stratosph.com
contact@stratosph.com

Head to head comparison between Deep Learning and Traditional Machine Learning Method for Cryptocurrency Price Prediction

February 2022

There is a widespread impression that the crypto market is less stable than traditional commodity markets. It is perceived to be highly volatile, uncertain, and unpredictable due to the fact that it is being affected by many technical, sentimental, and regulatory factors. Many studies have been done on various cryptocurrencies to forecast price movements. Many Artificial Intelligence methods have been proposed to solve this task. Two main families of models are usually compared against each other: the traditional and the deep learning one. It is reported that the forecasts are accurate enough to generate sustainable trading strategies. However, after conducting in-house research, in our point of view, the final conclusions are not clear.

In this short note, we would like to once again benchmark them in a systematic manner. We use two well known and powerful AI frameworks: pytorch for deep learning and scikit-learn for traditional methods to validate our research.

To represent Deep Learning methods, we use the recurrent family with two members: Long Short Term Memory and Gated Recurrent Units (GRU). For the traditional one, we try all the usual methods available in scikit-learn. We present below the results for the most representative ones.

Experiment setup

Data

We retrieve data from the public API of cryptocompare.com. The data consists of price (open, close, high, low) and volume. We focus on bitcoin (btc) and ethereum (eth). The task is to predict their close price in the next day using the data in the past. Instead of predicting the absolute value of the price y_i, we make the models predicting its relative ratio with the past price: $y_i / y_{i-1} - 1$. We think that this normalization of the output will make models more stable.
The test data is the data in the last 3 months. The remaining data is used as the training data.
To compare methods, we use mean absolute error and root-mean-square error.
Mean absolute error (MAE) is calculated as:

$$\text{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \left| y_i - \hat{y}_i \right|.$$

While root-mean-square error is calculated using this formula:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

The model which gives lower errors is said to be better.

Experiment 1

The goal of this experiment is to do feature selection, i.e., We compare three configurations:

- Use only close price or,
- Use it in corporate with 3 other price data (open price, high price, low price) or,
- Use it in corporate with 3 other price data (open price, high price, low price) and 2 volume data (volume frome, volume to).

The model used in this experiment is Long Short Term Memory (LSTM) with the configuration as follows:

window_len = 7
zero_base = True
lstm_neurons = 100
epochs = 20
batch_size = 32
loss = 'mse'
dropout = 0.2
optimizer = 'adam'

coin	bitcoin		ethereum	
error	mean absolute error	root-mean-square error	mean absolute error	root-mean-square error
all	1529.30	2011.56	128.68	169.19
price close + price open, price high, price low	**1386.51**	**1876.27**	125.87	168.03
price close	1439.45	1914.64	**118.60**	**158.33**

Remarks

- The volume information is not useful, at least with this type of model.
- Additional price data (price open, high, low) are useful for bitcoin, but only slightly. On the contrary, they seem not useful for Ethereum.

Experiment 2

The goal is to compare the traditional methods with more advanced and state-of-the-art deep learning methods. The past data is the best configuration for bitcoin induced from experiment 1 (price close + price open, price high, price low). The LSTM model is kept while the other traditional methods are evaluated.

coint	bitcoin		ethereum	
error	mean absolute error	root-mean-square error	mean absolute error	root-mean-square error
LSTM	1386.51	1876.27	125.87	168.03
SVM	1574.03	2124.39	151.64	188.38
Linear Regression	1410.46	1864.50	200.99	266.46
Decision Tree	1924.63	2472.14	167.07	227.60
MLP Regression	1512.07	2021.39	135.11	178.05
Lasso	3803.97	4656.27	334.55	418.10
Elastic Net	3803.97	4656.27	334.55	418.10
Ridge	**1379.25**	1856.84	118.09	154.62
Bayesian Ridge	1385.34	**1852.70**	**116.49**	**153.01**

Remarks:

- For bitcoin, LSTM is the best if we consider both two criteria. Surprisingly, the difference between two groups is not large: mean absolute error: 1386.51 of LSTM versus 1379.25 of Ridge and root-mean-square error 1876.27 of LSTM versus 1852.70 of Bayesian Ridge.
- For ethereum, similar things happen. Bayesian Ridge is the best but again, the differences are small.

Experiment 3

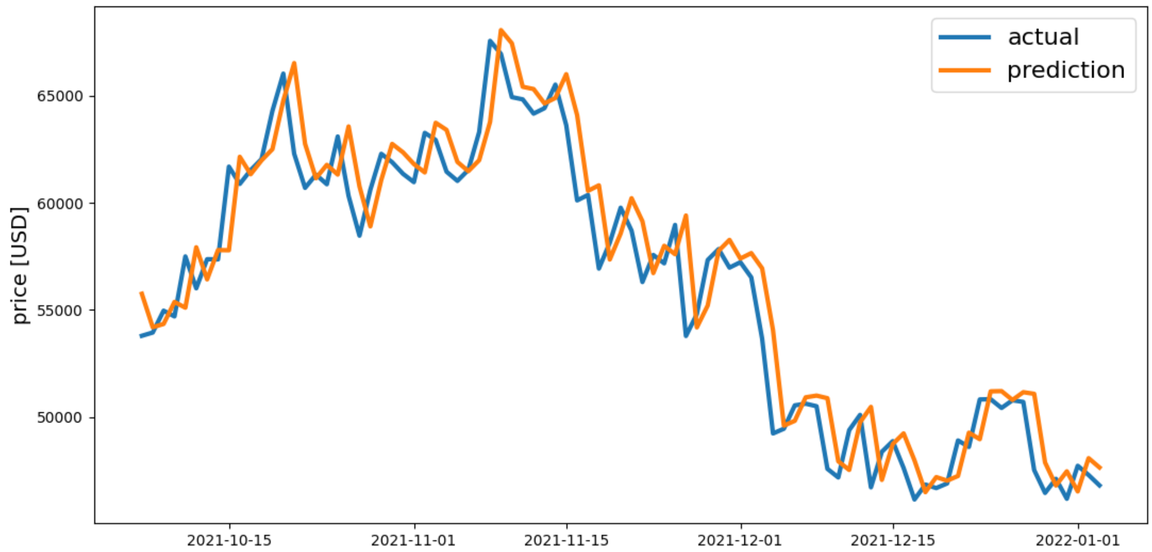
Window size: We compare between only using the current day data versus the past n days (with n = 1, 3, 7). We use open, close, high, low prices.

coin		bitcoin		ethereum	
window		mean absolute error	root-mean-square error	mean absolute error	root-mean-square error
7	LSTM	1386.51	1876.27	125.87	168.03
	Linear Regression	1410.46	1864.50	200.99	266.46
	Ridge	1379.25	1856.84	118.09	154.62
	Bayesian Ridge	1385.34	1852.70	**116.49**	**153.01**
3	LSTM	1399.20	1843.06	126.79	162.24
	Linear Regression	1407.49	1855.24	123.60	159.18
	Ridge	1384.64	1851.56	123.82	159.42
	Bayesian Ridge	1400.40	1851.91	123.22	158.75
1	LSTM	**1366.77**	**1801.40**	122.31	159.16
	Linear Regression	1375.67	1814.68	122.52	160.09
	Ridge	1375.67	1814.68	122.52	160.09
	Bayesian Ridge	1375.67	1814.68	122.52	160.09

Observations:

- For Bitcoin, n = 1 gives the best results. It means that the past data is not very meaningful for the model we are currently testing.
- For Bitcoin, LSTM gives the best results but the difference in quality is small, likely insignificant in practice.
- For Ethererum, n = 7 gives the best results, with the Bayesian Ridge method. Nevertheless the differences are also small.

The graph below illustrates the results for the model LSTM with n = 1.



References

[1] Deep Learning-Based Cryptocurrency Price Prediction Scheme With Inter-Dependent Relations https://ieeexplore.ieee.org/document/9558869
[2] A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms https://www.mdpi.com/2673-2688/2/4/30
[3] Cryptocurrency price prediction using deep learning https://towardsdatascience.com/cryptocurrency-price-prediction-using-deep-learning-70cfca50dd3a