BITCOIN PRICE PREDICTIONS

Price predictions using LSTM deep learning & ensemble methods

Ken Constable, December 2021
Github: https://github.com/kconstable
Kaggle: https://www.kaggle.com/kenconstable

Research Problem

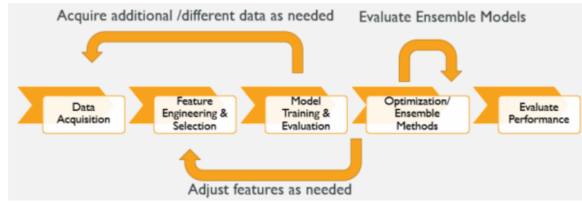
Bitcoin prices are extremely volatile and are highly influenced by politics and investor sentiment. It has experienced extreme growth & volatility over the last 3-years, with a total return of 1,475% during the period (figure 1). Bitcoin cannot be analyzed using fundamental analysis like traditional stocks, and technical indicators and economic indicators alone cannot accurately predict Bitcoin prices. This project explores whether a trading strategy can be developed to exploit price prediction differentials between long and short ensemble models consisting of long-short-term-memory (LSTM) recurrent neural nets and time-series forecasting models.

Figure 1: Bitcoin Prices & Volume

Methodology

A five-step methodology was followed, where the first stage was to acquire & clean structured and unstructured data from multiple sources as possible features in the LSTM models. Additional features such as sentiment scores, calculation of technical indicators, and the creation of lagged features were explored in the second stage. Features were then selected using recursive feature elimination (RFE) to select the most influential features on the closing price of Bitcoin. The third stage consisted of transforming the feature data in preparation for deep learning and building and training LSTM models. The models were optimized, and the long and short ensemble models were constructed in the fourth stage. The model predictions were evaluated in a historical back-test on the testing data, and a trading strategy was evaluated for utility in the last stage (figure 2).

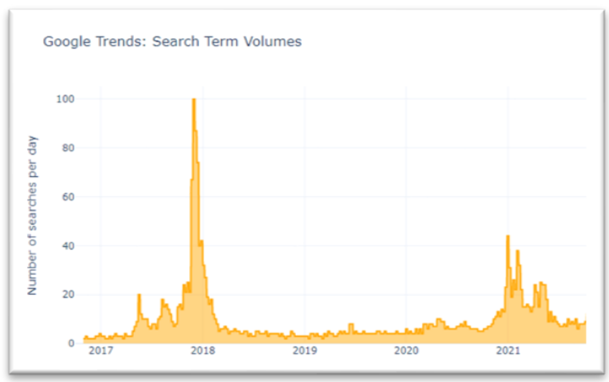
Figure 2: Project Methodology



Data Acquisition

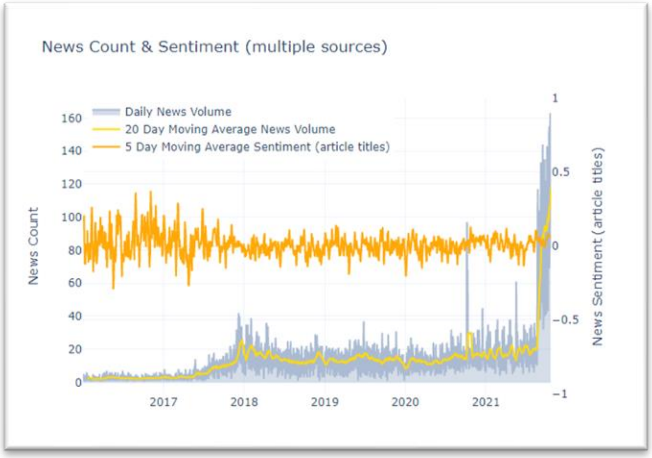
Data from six different sources were collected, cleaned, and analyzed. Commodity prices, market index values, FX rates, and economic indicators were sourced from Alpha Vantage. Bitcoin spot and futures prices were pulled from the public API at gate.io. Interest in investing in Bitcoin was quantified by querying google trends for the number of web searches for "how to buy bitcoin" and then converted into a time-series of daily data points (figure 3).

Figure 3: Bitcoin Web Searches



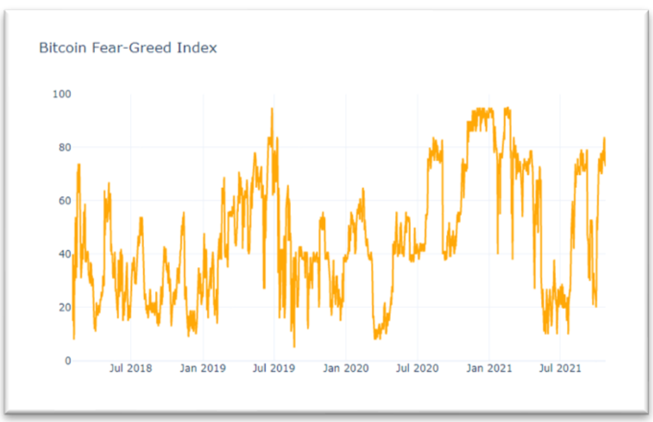
General interest in Bitcoin was determined using the google news API to pull news stories that mention bitcoin in the body of the article and aggregated by publication date (figure 4).

Figure 4: Bitcoin News Volume & Title Sentiment



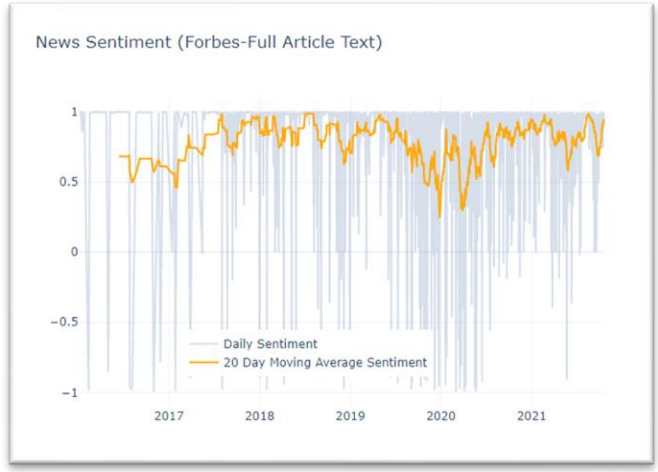
Sentiment towards Bitcoin was estimated in three ways; The fear-and-greed-index published by Alternative was used as a proxy of major crypto-currency sentiment and included as a possible model feature (figure 5).

Figure 5: Bitcoin Fear & Greed Index



The sentiment score was calculated for the titles of approximately 30,000 news articles pulled from google news (figure 4), and approximately 3,500 full-text articles about Bitcoin on forbes.com were scaped and used as a source to calculate sentiment (figure 6)

Figure 6: Bitcoin Sentiment (full-text articles from forbes.com)



Feature Engineering

Technical indicators including Bollinger Bands, the moving-average-convergence-divergence (MACD), relative-strength-index (RSI), and the stochastic oscillator were calculated for Bitcoin prices and the calculated sentiment scores and added as additional model features (figure 7).

Figure 7: Bitcoin Technical Indicators

Some indicators can have a delayed effect, where the value from a previous period could have a larger influence on bitcoin prices than the current value of the indicator. To capture this effect, correlations between closing prices and current/lagged features were calculated and compared for all features, with the largest added as an additional feature. The number of days lagged was searched for iteratively until the correlation with the largest magnitude was found between 10 and 120 days before the current day.

Feature Selection

Recursive Feature Elimination (RFE) with a Random Forest regression model was used to calculate the importance and rank of each feature. The random forest estimator starts by using all available features, then eliminates the least influential feature recursively until the desired number of features remains. A total of 76 features were considered for inclusion in the LSTM models, and the RFE process selected the top 44 features.

Selected Features

- Bitcoin Spot Prices (open, high, low, close, volume)
- Bitcoin Futures Prices (open, high, low, close)
- Other Cryptocurrency Prices (ETH, DOGE, LTC)
- Crypto Fear and Greed Index
- The sentiment of Bitcoin articles (full-text) from Forbes.com
- The sentiment of Bitcoin articles (title of article, any source) from google news
- Number of news articles written about Bitcoin from google news
- Number of Bitcoin-related search queries on google trends
- Market Indexes (Energy Sector, The Nasdaq, Volatility index-lagged 110 days)
- Commodities (Natural gas prices, Oil prices lagged-110 days)
- FX Rates (USD/EUR, USD/GBP)
- Technical Indicators on Bitcoin spot prices (RSI, Bollinger Bands, MACD, Stochastic Oscillator)

Modeling

LSTM Models

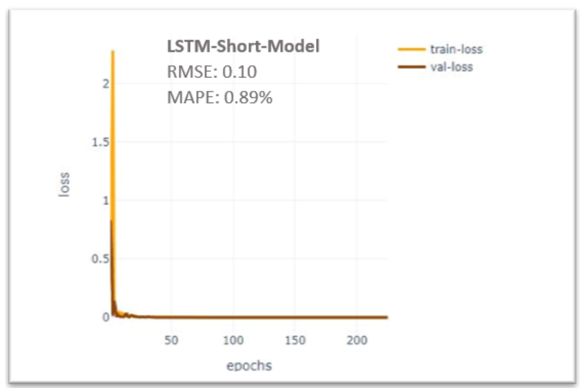
LSTM models are a special type of recurrent neural net (RNN) that are capable of learning long-term dependencies in sequences of data. They can recognize patterns in sequences of data by feeding the node output back into itself as input which makes them good candidates for predicting time-series data.

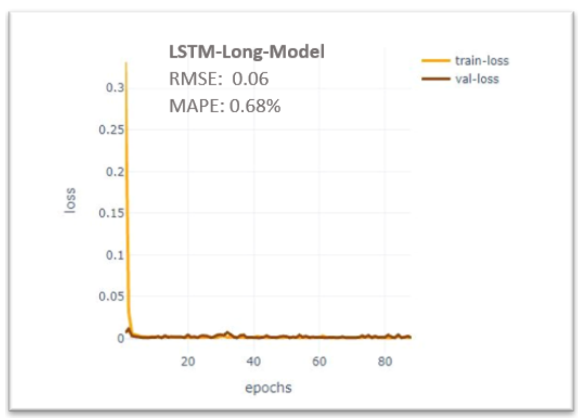
Two LSTM models were trained. A short model takes the previous 25-days of feature data as input and predicts tomorrow's closing price, and a long model takes the previous 100-days of feature data as input and predicts the next 3-days of prices. The short model captures short-term patterns while the long model can better detect longer-term trends.

Training

Daily data was collected from Feb 2, 2018, to Sep 28, 2021. All feature data was complete except for Bitcoin Futures, which was only available from November 2019. The missing data was imputed using a Random Forest imputer. The data was split into training (80%) and test sets (20%), with 10% of the training data withheld for validation. Early stopping based on validation loss was employed which determined the number of epochs used in training. The Root Mean Square Error (RMSE) & Mean Absolute Percentage Error (MAPE) were used to evaluate performance on the test data (figure 8).

Figure 8: LSTM Training Metrics





Optimization

The LSTM model hyper-parameters were optimized using keras-tuner to find optimal network capacity, layers, nodes, and batch sizes to minimize the loss function.

Ensembles

Two additional models were added to the LSTM models to form ensemble models to reduce daily prediction errors. The Facebook Prophet model, which is a time-series forecasting model similar to an ARIMA model, was trained on Bitcoin closing prices to predict the next 3-days of prices. LSTM models weigh all inputs equally, but prices are often influenced more by recent data than older data. To adjust for this, a naïve model which predicts that tomorrow's closing price will be the same as today's closing price was added to the short-ensemble model. This effectively adds weight to the most recent data inputs (table 1).

Table 1: Ensemble Model Weights

Models	**Short-Ensemble-Model**	**Long-Ensemble-Model**
LSTM	60% (short)	80% (long)
Prophet	20%	20%
Naïve	20%	0%

Figure 10: Predictions

Figure 11: Prediction Errors

Model Evaluation

Historical Back-Test

A historical back-test of model predictions was done on the out-of-sample test data from the end of January 2021 to the end of September 2021 to see how the models would have performed. On January 28th, the models predict the closing prices for January 29th to the 31st. On January 29th, the actual price is compared to the predicted price, and the 1-day prediction error is calculated. The 3-day prediction error is calculated 3-days after they were initially predicted once actual prices are known. The model then rolls forward by a day and repeats the process until the end of September. The model predictions are plotted in figure 10 and the distribution of prediction errors for each model is shown in figure 11. By combining individual models into ensembles, we can reduce the daily prediction errors.

Trading Strategy

A true test of whether a model can add value is to apply a trading strategy to model predictions and evaluate profit/loss in a back-test. Simple trading rules were developed that take advantage of price differentials between the long and short models to see how the strategy would have performed on the out-of-sample test data.

Buy-Signal

A buy signal is indicated when the long-model price in 3-days is greater than the short-model price tomorrow, provided the differential in prices is greater than a 1% threshold.

Sell-Signal

A sell signal is indicated when the price tomorrow as predicted by the short-model is greater than the long-price model in 3-days, provided the differential in prices is greater than a 1% threshold. An additional sell signal was added to sell when the maximum loss from the previous purchase exceeds 15%. This prevents the model from holding indefinitely in a falling market when the long-model is consistently higher than the short-model, even though both are predicting falling prices.

To test the strategy, the back-test starts with $40,000 in cash and purchases 1-bitcoin on Jan 30, 2021. The model then buys/sells a single bitcoin according to the trading signals. When not invested in bitcoin, the assets are held in cash until the next buy-signal is indicated. The transaction costs are assumed to be zero. The light blue sections in the plot in figure 12 show periods when the model invested in bitcoin, and the dotted-gray line show the total value of the portfolio after each trade is executed which represents realized profit/loss.

The model purchased Bitcoin on market dips and sold and took profits in a rising market. In falling markets, the model will sell at a loss and wait for the next buy signal to make another purchase. This can be seen in the period between May and July, where the model sold when the loss from the previous purchase surpassed 15%, then bought back into the market when prices flattened. After 6-trades were executed, a profit of 20k was realized which is a 50.3% return on the initial investment of 40k. For comparison, if you bought and held bitcoin for the same period, you would have an unrealized profit of 19.7% (figure 12).

Figure 12: Trading Profit / Loss

Insights and Future Work

Predicting Bitcoin prices is particularly difficult. Models will likely never have enough information, and factors that influence prices can change over time. Trying to create a model to accurately predict an absolute closing price is likely a futile effort. However, there may be value in comparing relative price predictions on a short and longer-term basis. There will always be noise in daily fluctuations but knowing what a price should be based on known factors could be exploited.

For the testing period at least, the ensemble models and trading strategy added significant value over a simple buy-and-hold strategy. However, at any given time this situation could be reversed, and it is not certain this strategy can add value over long periods. The model also assumes there are zero transaction costs which is not a realistic assumption. Transaction costs could greatly reduce profits if the models signal frequent trading.

Ensemble methods add a lot of value, even when the individual models provide poor predictions. This was evident with the prophet model which can greatly over or under-estimate prices. It was shown that allocating a small portion of the ensemble weight to this model lowers the overall prediction error.

The length of the feature window is a significant factor that can be used to calibrate prediction windows. Experimentation with shortening (intra-day) and lengthening (weeks, months?) these windows might reveal trading strategies that have practical utility.

Transfer learning between cryptocurrencies is likely not applicable. Many of the features used in this model were customized to bitcoin such as futures prices, news articles, and web searches. Different cryptocurrencies will require different features.

Additional models such as Random Forest or XGBoost could be explored for inclusion in the ensembles. As a final thought, the trading strategy used in this study was quite simple. Much more sophisticated strategies could be developed and implemented which may improve the results.