Regional Commodities Price Volatility Assessment Using Self-Driven Recurrent Networks*

Pablo Negri[1,3], Maria Priscila Ramos[2,4], and Martin Breitkopf[4]

[1] Departamento de Computacion, UBA-FCEyN, Argentina
[2] Departamento de Economia, UBA-FCE, Argentina
[3] Instituto de Investigacion en Ciencias de la Computacion (ICC), CONICET-UBA
[4] Instituto Interdisciplinario de Economia Politica (IIEP), CONICET-UBA

Abstract. The high volatility of the agricultural and energy commodity prices in the international market is a concern due to their transmission to regional prices, increasing instability in domestic markets. This paper evaluates the performance of recurrent networks (RNN and LSTM) to predict regional prices reactions under international shock simulations. Experiments are run to soybean and corn regional prices in Argentine by considering exogenous changes of the international oil price - both agricultural commodities are inputs for biofuels' production - and also of their international prices. Results are in line with the econometric literature and consistent with the dynamic of regional prices in Argentina's markets. Thus, the RNNs could be a useful tool for timely economic policy decisions that cushion external price shocks in domestic markets.

Keywords: Recurrent Neural Networks · Regional Commodities Prices · Shock Simulations.

1 Introduction

The definition of new trade policy instruments for monitoring and stabilizing agricultural commodities prices at borders must meet specific domestic socio-economic objectives. Thus, it is essential to understand how changes in international and internal prices propagate geographically within a country. Without an accurate measurement of these effects, any quantitative analysis would be flawed, and the calibration of contingency measures distorted. For example, assuming perfect price transmission would be a risky simplification and would lead to an overestimation of the corrective power of trade policy instruments (e.g. export duties or subsidies).

The literature on price volatility focuses mainly on the cases of large exporters (e.g. United States) and more recently on the case of countries with a high food dependence on agricultural imports (e.g. Sub-Saharan African countries). The related economic and econometric literature evidences the inter-dependencies

* This work was funded by the University of Buenos Aires PDE Nro. 24-2019.

between the different agricultural products [14,8,13,12], and between agricultural and energy markets [15,8], explaining the dynamics of price volatility between markets. Most of these works use GARCH or MGARCH models [15,8] to assess agricultural price volatility as a function of its history. The first one captures the effects on short-term but also long-term price volatility between markets, and the second analyzes the interdependence between them (e.g. spillover effects).

While econometric methods of Vector Auto-regressive (VAR) models remain as the benchmark for price forecast, many research works are pointing to neural networks as a more precise method. Wang et al. [17] use a Back Propagation Neural Network (BPNN) to predict prices of agricultural commodities such as wheat, soy, or corn, and conclude that their predictions are more accurate than an econometric method used for comparison. Fang et al. [7] arrive at similar conclusions using a traditional Neural Network (NN).

Most of the existing research uses NN static models to predict future prices; however, they only use the state of the network in one period to predict values for the next, losing all memory of the network for the next step [11]. For time series, where each value is related to previous and next values, using static models does not properly capture the dynamics. This is particularly true for series with sudden movements or "shocks" , where predictions for static models tend to detach rapidly from real values. Conversely, a dynamic model could accurately learn from shocks and consider their information for prediction.

Recurrent Neural Networks (RNN) are a potential accurate prediction model for agricultural prices. RNNs are Neural Networks that link actual variables on their prior states, giving them a "dynamic memory" [6]. This is extremely useful to predict within a time series, where each element fed to the model is related to the previous and next values. Wang [20] uses an Echo State RNN to predict stock prices from the S&P 500, while Boyko et al. [4] use Long-Short Term Memory (LSTM), to predict upon the same database. Both papers arrive at satisfying conclusions. Moreover, Wang and Wang [18] use an Elman RNN, similar to the one used in our experiments, with a successful prediction to estimate future oil price. It is worth noting that data harmonization before applying any Machine or Deep Learning method can improve these RNN performance [19,7,17]. Furthermore, this RNN literature makes predictions based only on one single input (i.e., time lags of the same price). Nevertheless, a dynamic network could learn and forecast based also on other elements (e.g., international oil price) strongly related to the variable target.

This work implements RNN and LSTM architectures to simulate the dynamics of a closed system of prices (i.e., international prices of oil, soybean, and corn and Argentina's regional -Bahia Blanca, Rosario and Quequen - prices the same agricultural products). We focus on the training and evaluation of these models to estimate inter-dependencies between the inputs, and predict the dynamics of the regional prices. In our experiments, each international commodity is stressed under a strong shock (i.e. international price of oil), and the evolution of the regional prices on each recurrent model is evaluated as a self-driven dynamic. Recurrent models' results show good performance compared to econometric analysis, validating the use of the RNN and LSTM as a realistic engine for this application.

The paper is organized as follows. The next section states the problem, depicts the recurrent models, and details the training procedure. Experiments and analysis are detailed in section 3. Section 4 concludes the paper and propose future works.

2 Commodities Prices Prediction Models

2.1 Problem Formulation

The prediction models will work with temporal sequences corresponding to commodities prices. We define three kinds of series:

- $\mathbf{e}^{(t)}$ an exogenous price sequence dependent to $\mathbf{i}^{(t)}$.
- $\mathbf{i}^{(t)}$ a price sequence that it is related with $\mathbf{e}^{(t)}$.
- $\mathbf{r}^{(t)}$ a price sequence dependent to $\mathbf{i}^{(t)}$ and $\mathbf{e}^{(t)}$.

were (t) indicates the value of the price at time t. In our experiments, $\mathbf{e}^{(t)}$ is the international price of oil. The sequences $\mathbf{i}^{(t)}$ are international prices of agricultural commodities associated with bio-diesel (soybean) and bio-ethanol (corn). Because these bio-fuels (partially) replace gasoline, we can state that $\mathbf{e}^{(t)}$ and $\mathbf{i}^{(t)}$ are interdependent variables. Finally, $\mathbf{r}^{(t)}$ corresponds to agricultural commodities prices in different regions of Argentina. The dynamic of these prices involves local factors, and (what we expect to prove) external ones such as the $\mathbf{i}^{(t)}$ sequences.

The model, which simulates the behavior of the closed price system, could capture variables' inter-dependencies from the data at the learning process. This dynamic can be evaluated using *shocks*. A shock is an abrupt change in the price of one of the products in the system that could affect other products' prices. For instance, we are interested in evaluating prices' inter-dependence when applying an oil price shock. This kind of behavior happens in real life, due to political changes, wars, pandemics, and more lastly, environmental concerns.

We choose the RNN model to learn the dynamics of the closed system and predict the stationary values after the shock. Static models could not produce this kind of results as it is needed a system that receives as inputs their precedent outputs. The next section introduces the RNN models.

2.2 Recurrent Neural Network Architecture

Temporal series denoted as $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(T)})$ are usually the inputs of RNN models. In our case, $\mathbf{x}^{(t)}$ is a vector containing the commodity prices at week t including prices data from the three series $(\mathbf{e}^{(t)}, \mathbf{i}^{(t)}, \mathbf{r}^{(t)})$. Equivalently, the target sequences corresponding to the expected commodity prices is stated as $(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(T)})$. The predictions produced by the recurrent model are denoted as $\hat{\mathbf{y}}^{(t)}$.

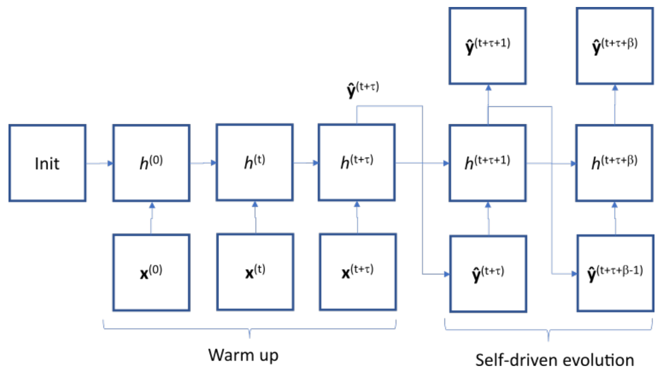

Fig. 1. System architecture and evolution.

The forward pass of a simple recurrent network model [11] introduces $h^{(t)}$, the hidden state of the network at time t and is defined by two equations:

$$\mathbf{h}^{(t)} = \sigma(W^{hx}\mathbf{x}^{(t)} + W^{hh}h^{(t-1)} + b_h) \tag{1}$$

$$\hat{\mathbf{y}}^{(t)} = \sigma(W^{yh}h^{(t)} + b_y) \tag{2}$$

Eq. 1 obtains $h^{(t)}$ as the combination of the input $\mathbf{x}^{(t)}$ at time t and $h^{(t-1)}$, which corresponds to the hidden previous state. These recurrent connections are what give the model memory [6]. We express the estimation of target $\mathbf{y}$ of equations 1 and 2 at time t as a dependent function R with internal parameters $\{W^{hx}, W^{hh}, W^{yh}, b_h\}$:

$$\hat{\mathbf{y}}^{(t)} = R(\mathbf{x}^{(t)}|h = h^{(t-1)}) \tag{3}$$

Modern RNN architectures introduce several improvements overcoming traditional training problems. Long-Short Term Memory model [9] (LSTM) is one of the most successful networks widely employed on several applications, such as natural language processing. LSTM deals with long-term dependencies incorporating gates to the recurrent cell.

This work implements recurrent neural networks with both RNN-Elman and LSTM cells with a forget gate. Also, we'll deploy a stacked RNN and LSTM network [21]. In practice, an easy way to increase the depth of the recurrent network is to stack the cells into L layers. This architecture has proved to improve efficiency and performance in problems like vehicle-to-vehicle communication [5] and French-English translation [16].

2.3 Training Procedure

The training follows a mini-sequences batch procedure. We split the training sequence into mini-sequences of τ length $(\mathbf{x}^{(t)}, ..., \mathbf{x}^{(t+\tau-1)})$, referred as $\mathbf{X}^{(t,\tau)}$. The target $\mathbf{Y}^{(t,\beta)}$ is also a sequence that consists of the price values of interest from τ to β: $(\mathbf{x}^{(t+\tau)}, ..., \mathbf{x}^{(t+\tau+\beta-1)})$. They are the "future" prices that the model should predict in a self-driven way. More precisely, the inputs always correspond to all the agricultural prices $\mathbf{x}^{(t)} = (\mathbf{e}^{(t)}, \mathbf{i}^{(t)}, \mathbf{r}^{(t)})$, while outputs are subject to which variable receives the exogenous shock. For example, if the shock is applied on the international oil price, the output becomes $\mathbf{y}^{(t)} = (\mathbf{i}^{(t)}, \mathbf{r}^{(t)})$. If another variable is selected to be shocked, it should be excluded from the target. The τ inputs feed the RNN model, updating the internal hidden states. This step can be thought as a *warm-up* of the internal variables from a (always the same) initial value. Then, for the next β time steps, equation 3 is modified by:

$$\hat{\mathbf{y}}^{(t)} = R(\hat{\mathbf{y}}^{(t-1)}|h = h^{(t-1)}) \tag{4}$$

this outputs are then reserved as sequence target $\hat{\mathbf{Y}}^{(t,\beta)} = (\hat{\mathbf{y}}^{(t+\tau)}, ..., \hat{\mathbf{y}}^{(t+\tau+\beta-1)})$. The loss function is defined as a mean squared error on the output sequence $\hat{\mathbf{Y}}$:

$$\mathcal{L} = \sum_{\beta} \frac{1}{\beta} ||\hat{\mathbf{Y}} - \mathbf{Y}|| \tag{5}$$

3 Experiments

3.1 Data

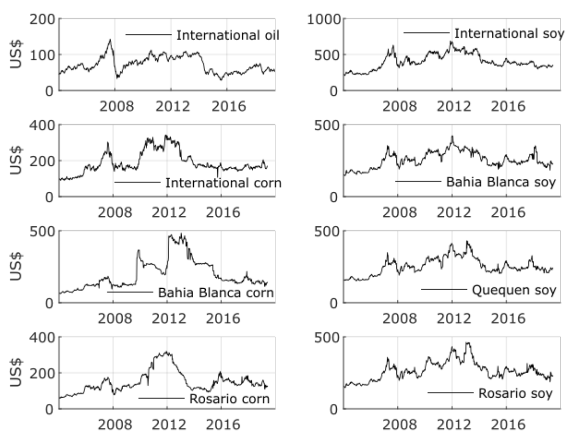

Fig. 2. International and Regional Commodities Prices data series from 2005 to 2019.

We have built a database of weekly prices in US dollars between January 2005 and August 2019, leading to a sample of 772 observations for each price.

Prices considered in the database are: Soybean and corn prices per ton in three regional markets in Argentina (Bahia Blanca - BB, Rosario - Ros, Quequén, QQ, the latter only for soybean) from GRANAR[2]; Soybean and corn international prices per ton from FAOSTAT[1]; Oil international price per barrel from the Western Texas Intermediate, WTI.

Before testing the RNN models, we have analysed the data in order to evaluate the presence of a stable long-term relationship between regional, international prices of each agricultural commodity and the international price of oil. We follow the Johansen's approach [10] for an appropriate cointegration analysis, so we evaluate multivariate stationarity of price variables in each system (i.e., each system is composed of regional and international prices of one of the agricultural commodities and the international oil price)[10,3]. Soybean markets (regional and international) and international oil market are integrated, being the Vector Error Correction (VEC) model most appropriate for the regional soybean price estimation. Cointegration between corn prices (regional and international) and oil is not verified, being regional corn prices estimated through a VAR model. Impulse-Response functions have been run in both regional price systems by shocking (own commodity and oil) international prices to know the convergence path for regional prices. These econometric estimations provide a reference for regional price behaviors under the recurrent network architectures.

3.2 Hyperparameters selection

Four recurrent architectures are implemented: RNN-1c, RNN-2c, LSTM-1c and LSTM-2c. Two of them consist of a single RNN and an LSTM cell. The hidden states h for RNN and (h, c) for LSTM, have H hidden units. The other architectures stack a second recurrent cell to the network with the same number of hidden units H.

We run a K-fold cross validation training, with $K = 5$, using the following set of values for $H = [4, 8, 12, 16, 20, 24, 28, 32]$. Moreover, the training is controlled by τ (*warm-up*) and β (*self-driven*) variables. Thus, the set of values for each variable are $\tau = [6, 7, 8, 9, 10]$ and $\beta = [1, 2, 3, 4]$. Note that $\beta = 1$ corresponds to a classical single prediction of the $t + 1$ output value, while $\beta > 1$ applies the loss function of eq. 5 to a sequence of targets.

Each K-fold is evaluated by two means squared error indices on the target prices of the validation split: a $MSE^{(t+1)}$ prediction, and a $MSE^{(t+N)}$ prediction. Let be $\mathbf{x}^{(t)}$ the model input, $MSE^{(t+1)}$ is computed by the mean squared error between $\hat{y}^{(t)}$ and $\mathbf{x}^{(t+1)}$. $MSE^{(t+N)}$ is obtained by using eq. 4 for a *self-driven* estimation for N steps. Then, the error is computed between prediction $\hat{y}^{(t+N-1)}$ and $\mathbf{x}^{(t+N)}$, and measures how well the recurrent model adjusts the self-driven dynamic after N steps to the real values. In this work, we fix $N = 4$ which means a month of self-driven evolution. We employ an SGD optimizer with an initial learning rate of $1e - 2$. After 20 epochs, the learning rate is reduced by half. Table 1 shows the best results of each architecture sorted by the $MSE^{(t+N)}$ index. As can be seen, recurrent cells with a high number of hidden units H get the lowest errors. In the case of τ, *warm-up* phase seems more important for RNN cells. LSTM cells incorporate additional gates, then, this is a normal conclusion. This is expected for models like LSTM having several gates to remember/forget input data. Increasing τ also increases the temporal drift of the system itself. In the case of β parameter, the best results for RNN are obtained using values greater than one. On the other hand, LSTM prefers lower values of β.

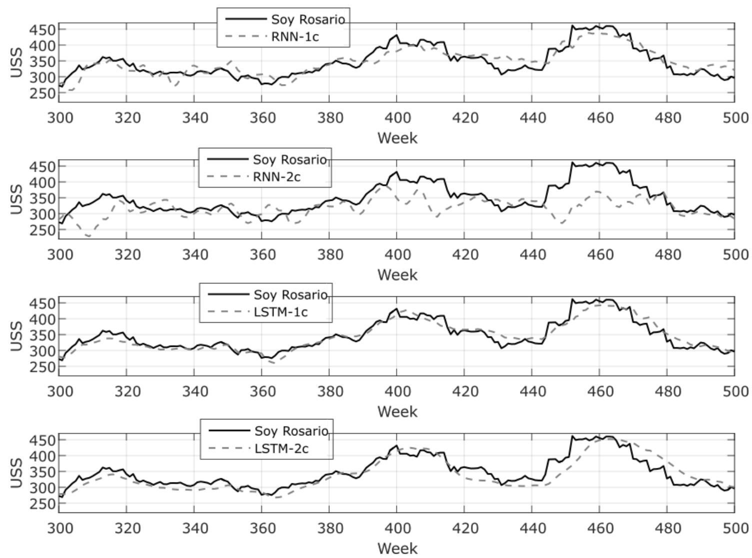

Fig. 3. System architecture and evolution.

Fig. 3 samples the $t+1$ predictions of the four models on a portion of the soybean times series prices from Rosario port. We can appreciate different behaviors for each model. RNN-1c model predicts the series values with a low error but a rapid dynamic. RNN-2c, on the other hand, seems to have a sinusoidal dynamic near the series values, but sometimes the error is high, which is consistent with the high value of their $MSE^{(t+1)}$ index on table 1. LSTM-1c and LSTM-2c predict accurately the average of the series values but have a very low dynamic. This soothing effect is more remarkable on the LSTM-2c predictions.

Architecture	H	τ	β	$MSE^{(t+1)}$	$MSE^{(t+N)}$
RNN-1c	32	10	2	0.159±0.093	0.195±0.156
RNN-2c	32	10	4	0.426±0.182	0.207±0.156
LSTM-1c	32	8	2	0.124±0.083	0.247±0.135
LSTM-2c	32	6	1	0.196±0.182	0.288±0.214

Table 1. Hyperparameters with the best results of the K-Fold Cross Validation.

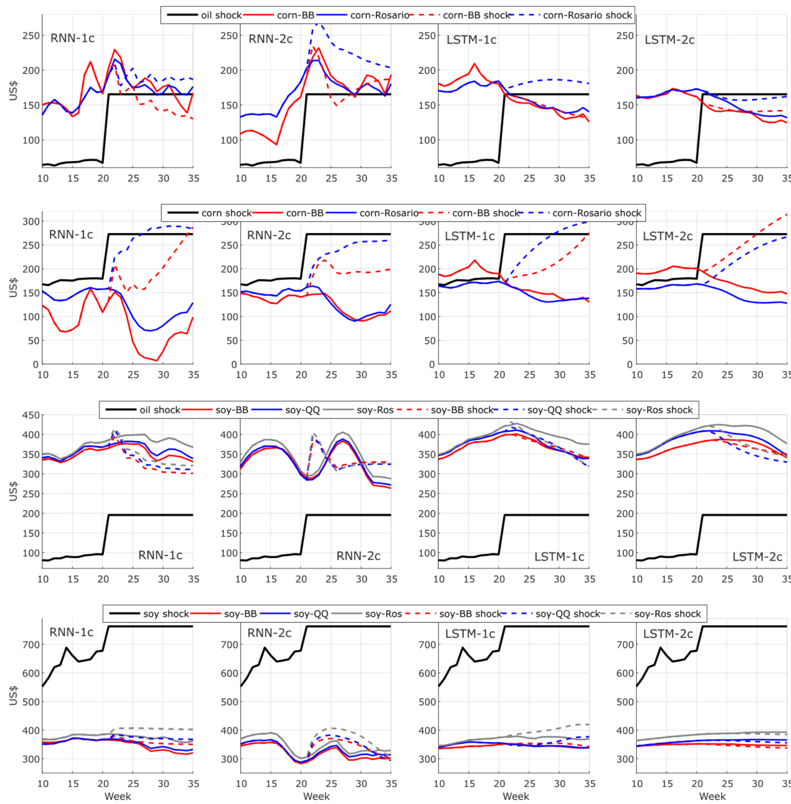

Fig. 4. Shock prediction results.

3.3 International shock simulation

The experiments seek to validate the self-driven evolution of the recurrent networks when a permanent exogenous change (an increase of 100 US$) is introduced in each of international prices (own commodity and oil).

The tests are conducted as follows. For example, to test soybean exogenous change shock, we train the four models with all the commodities prices as inputs and a target that does not predict international soybean. Thus, we split the data sequences into temporal frames of $T = 35$ weeks. The first $\tau = 20$ weeks are employed as *warm-up*, and at $t = 20$, the value of the international soybean price is increased by 100 US$, keeping this value until the end of the test. At this point, the system uses both the new value of the international soybean price and the self-prediction of other prices as input.

For example, in Fig. 4, we depict in black line the international variable we employ to perform the shock and in colors (red, blue, green) the evolution of the regional prices. In solid lines, the picture draws the regional prices without the shock, and the dashed lines depict the self-driven dynamic of the system.

The chosen time-frames in Fig. 4 are in line with the regional prices behaviors in the Impulse-Response function based on econometric models considered as reference. It is worth mentioning that we need to train a different model each time we change the international price to perform the shock.

According to the results in Fig. 4, when considering an exogenous increase of the international oil price, soybean prices in regional markets of Argentina are immediately impacted, but the reaction depends on the model considered, e.g. the RNN-2c displays greater volatility. Nevertheless, the decreasing convergence paths of all models (consistent with econometric estimations) lead to the same new stationary state.

While the regional soybean prices in Argentina recover stability near to the path without shock, the regional corn prices show greater volatility facing the same exogenous shock. Except for the LSTM-2c, regional corn prices display a great difficulty to recover the path without shock, and Bahia Blanca and Rosario corn markets show different behaviors between them and across models. Their different paths of convergence increase the price-gap between regions (supported by the econometric estimations).

Finally, when assuming an exogenous increase in the international price of their agricultural commodity, regional markets prices display greater positive reactions (particularly for corn) and convergence towards higher values compared to their values without shock. Regional soybean prices converge to a higher price in the new stationary state, except under the LSTM-2c, which brings the price back to the path without shock. Reactions of regional corn prices to their international price increase are greater than in the case of soybean and tend to converge close to the new level of the international price of corn.

The difference between the reactions of soybean and corn regional prices to their own international prices is due to Argentina's soybean and corn markets particularities. These results are in line with the role of Argentina as a big soybean producer in the international market, so it is considered as a price maker. Conversely, in the international corn market Argentina is a relatively small player being a price-taker, so a change in the international price of corn is strongly transmitted to regional prices.

4 Conclusions

In this paper, we have trained four recurrent networks to forecast the reaction of regional commodity prices when an exogenous variable (i.e., an international price) is shocked. Results have been validated since they are in line with estimations from econometric auto-regressive models. The self-driven dynamic of recurrent networks has been demonstrated to be consistent with the behavior of Argentina's soybean and corn markets. To reduce regional price volatility, RNNs become a new tool to predict domestic prices' reactions to international changes and provide relevant insights for policy-makers decisions.

Further works should consider more complex recurrent networks, including other variables related to these agricultural and energy prices (e.g., bio-ethanol and bio-diesel prices) and also other regional variables that condition regional price path-through (e.g., transport costs).

References

1. Food and agriculture organization, http://www.fao.org/faostat/en/
2. Granar, http://www.granar.com.ar/
3. Ahumada, H., Cornejo, M.: Long-run effects of commodity prices on the real exchange rate: evidence from argentina. Económica **61** (2015)
4. Boyko, N., Ivanets, A., Bosik, M.: Forecasting economic and financial indicators by supply of deep and recovery neural networks. ECONTECHMOD: An International Quarterly Journal on Economics of Technology and Modelling Processes **7** (2018)
5. Du, X., Zhang, H., Van Nguyen, H., Han, Z.: Stacked lstm deep learning model for traffic prediction in vehicle-to-vehicle communication. In: 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall). pp. 1–5. IEEE (2017)
6. Elman, J.L.: Finding structure in time. Cognitive science **14**(2), 179–211 (1990)
7. Fang, Y., Guan, B., Wu, S., Heravi, S.: Optimal forecast combination based on ensemble empirical mode decomposition for agricultural commodity futures prices. Journal of Forecasting **39**(6), 877–886 (2020)
8. Gardebroek, C., Hernandez, M.A., Robles, M.: Market interdependence and volatility transmission among major crops. Agricultural Economics **47**(2), 141–155 (2016)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Johansen, S., et al.: Likelihood-based inference in cointegrated vector autoregressive models. Oxford University Press on Demand (1995)
11. Lipton, Z.C., Berkowitz, J., Elkan, C.: A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019 (2015)
12. Minot, N.: Food price volatility in sub-saharan africa: Has it really increased? Food Policy **45**, 45–56 (2014)
13. Minot, N., et al.: Transmission of world food price changes to markets in Sub-Saharan Africa. Citeseer (2010)
14. Pietola, K., Liu, X., Robles, M., et al.: Price, inventories, and volatility in the global wheat market. Tech. rep., Int. Food Policy Research Institute (IFPRI) (2010)
15. Serra, T., Gil, J.: Price volatility in food markets: can stock building mitigate price fluctuations? European Review of Agricultural Economics **40**(3), 507–528 (2013)
16. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215 (2014)
17. Wang, D., Yue, C., Wei, S., Lv, J.: Performance analysis of four decomposition-ensemble models for one-day-ahead agricultural commodity futures price forecasting. Algorithms **10**(3), 108 (2017)
18. Wang, J., Wang, J.: Forecasting energy market indices with recurrent neural networks: Case study of crude oil price fluctuations. Energy **102**, 365–374 (2016)
19. Wang, J., Li, X.: A combined neural network model for commodity price forecasting with ssa. Soft Computing **22**(16), 5323–5333 (2018)
20. Wang, Y.: Applications of recurrent neural network on financial time series. MCs thesis, Imperial College London (2017)
21. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural computation **31**(7), 1235–1270 (2019)