technische universität
dortmund

Bitcoin and Twitter - How to predict the Bitcoin price using tweets?

Felix Fikowski

May 22th, 2023

TU Dortmund University

Advanced Text Mining Methods WS2022/23

Contents

Motivation

Data

- Bitcoin Data
- Twitter Data

Methods

- Vector Autoregressive Process
- Forecasting

Analysis

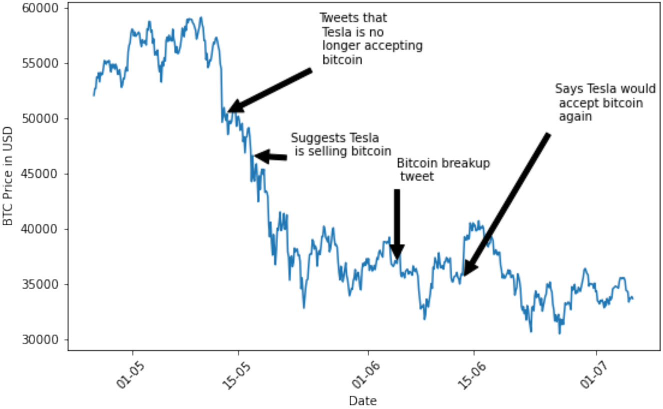
- AR-Process
- VAR-Processes
- Model Comparison

Conclusion and Outlook

Motivation

- Popularity of Crypto currencies and social media are closely related
- Tweets of influential people had a reasonable effect on the price development of Bitcoin
- Outstanding in this regard are the tweets of Elon Musk

Motivation



Musks tweets and the Bitcoin price development in 2021

Question

How can we formalize this observation in a quantitative framework to forecast the Bitcoin price?

Idea

- Apply a sentiment analysis to the tweets
- Exploit the correlation between the sentiments and the Bitcoin price

How to integrate a sentiment analysis in a forecasting model? - A time series approach

1. Get a sufficiently large amount of twitter data
2. Generate the sentiment score to each tweet
3. Group the tweets in the desired frequency and calculate the mean of the sentiment scores
4. Integrate the resulting time series in the used time series model

Bitcoin Data - Source

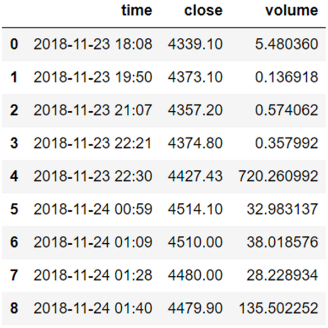
- Kaggle data set of Klein (2023) which contains csv-files to 400+ crypto currency pairs
- We focus on the Bitcoin US Dollar pair
- The data was collected from 2013 till today at 1-minute resolution
- Columns: `time`, `open`, `close`, `high`, `low`, `volume`
- Restrict the time frame to 2018-11-23 to 2019-03-29

Bitcoin Data - Preprocessing

- Focus on the columns `close` and `volume`
  - `close`: the last price at which the bitcoin traded
  - `volume`: refers to the total number of coins exchanged between buyers and sellers
- `close` serve as our dependent variable while the `volume` is used as independent variable

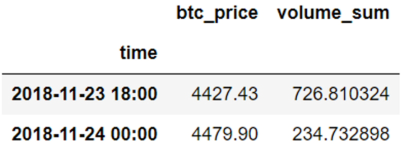
Bitcoin Data - Preprocessing

- Resample the data in a 6 hour frequency
- Take the last closing price and sum the volume
- Example:



	time	close	volume
0	2018-11-23 18:08	4339.10	5.480360
1	2018-11-23 19:50	4373.10	0.136918
2	2018-11-23 21:07	4357.20	0.574062
3	2018-11-23 22:21	4374.80	0.357992
4	2018-11-23 22:30	4427.43	720.260992
5	2018-11-24 00:59	4514.10	32.983137
6	2018-11-24 01:09	4510.00	38.018576
7	2018-11-24 01:28	4480.00	28.228934
8	2018-11-24 01:40	4479.90	135.502252

Original Dataframe

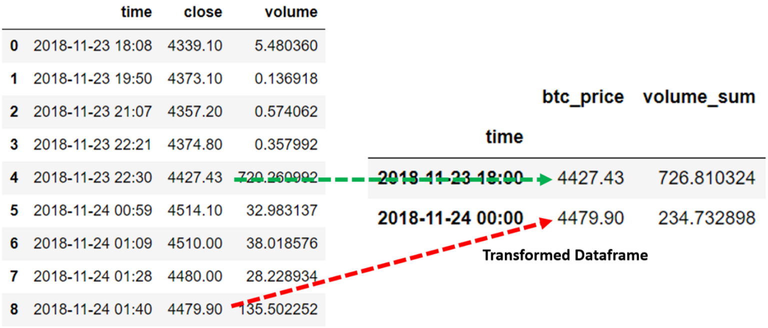


	btc_price	volume_sum
time		
2018-11-23 18:00	4427.43	726.810324
2018-11-24 00:00	4479.90	234.732898

Transformed Dataframe

Bitcoin Data - Preprocessing

- Resample the data in a 6 hour frequency
- Take the last closing price and sum the volume
- Example:



	time	close	volume
0	2018-11-23 18:08	4339.10	5.480360
1	2018-11-23 19:50	4373.10	0.136918
2	2018-11-23 21:07	4357.20	0.574062
3	2018-11-23 22:21	4374.80	0.357992
4	2018-11-23 22:30	4427.43	720.260992
5	2018-11-24 00:59	4514.10	32.983137
6	2018-11-24 01:09	4510.00	38.018576
7	2018-11-24 01:28	4480.00	28.228934
8	2018-11-24 01:40	4479.90	135.502252

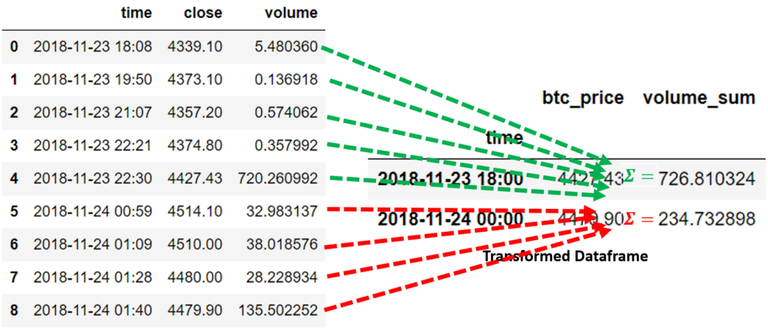
Original Dataframe

	btc_price	volume_sum
time		
2018-11-23 18:00	4427.43	726.810324
2018-11-24 00:00	4479.90	234.732898

Transformed Dataframe

Bitcoin Data - Preprocessing

- Resample the data in a 6 hour frequency
- Take the last closing price and sum the volume
- Example:



	time	close	volume
0	2018-11-23 18:08	4339.10	5.480360
1	2018-11-23 19:50	4373.10	0.136918
2	2018-11-23 21:07	4357.20	0.574062
3	2018-11-23 22:21	4374.80	0.357992
4	2018-11-23 22:30	4427.43	720.260992
5	2018-11-24 00:59	4514.10	32.983137
6	2018-11-24 01:09	4510.00	38.018576
7	2018-11-24 01:28	4480.00	28.228934
8	2018-11-24 01:40	4479.90	135.502252

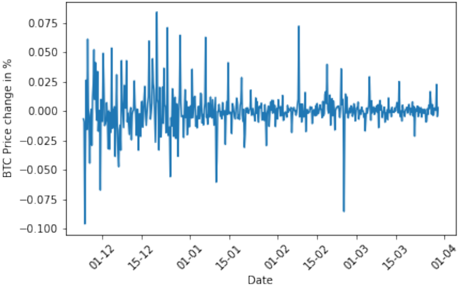
Original Dataframe

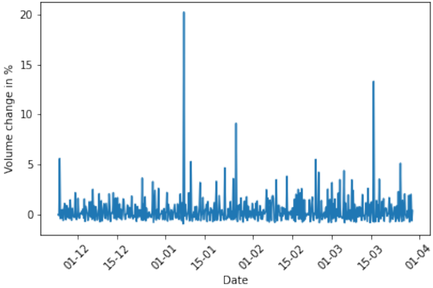
	btc_price	volume_sum
time		
2018-11-23 18:00	4427.43	$\Sigma =$ 726.810324
2018-11-24 00:00	4479.90	$\Sigma =$ 234.732898

Transformed Dataframe

Bitcoin Data

- Calculate the percentage changes for the Bitcoin price and the traded volume
- This results in 503 observations for the variables `btc_price_per` and `volume_sum_per`



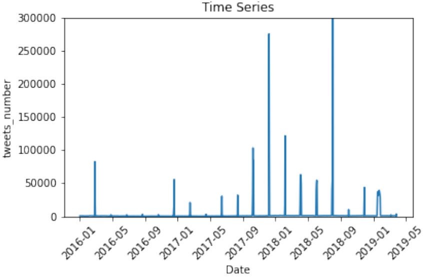


Twitter Data - Source

- Kaggle data set from Bouillet (2021)
- 16M tweets were collected between 2016-01-01 to 2019-03-29
- Tweets contain “Bitcoin” or “BTC”
- Each tweet builds a row
- Columns: `User`, `fullname`, `tweet-id`, `timestamp`, `url`, `likes`, `replies`, `retweets`, `text`

Twitter Data - Source

	tweets per day
mean	3337
std	31352
min	187
median	700
max	996011



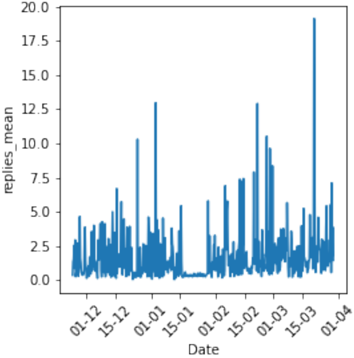
- Days with extreme outlying number of tweets may reflect the relevance on social media or inconsistencies in data procurement
- Assumption: The data set is representative to reflect the sentiment

Twitter Data - Noise Elimination

- Tweets containing the words “Giveaway”, “Cashback”, “Airdrop” and “nft” are seen as spam and get removed
- Usage of a language detection tool to remove non-english tweets
- Restrict the time frame to 2018-11-23 to 2019-03-29
- After the noise elimination and the restriction of the time frame 416,996 tweets remain

Twitter Data - Variable Extraction of Meta Variables

- To reflect twitter meta data, we resample the in a 6 hour frequency `replies` with the mean
- This results in the variable `replies_mean`



Sentiment Analysis

- Two different approaches:
  1. Dictionary based sentiment using Vader
  2. Using a pre-trained Language Model
     - finBERT: Pre-training and fine-tuned BERT model for the financial domain by Araci (2019)
     - TimeLM-19: Pre-training and fine-tuned roBERTa model for text data as tweets by Loureiro (2022)

Vader Sentiment

- Dictionary based sentiment tool designed to work well on social media text and other informal text
- Distinguish between positive (1), negative (-1), and neutral (0) words
- The sentiment score is then computed based on the weighted sum

	sentiment_vader
count	416,996
mean	0.14
std	0.37
min	-0.99
25%	0.00
50%	0.00
75%	0.40
max	0.99

roBERTa - differences to BERT

- RoBERTa (Robustly Optimized BERT Pretrained Approach) is a variation of BERT and was introduced 2019 by Facebook AI
- Main differences to BERT
  - It was trained on a larger and more diverse corpus of text data which also includes web pages
  - RoBERTa was trained for a longer period of time and with a different learning rate schedule
- This improvements have resulted in RoBERTa outperforming BERT on a variety of benchmark NLP tasks

finBERT vs. TimeLM-19

	finBERT	TimeLM-19
Pre-Training	• On a financial corpus of Reuters containing 46,143 documents • With > 29M words and 400K sentences	• On 124M tweets obtained by the Twitter Academic API • evenly distributed across time
Fine-Tuning	• Sentiment analysis data set Financial PhraseBank from Malo et al. (2014) • 4845 financial news sentences	• On SemEval-2017 from Rosenthal et al. (2019) • 60K tweets chosen based on popular events

finBERT vs. TimeLM-19

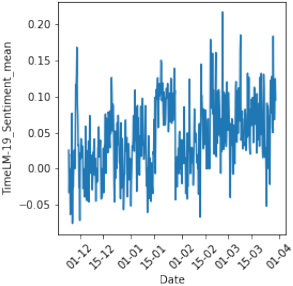
	finBERT	TimeLM-19
Pre-Processing	• Lower casing • Replace @user123 with user • Remove hashtags • Replace links of websites with website • Remove punktuation including emojies	• Replace @user123 with user • Replace links of websites with http • Remove punctuation

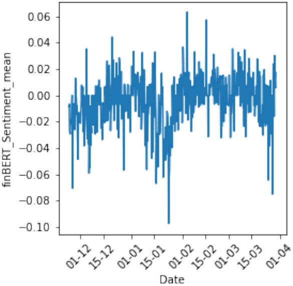
Sentiment Analysis - Comparisons

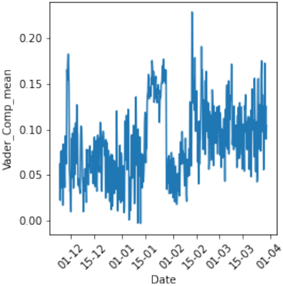
finBERT TimeLM-19	-1	0	1	
negative = -1	10915	28695	215	39825
neutral = 0	15949	274042	9418	299409
positive = 1	715	68255	8792	77762
	27579	370992	18425	416996

Sentiment Analysis - Comparisons

- As a next step we transform the sentiment score data into a time series as we did with the Bitcoin data
- Results in the variables `finBERT_sentiment`, `TimeLM-19_sentiment` and `vader_sentiment`







Autoregressive Process (AR process)

- A time series model in which the value of a variable at time t is a linear function of its past values

Vector Autoregressive Process (VAR Process)

- A multivariate time series model that assumes that each variable in the process depends linearly on the past values of all the variables in the process
- In other words, a VAR process is a collection of multiple AR processes
- Each variable is modeled as a function of its own past values and the past values of all the other variables

VAR Process - Modeling

- According to chapter 2.2 of Kilian and Lütkepohl (2017)
- A VAR process models the K time series variables $y_t = (y_{1t}, ..., y_{Kt})'$

$$y_t = [v, A_1, ..., A_p]Z_{t-1} + u_t \tag{1}$$

  - v: deterministic part and a vector
  - A_i with $i = 1, ..., p$: $K \times K$ matrices containing the coefficients $a_{nm,i}$ with $n, m = 1, ..., K$
  - $Z_{t-1} \equiv (1, y'_{t-1}, ..., y'_{t-p})'$
  - $u_t = (u_{1t}, ..., u_{Kt})'$: zero mean error process
  - p: order of the process
- Estimation of the coefficients via least square estimator

Autoregressive Process

- Autoregressive processes of order p can be seen as a VAR(p) processes with $K = 1$:

$$y_t = v + a_1 y_{t-1} + \dots + a_p y_{t-p} + u_t \tag{2}$$

- The AR model can be estimated through the least-squares method

VAR Process - Forecasting

- In this project we will only focus on one step ahead predictions
- The one-step-ahead prediction is calculated through

$$\hat{y}_t(1) = [\hat{A}_{0|t}, \hat{A}_{1|t}, ..., \hat{A}_{p|t}]Z_t \quad (3)$$

  - $\hat{A}_{0|t}$: estimated constant based on the available observations up to period t
  - $\hat{A}_{i|t}$ with $i \in \{1, ..., p\}$: the estimated coefficients

- Based on the forecasted value the root mean squared forecasting error can be calculated

Root Mean Squared Forecasting Error

$$RMSFE = \sqrt{(T-1)^{-1} \sum_{t=1}^{T-1} (y_{i+1} - \hat{y}_{t+1|t})^2} \tag{4}$$

- Used as a measure for evaluating the performance of a forecasting model

VAR Process - Lag-Order Selection Procedure

- Determine the model order based on Akaike's Information Criterion (AIC) according to chapter 2.6.3 of Kilian and Lütkepohl (2017)
- The AIC is defined as

$$AIC(m) = log(det(\tilde{\Sigma}_u(m))) + \frac{2}{T}(mK^2 + K) \tag{5}$$

  - $\tilde{\Sigma}_u(m) = \frac{1}{T}\sum_{i=1}^{T} \hat{u}_i\hat{u}_i'$: residual covariance matrix estimator of a VAR model
  - m: lag order
  - $\hat{u}_i$: residuals based on the least-square estimator

VAR Process - Lag-Order Selection Procedure

- Trade-off between the goodness of fit and increasing complexity with increasing lag order
- With increasing lag order the second part of the AIC, $\frac{2}{T}(mK^2 + K)$ increases and penalizes for higher lag orders for a given sample size T
- The m for which the AIC is minimal, is the determined lag-order

Analysis

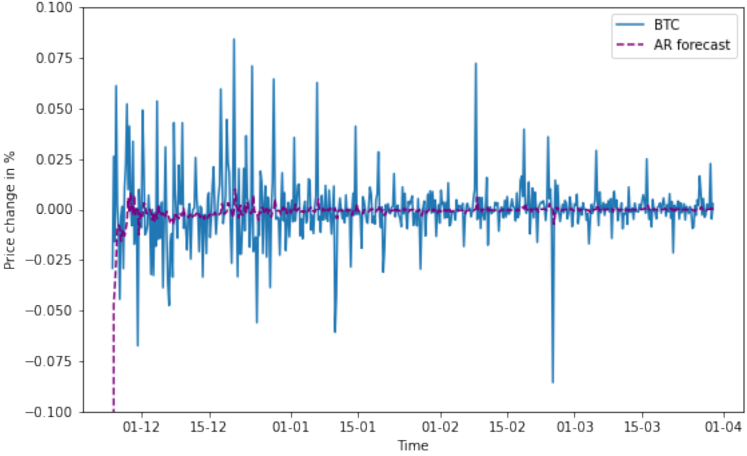
Steps

- A stochastic process is defined to be weakly stationary, if:
  1. AR-Process
  2. VAR-Processes
     2.1 Lag Order Selection
     2.2 Forecasting
  3. Model Comparison

AR-Process

- A 6-hours-ahead forecast for the `btc_price_per` is generated using an AR(1) model
- The first three observations are needed to generate the first model
- Forecasts are calculated for 2018-11-25 00:00 to 2019-03-29 18:00

AR-Process


BTC
AR forecast
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
Price change in %
01-12
15-12
01-01
15-01
01-02
15-02
01-03
15-03
01-04
Time

VAR-Process - Setup

- Reminder: We have the time series variables `btc_price_per`, `volume_per` and `replies_per`
- Plus the sentiment related time series `finBERT_sentiment`, `TimeLM-19_sentiment` and `vader_sentiment`
- Combine each sentiment with the time series `btc_price_per`, volume_per and replies_per to a data set
- This results in three different data sets

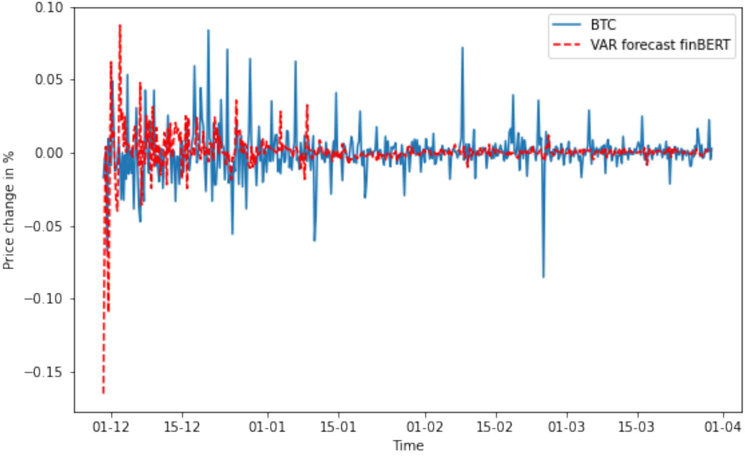
VAR-Process - Lag Order Selection

- For all three data sets the optimal lag-order is determined
- The maximal possible lag-order is set to $p = 5$

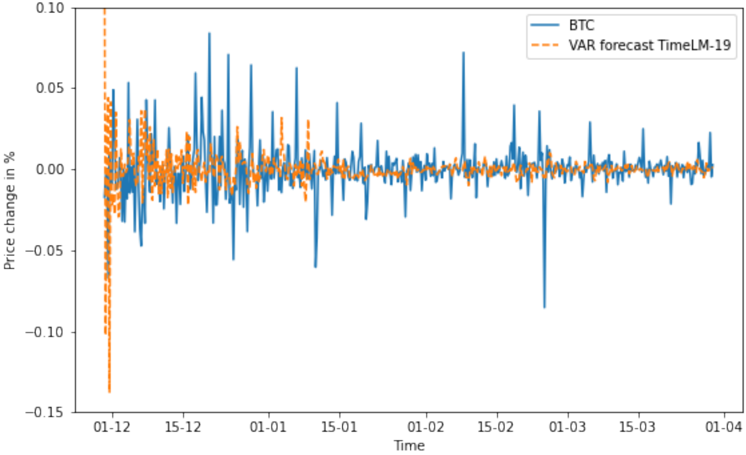
AIC(n)	1	2	3	4	5
finBERT	-14.099	-14.108	-14.084	**-14.115**	-14.09
TimeLM-19	-12.551	-12.579	-12.581	**-12.600**	-12.598
Vader	-12.998	-13.028	-13.045	**-13.065**	-13.062

- The first 21 observations are needed to generate the first model
- Forecasts are calculated for 2018-11-29 12:00 to 2019-03-29 18:00

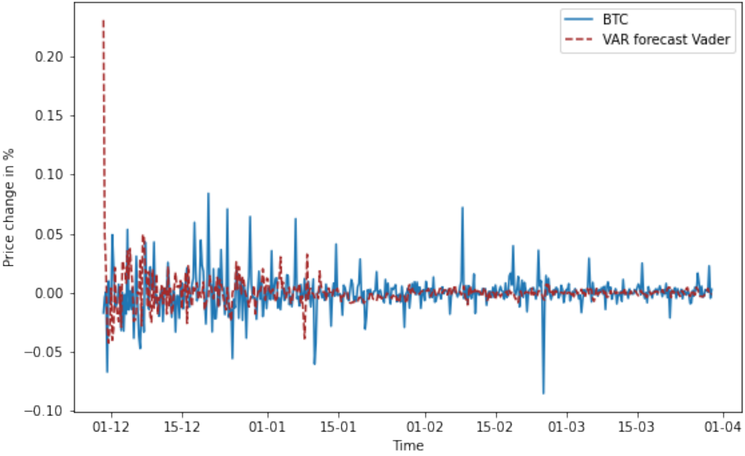
VAR-Process - finBERT



VAR-Process - TimeLM-19



VAR-Process - Vader



Model Comparison

- The VAR(4) models have a clearly smaller RMSFE compared to the AR(1) model
- Among the VAR(4) models the one using the `finBERT_sentiment` has the lowest RMSE
- However, the result should be treated with caution, since it is not clear to what extent it is distorted by the first deviating values

	RMSE
AR(1)	0.138
VAR(4) finBERT	0.0206
VAR(4) TimeLM-19	0.0213
VAR(4) Vader	0.0217

Conclusion

- We have performed a sentiment analysis on tweets which contained #btc or #bitcoin using finBERT, TimeLM-19 and Vader
- The different sentiment tools showed diavating outcomes on certain time frames
- This result should be treated with caution, since the integrity of the Twitter data is not clear
- Further, we used the sentiment data to integrate it in different VAR model forecasts
- We showed that the VAR processes yield a much lower RMSFE compared to a simple AR(1) process
- It is not clear yet to what extent it is distorted by the first deviating values of the forecast

Outlook

- Train a individual Language model for this task
- Include time series which consider other factors like energy prices or the stock market
- Forecast using rolling window approach
- Try different model approach

Literature

Bouillet, Alexandre (2021)
Bitcoin tweets - 16M tweets
https://www.kaggle.com/datasets/alaix14/bitcoin-tweets-20160101-to-20190329

Klein, Carsten (2023)
400+ crypto currency pairs at 1-minute resolution
https://www.kaggle.com/datasets/tencars/392-crypto-currency-pairs-at-minute-resolution?select=btceur.csv

Araci,Dogu (2019)
FinBERT: Financial Sentiment Analysis with Pre-trained Language Models
https://arxiv.org/abs/1908.10063)

Literature

D. Loureiro and F. Barbieri and L. Neves and L. E. Anke and J. Camacho-Collados (2022)
TimeLMs: Diachronic Language Models from Twitter
https://arxiv.org/abs/2202.03829)

S. Rosenthal, N. Farra, and P. Nakov (2019)
Semeval-2017 task 4: Sentiment analysis in twitter
https://arxiv.org/abs/1912.00741)

L. Kilian and H. Lütkepohl (2017)
Structural Vector Autoregressive Analysis
https://doi.org/10.1017/9781108164818

J. D. Cryer and K. Chan (2008)
Time Series Analysis
https://doi.org/10.1007/978-0-387-75959-3

Literature

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala (2008)
Time Series Analysis
https://doi.org/10.1007/978-0-387-75959-3

Thanks for your Attention!