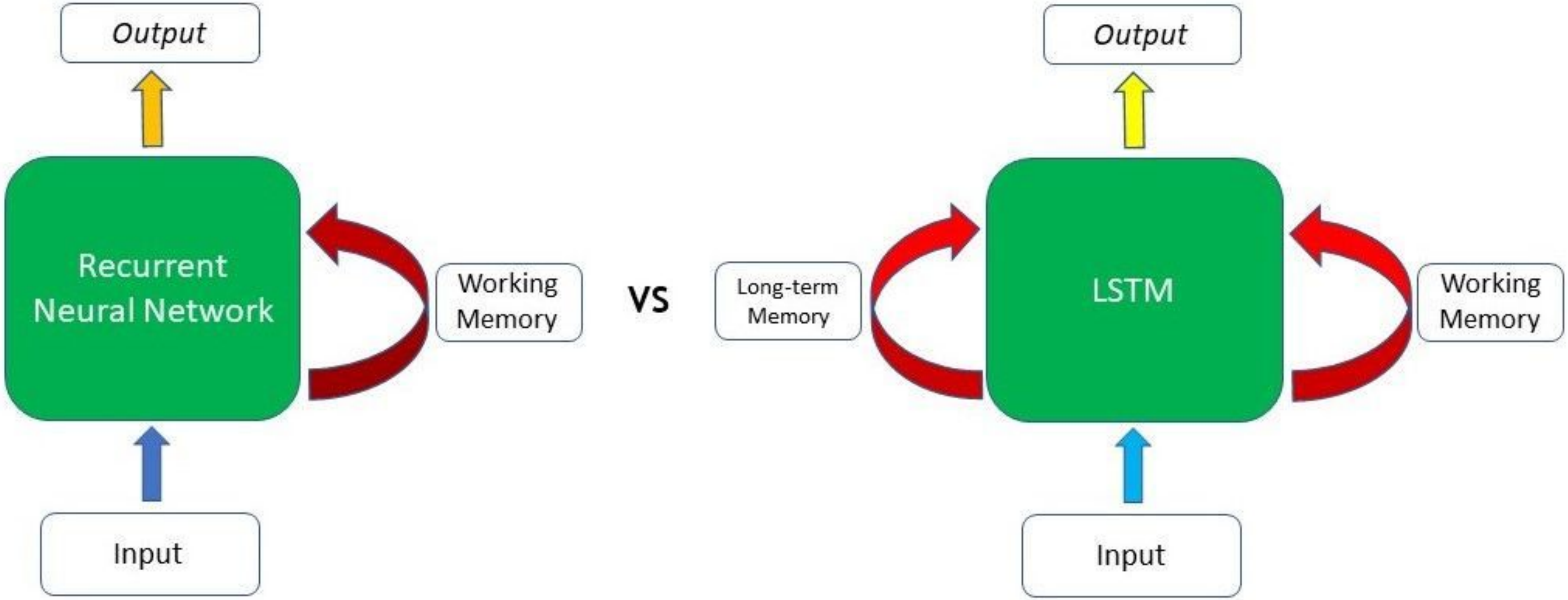


Model

LSTM

- Long short-term memory (LSTM) network is a recurrent neural network (RNN), aimed to deal with the vanishing gradient problem present in traditional RNNs. Its relative insensitivity to gap length is its advantage over other RNNs, hidden Markov models and other sequence learning methods.

Classic RNN vs LSTM



Nov 12 is first day of prediction

Note we likely would have bought/sold correctly based on our predictions relative to market prices

```
Predictions for the next 10 elements:
[[1.2299799]
 [1.8996669]
 [2.0241373]
 [3.0054111]
 [1.8073826]
 [2.350042 ]
 [1.8226259]
 [2.018779 ]
 [2.5403645]
 [2.4648616]]
```

```
df[-4:]
```

datetime	close	realized_absolute_vol
2023-11-12	37086.0	0.177649
2023-11-13	36485.0	1.620558
2023-11-14	35564.0	2.524325
2023-11-15	37877.0	6.503768

Final Model RMSE

Train: 2.51 RMSE
Test: 2.52 RMSE

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

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ are predicted values

$y_1, y_2, \ldots, y_n$ are observed values

n is the number of observations

Next Steps

- Multivariate model intaking factors such as where the market is pricing forward vols
- Add day of week explicitly to model, add up/down move explicitly to the model
- Add calendar data such as CPI/FOMC, US market holiday etc.
- Consider dampening the fat tails in training data such that the model can better predict and tails don’t overly impact RMSE results

“I lost the five grand. What’s our next step?”

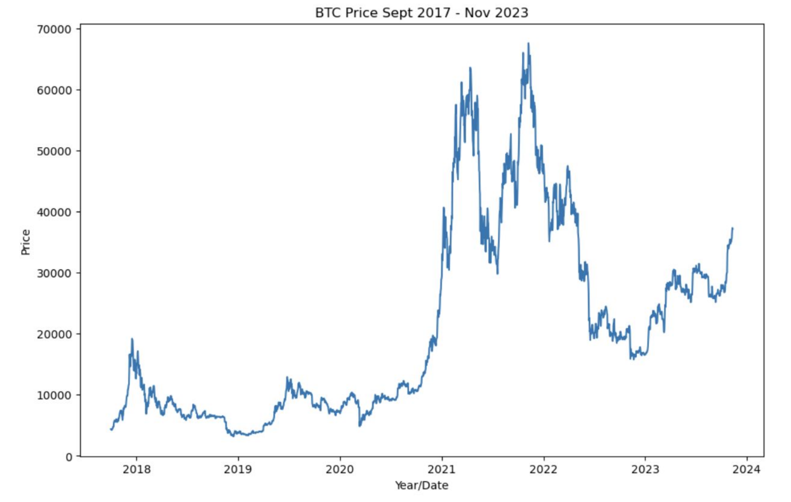
Contact Information

Michael Hammer
Email: michaelhammerb@gmail.com
Github: https://github.com/michaelhammer1
LinkedIn: https://www.linkedin.com/in/michaelhammerb/

```
df[['close','realized_absolute_vol']].mean()
```

```
close                    20812.966010
realized_absolute_vol        2.576125
dtype: float64
```

```
Monday average vol 3.081390495122366
Tuesday average vol 2.6493359499267495
Wednesday average vol 2.8456509522497555
Thursday average vol 3.010997561565898
Friday average vol 2.612149440322917
Saturday average vol 1.742963021584130
Sunday average vol 2.0903891132113
```



Goal?

- Build a model to predict future realized volatility that is better than average vol / the previous days vol
- Show real life prediction and what happened
- Get a measurable day-of-week effect