What drives bitcoin? An approach from continuous local transfer entropy and deep learning classification models

Andrés García-Medina
CONACyT Research Fellow at Centro de Investigación en Matemáticas
andres.garcia@cimat.mx

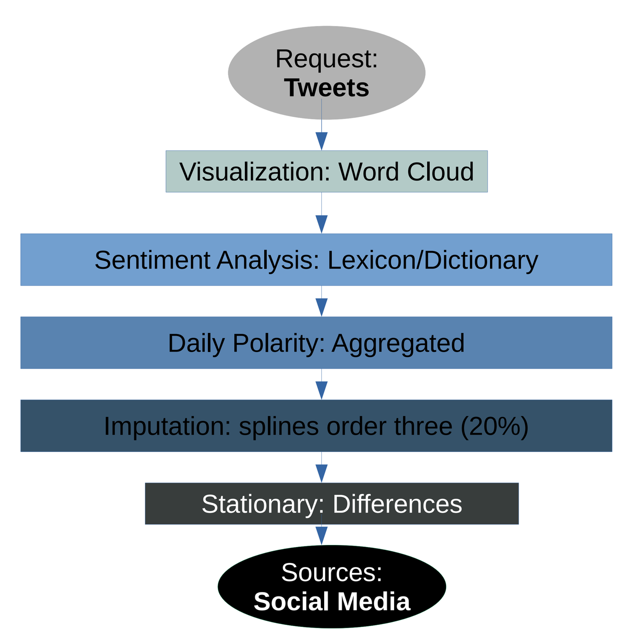
Abstract

Bitcoin has attracted attention from different market participants due to unpredictable price patterns. Sometimes, the price has exhibited big jumps. Bitcoin prices have also had extreme, unexpected crashes. We test the predictive power of a wide range of determinants on bitcoins' price direction under the continuous transfer entropy approach as a feature selection criterion. Accordingly, the statistically significant assets in the sense of permutation test on the nearest neighbour estimation of local transfer entropy are used as features or explanatory variables in a deep learning classification model to predict the price direction of bitcoin. The proposed variable selection methodology excludes the NASDAQ index and Tesla as drivers. Under different scenarios and metrics, the best results are obtained using the significant drivers during the pandemic as validation. In the test, the accuracy increased in the post-pandemic scenario of July 2020 to January 2021 without drivers. In other words, our results indicate that in times of high volatility, Bitcoin seems to self-regulate and does not need additional drivers to improve the accuracy of the price direction [1].

Data

01/Jan/2017 to 09/Jan/2021 at a daily frequency for a total of $n = 1470$ observations.

Type	Variables
Investor attention	Google Trends
Social media	BBC Breaking News
	Department of State
	United Nations
	Elon Musk
	Donald Trump
Twitter-EPU	Twitter-based Uncertainty Index
Risk Aversion	Financial Proxy to Risk Aversion and Economic Uncertainty
Cryptocurrencies	ETH
	LTC
	XRP
	DOGE
	TETHER
Financial indices	Gold
	Silver
	Palladium
	Platinum
	DCOILBRENTEU
	DCOILWTICO
	EUR/USD
	S&P 500
	NASDAQ
	VIX
	ACWI
	Tesla



Transfer Entropy

Transfer Entropy (TE) measures the flow of information from system Y over system X in a non-symmetric way. Let us denote the sequences of states of the system X and Y as follows: $x_i = x(i)$ and $y_i = y(i), i = 1, \dots, N$ Entropy or Information Transfer is defined as [2]:

$$T_{Y\to X}(k,l) = \sum p(x_{i+1}, x_i^{(k)}, y_i^{(l)}) \log \frac{p(x_{i+1}|x_i^{(k)}, y_i^{(l)})}{p(x_{i+1}|x_i^{(k)})}, \quad (1)$$

The idea is to model the time series as a Markovian system and incorporate the temporal dependencies by considering the states x_i and y_i to predict the next state x_{i+1}. The deviation of the generalized Markov property is then measured: $p(x_{i+1}|x_i, y_i) = p(x_{i+1}|x_i)$. If there is no deviation Y has no influence on X.

Local Transfer Entropy

TE metric can be formulated as a global average or expectation value of a local TE at each observation [3]:

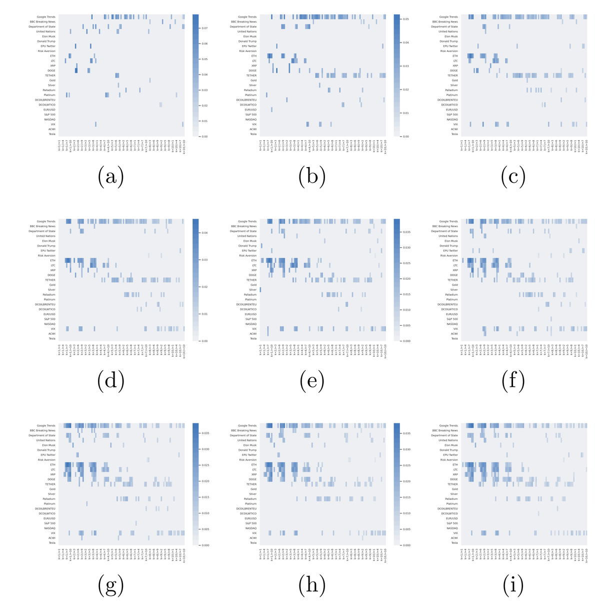
$$\begin{aligned} T_{Y\to X}(k,l) &= \langle t_{Y\to X}(i+1,k,l)\rangle, \\ t_{Y\to X}(i+1,k,l) &= \log \frac{p(x_{i+1}|x_i^{(k)}, y_i^{(l)})}{p(x_{i+1}x_i^{(k)})} \end{aligned} \quad (2)$$

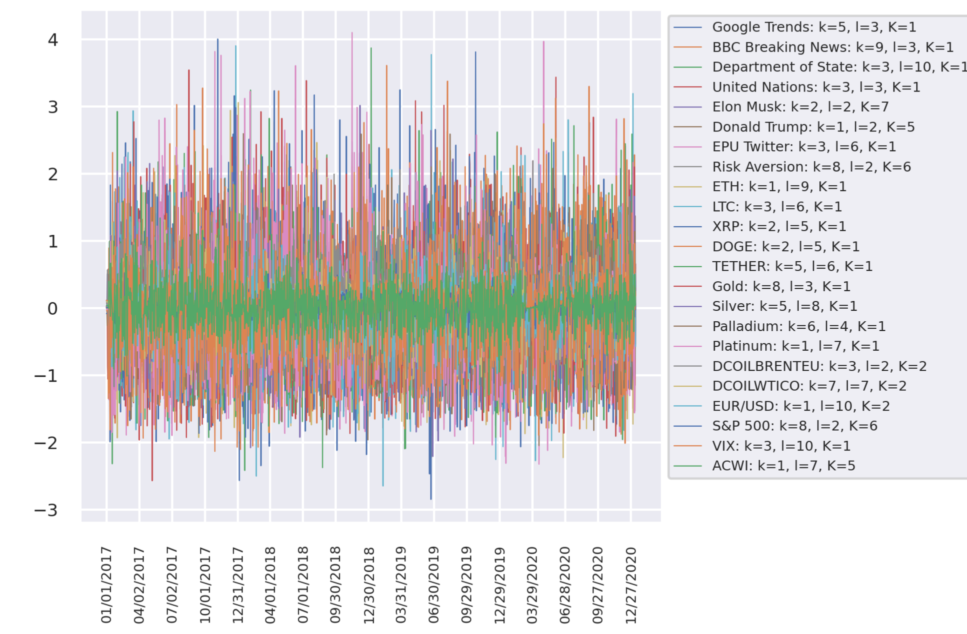
The measure is local in the sense it is defined at each time n for each destination element X in the system and each causal information source Y of the destination. The local TE may be either positive or negative (with the source $y_i^{(l)}$ being either informative or misinformative respectively) for a specific event set $(x_{i+1}, x_i^{(k)}, y_i^{(l)})$

Variable Selection

We applying the local TE from each source to bitcoin using the nearest-neighbor [1] estimation [4]. Here, the Markovian order k, l and neighbor parameter K are varying from 1 to 10, for a total of 1000 different estimations for each driver. The permutation testing is used to measure the statistical significative flow of information. Then, the highest TE on the tuple $\{k, l, K\}$ of each sifgnificative driver is considered further as feature of a deep learning model.

[1] Also known as Kraskov estimation





Deep learning classification models

We can think of artificial neural networks (ANNs) as a mathematical model whose operation is inspired by the activity and interactions between neuronal cells due to their electrochemical signals. The main advantages of ANNs are their non-parametric and nonlinear characteristics. The essential ingredients of an ANN are the neurons that receive an input vector x_i, and through the point product with a vector of weights w, generate an output via the activation function $g(\cdot)$:

$$f(x_i) = g(x_u \cdot w) + b, \quad (3)$$

where b is a trend to be estimated during the training process. The basic procedure is the following. The first layer of neurons or input layer receives each of the elements of the input vector x_i and transmits them to the second (hidden) layer. The next hidden layers calculate their output values or signals and transmit them as an input vector to the next layer until reaching the last layer or output layer, which generates an estimation for an output vector. Particularly, we considered models with several layers also known as deep learning models [5].

Specifications

- Training set: 75% (before pandemic)
- Validation set: 13% (pandemic)
- Test set: 12% (post-pandemic)
- scenarios:

S1 : univariate
S2 : all features
S3 : significative features
S4 : local TE
S5 : significative features + local TE

- architectures:

D1 : Deep LSTM
D2 : Wide LSTM
D3 : Deep Bidirectional LSTM
D4 : Wide Bidirectional LSTM
D5 : CNN

Classification metrics

Design	Case	Acc	AUC	TPR	TNR	PPV	FOR	BA	F1	#
D1	S1	64.30	0.4981	90.99	11.67	67.01	60.38	51.33	77.18	
	S2	56.82	0.4946	74.08	22.78	65.42	69.17	48.43	69.48	
	S3	61.78	0.5188	79.58	26.67	68.15	**60.17**	**53.12**	73.42	2
	S4	51.96	0.4898	60.70	34.72	64.71	69.06	47.71	62.65	
	S5	60.47	0.4842	83.66	14.72	65.93	68.64	49.19	73.74	
D2	S1	60.75	0.4786	85.92	11.11	65.59	71.43	48.51	74.39	
	S2	52.06	0.4870	56.48	43.33	66.28	66.45	49.91	60.99	
	S3	53.46	0.4997	56.76	46.94	67.85	64.50	51.85	61.81	
	S4	55.70	0.4794	70.56	26.39	65.40	68.75	48.48	67.89	
	S5	50.93	0.4806	55.49	41.94	65.34	67.67	48.72	60.02	
D3	S1	**65.05**	0.5072	**95.21**	5.56	66.54	62.96	50.38	**78.33**	3
	S2	55.70	0.5248	63.38	40.56	67.77	64.04	51.97	65.50	
	S3	57.38	0.5176	67.32	37.78	68.09	63.04	52.55	67.71	
	S4	51.40	0.5051	52.96	48.33	66.90	65.75	50.65	59.12	
	S5	54.21	0.5094	60.56	41.67	67.19	65.12	51.12	63.70	
D4	S1	61.21	0.4831	86.48	11.39	65.81	70.07	48.93	74.74	
	S2	48.13	0.4718	48.17	48.06	64.65	68.02	48.11	55.21	
	S3	47.20	0.4771	43.24	**55.00**	65.46	67.05	49.12	52.08	1
	S4	45.98	0.4359	50.99	36.11	61.15	72.80	43.55	55.61	
	S5	51.96	0.4743	55.49	45.00	66.55	66.11	50.25	60.52	
D5	S1	58.04	0.5017	78.59	17.50	65.26	70.70	48.05	71.31	
	S2	55.23	0.4942	62.39	41.11	67.63	64.34	51.75	64.91	
	S3	54.21	0.4994	63.10	36.67	66.27	66.50	49.88	64.65	
	S4	54.49	0.5269	62.39	38.89	66.82	65.60	50.64	64.53	
	S5	55.79	**0.5316**	61.83	43.89	**68.49**	63.17	52.86	64.99	2

Conclusions

- An attention must be paid to the evidence about the order $k = l = 1$ throws values near zero. Practitioners usually assume this scenario under Gaussian estimations. Then, a precaution must be put to the memory parameters of Markov at least when working with the KSG estimation.
- On the other hand, the forecasting of Bitcoin's price direction improves in the validation set, but not for all metrics in the test dataset when including significant drivers or local TE as a feature.
- Two methodological contributions to highlight are the use of nontraditional indicators such as market sentiment, as well as a continuous estimation of the local TE as a tool to determine additional drivers in the classification model.
- Finally, the models presented here are easily adaptable to high-frequency data because they are non-parametric and nonlinear in nature.

Acknowledgements

This research was funded Consejo Nacional de Ciencia y Tecnología (CONACYT, Mexico) through fund FOSEC SEP-INVESTIGACION BASICA (Grant No. A1-S-43514).

References

[1] A. García-Medina and T. L. D. Huynh, "What drives bitcoin? an approach from continuous local transfer entropy and deep learning classification models," *arXiv preprint arXiv:2109.01214*, 2021.

[2] T. Schreiber, "Measuring information transfer," *Physical review letters*, vol. 85, no. 2, pp. 461–464, 2000.

[3] J. T. Lizier, M. Prokopenko, and A. Y. Zomaya, "Local information transfer as a spatiotemporal filter for complex systems," *Phys. Rev. E*, vol. 77, p. 026110, Feb 2008.

[4] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.

[5] J. Brownlee, *Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018.