Received: 21 August 2022 | Revised: 19 January 2024 | Accepted: 26 February 2024

DOI: 10.1111/agec.12828

AGRICULTURAL ECONOMICS
The Journal of the International Association of Agricultural Economists
IAAE
WILEY

ORIGINAL ARTICLE

Machine learning to predict grains futures prices

Paolo Libenzio Brignoli[1] | **Alessandro Varacca**[2] | **Cornelis Gardebroek**[1] | **Paolo Sckokai**[3]

[1]Agricultural Economics and Rural Policy Group, Wageningen University, Wageningen, The Netherlands

[2]Department of Economics and Social Sciences - DISES, Università Cattolica del Sacro Cuore, Piacenza, Italy

[3]Department of Agricultural and Food Economics, Università Cattolica del Sacro Cuore, Piacenza, Italy

Correspondence
Alessandro Varacca, Department of Economics and Social Sciences - DISES, Università Cattolica del Sacro Cuore, Piacenza, Italy.
Email: alessandro.varacca@unicatt.it

Abstract

Accurate commodity price forecasts are crucial for stakeholders in agricultural supply chains. They support informed marketing decisions, risk management, and investment strategies. Machine learning methods have significant potential to provide accurate forecasts by maximizing out-of-sample accuracy. However, their inherent complexity makes it challenging to understand the appropriate data pre-processing steps to ensure proper functionality. This study compares the forecasting performance of Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs) with classical econometric time series models for corn futures prices. The study considers various combinations of data pre-processing techniques, variable clusters, and forecast horizons. Our results indicate that LSTM-RNNs consistently outperform classical methods, particularly for longer forecast horizons. In particular, our findings demonstrate that LSTM-RNNs are capable of automatically handling structural breaks, resulting in more accurate forecasts when trained on datasets that include such shocks. However, in our setting, LSTM-RNNs struggle to deal with seasonality and trend components, necessitating specific data pre-processing procedures for their removal.

KEYWORDS
agricultural futures prices, forecasting, machine learning, recurrent neural networks, time series

JEL CLASSIFICATION
C22, C53, Q13

1 | INTRODUCTION

Agricultural commodity price forecasting provides crucial information for stakeholders along the food supply chain, who are often faced with large price fluctuations and need to make decisions under risky conditions (Brandt & Bessler, 1981; Colino & Irwin, 2010; Colino et al., 2011). Access to reliable price forecasts assists farmers in making marketing decisions, facilitates the formulation of contracts between different operators and serves as a risk management strategy (Carter & Mohapatra, 2008; Hoffman et al., 2015; OECD, 2000). In particular, futures prices play an important role in providing price information ahead of time to acquaint financial investment decisions (Colino & Irwin, 2010; Zhao, 2021), and reduce market uncertainty (Wang et al., 2017; Xu, 2020). With regard to the latter, these instruments play a key role in crop insurance programs, where futures prices are typically used to define first stage and harvest prices (Ouyang et al., 2019).

In this context, the increasing availability of data, combined with remarkable advancements in computational power, has opened numerous possibilities for the application of Machine Learning (ML) techniques in agricultural economics, including futures price forecasting (Storm

et al., 2020). The ML domain encompasses a broad spectrum of predictive algorithms and modeling techniques, ranging from decision trees to neural networks, ensemble methods, and various other approaches. These methods are better suited than classical statistical tools for managing (very) large datasets and representing complex nonlinear relationships among variables. Moreover, they are designed and fine-tuned with the sole aim of attaining high predictive accuracy. It is therefore unsurprising to find several promising applications where out-of-sample ML forecasts outperform those from more traditional methods (Bojer & Meldgaard, 2021; Hyndman, 2020).

In recent years, the literature on time series forecasting using ML methods has provided valuable insights into their potential contribution. Focusing on the primary sector, recent contributions show ML producing encouraging results in forecasting agricultural (futures) prices (Mahto et al., 2021; Ouyang et al., 2019; Xu & Zhang, 2021a, 2022; Zhao, 2021). Specifically, NNs have proven useful in forecasting agricultural commodity prices over long-term periods, often outperforming linear models by effectively capturing complex nonlinear relationships inherent in the data (Bayona-Oré et al., 2021; Xu & Zhang, 2022). Furthermore, the application of deep learning models for time-dependent data has shown promising results in accurately capturing the directional movements of agricultural spot prices (RL & Mishra, 2021).

Although these results would suggest switching over to ML techniques in forecasting practice, it is often unclear how these methods can yield such accurate results, especially in presence of time-dependent features. This characteristic is sometimes called the "black box" nature of ML algorithms and it refers to their complex architecture, designed to better approximate non-linear functional forms (Mullainathan & Spiess, 2017; Sheikh & Jahirabadkar, 2018; Storm et al., 2020). Since in forecasting tasks the focus is mainly on prediction accuracy, with limited interest in understanding the underlying model, this seems less of a problem (Chernozukov et al., 2017). Nevertheless, producing accurate forecasts typically hinges on the modeler's capacity to effectively address various time series features such as non-stationarity, structural breaks, seasonality, and trends, as well as theoretical constraints such as cointegration. Such determinants of time series persistence, or the lack thereof, play a pivotal role not only in the measurable quality of the final methods but also in their ability to extrapolate. To this extent, data pre-processing represents one of the most important stages along the input-to-prediction modeling chain.

In this paper, we investigate how ML methods perform in agricultural futures price forecasting, compared to classic econometric approaches. Since these series are known to entail several problematic dependence structures such as seasonality patterns, trends, unit-roots structural breaks and potentially non-linear dynamics (Brooks et al., 2013; Chen et al., 2022; Wei & Leuthold, 2000), our goal is to understand to what extent different combinations of data pre-processing procedures and modern ML algorithms yield better forecasts than univariate and multivariate econometric time series models. Since we have a reasonable understanding of how the latter generate forecasts depending on data persistence, we believe that comparing these transparent and interpretable techniques to the less transparent ML models will help us understand under what circumstances the latter are preferable and why. Although futures price forecasting is equally important for many potential applications, our work focuses on mid and short-term predictions, thereby appealing more to financial operators seeking to address market uncertainty when planning commodity purchases, evaluating investment opportunities, or stipulating insurance contracts.

In practice, the comparison between ML and classical time series econometrics considers the degree to which they diverge and align in generating out-of-sample predictions for time-dependent data. We measure such proximity using time series derived from a parent dataset of commodity futures prices employed to produce predictions across multiple forecast horizons, different variable clusters, and data pre-processing techniques. Our model toolbox includes, on the one hand, classical econometric forecasting techniques like Auto-Regressive Integrated Moving Average (ARIMA) models, Vector Auto-Regression (VAR), and Vector Error Correction Models (Xu & Zhang, 2021a; Xu, 2020). On the ML side, we explore Recurrent Neural Networks (RNNs) and, in particular, we focus on their Long-Short Term Memory RNNs (LSTM-RNNs) extension (Bandara et al., 2019; Gilland, 2020). Both RNNs and LSTMs are designed to improve on classical Neural Networks (NNs) when the task involves generating predictions based on (time-) dependent data structures.

Our analysis highlights that the LSTM-RNN consistently performs equally or better than classical econometric methods, especially in longer forecast horizons. Although the ML model cannot independently address seasonality or trends, both of which have a significant negative impact on its performance, it does exhibit the ability to handle structural breaks. In fact, training our ML model on a dataset that includes structural breaks yields better predictions than classical methods, even when these are explicitly laid out to control for such abrupt changes in the data distribution. In particular, our findings suggest that LSTM-RNN models are particularly well-suited for situations where addressing a structural break entails splitting the dataset, thereby avoiding the resulting loss of information.

Complementing the existing literature, we contribute by addressing explicitly the impact of data pre-processing, the role of structural breaks, and the multivariate nature of these predictive tools. The importance of jointly exploring all these aspects is explicitly mentioned in the commentary of the M5 competition, where the best performing algorithms both included further explanatory variables and pre-processed the data to at least remove systematic dependence components (Makridakis et al., 2022). In this respect, the contribution of this paper is threefold: first, it provides specific insights into the importance of data pre-processing, especially when time series are characterized by structural breaks, a rather frequent feature of agricultural commodity futures prices. Second, we explore the multivariate nature of ML and traditional forecasting tools by organizing our dataset into three variable clusters that include both agricultural and financial time series (see section 2). These groupings are defined according to the corresponding cointegrating relationships. Third, we analyze explicitly the performance of LSTM-RNNs under different time horizons (short and medium-term), against a larger basket of traditional econometric models, both univariate and multivariate.

The paper is structured as follows: section 2 outlines the fundamentals of the methods employed in this paper, detailing their structure and operation. Section 3 presents the data, the various sets of variables used in our models, and the pre-processing procedures. Section 4 examines the estimation procedures for both classical and ML models, evaluating their forecasting capabilities and comparing accuracy measures. Section 5 presents the results of our analysis, with a focus on the forecasting performance of both classical and ML methods. Finally, section 6 provides conclusions, critical reflections, and a discussion of key insights.

2 | ECONOMETRIC AND MACHINE LEARNING APPROACHES FOR TIME SERIES FORECASTING

2.1 | Econometrics approaches and how ML differs from them

The most widely adopted econometric model for time series is the Autoregressive Integrated Moving Average (ARIMA) (Greene, 2020). ARIMA describes the relationship between future and past observations of a certain variable through a single linear-in-parameters autoregressive equation. This structure can be extended to the multivariate case by including other variables in a reduced form model where each series is regressed on its lags and, possibly, other series' lags. The resulting class of models is known as Vector Autoregressions (VAR) (Greene, 2020). A further extension to the VAR allows to tackling non-stationary series by explicitly modeling one or more cointegrating relationships among the series (Greene, 2020). The resulting model is known as Vector Error Correction Model (VECM) and it is widely adopted in the economic literature. The functional form, the lags, and all the other elements composing ARIMA, VARs and VECMs are chosen by the researcher depending on her representation of reality, knowledge of the underlying theory and a vast array of econometric specification tests. The act of deliberately choosing the models' structures through domain knowledge and formal testing creates the first and foremost gap between classical econometric tools and Machine Learning (ML) techniques.

What characterizes ML is the development of algorithms able to learn by practice, choosing its functional form and the number of parameters to include, meaning that the model is almost completely data-driven (Bishop, 2005). This means that the ability to perform a certain task (e.g., prediction), as measured by a certain performance (e.g., accuracy), improves the more experience (e.g., data) the algorithm is exposed to. Ideally, the larger the dataset, the better the performance, provided that both the data structure and the data quality are adequate for the task.[1] Without the need to explicitly represent or interpret the core of these tools, great flexibility is allowed, thereby making predictions potentially very accurate. In this context, one of the key differences between classical methods and ML algorithms is the role of *overfitting* and *regularization*.

To produce accurate forecasts, models of any kind should avoid interpolating the data, thereby picking up unnecessary noise that would hinder their ability to generalize to unobserved data with different distributions. Since the risk of overfitting typically grows with the number of parameters (i.e., overparameterization), highly complex models are inherently more sensitive to these concerns. For this reason, and unlike most econometric techniques, bias is not the focus of ML methods. Instead, such tools are crafted to minimize composite metrics like the Mean Squared Error (MSE), thus striving to strike a balance between variance and bias. The appropriate bias-variance trade-off is achieved through regularization, the strength of which is typically controlled through one or more hyperparameters. When forecasting non-stationary time series, overfitting is particularly problematic: since we expect that future observations exhibit rather different distributions, ML models need a smart way to penalize excessive

[1] One striking example is causal inference, where the application of off-the-shelf ML techniques does not guarantee the identification of treatment effects unless more assumptions and restrictions are imposed (Chernozukov et al., 2017).

complexities that would impose rigid constraints over the range of possible future outcomes. Moreover, the persistent nature of the data can itself lead to overfitting, even in the most parsimonious models.[2]

In the remainder of this section, we briefly describe the architecture and the intuition behind the class of NNs that we contrast to ARIMA, VAR and VECM models. Since classical NNs (referred to as "vanilla" NNs) are designed to best work with independently and identically distributed data, we rely on RNNs to account for time dependencies in the data. RNNs have also been extended to incorporate long-term dependencies through the inclusion of "memory cells." Such extension goes under the generic name of LSTM-RNNs.

2.2 | NNs, RNNs, and LSTM RNNs

NNs take their name from the functioning of biological neurons, but despite some similarities, the "artificial brain" and the natural one bear little resemblance to each other. The basic building block of a NN is the neuron, which can be regarded as an element holding a value between 0 and 1. This value is computed starting from the weighted summation of the inputs, where the weights assigned represent the parameters of the resulting model. The result from the sum is then fed into the activation function which forces the value of the summation in the 0–1 range. There are several possible choices for the activation function, but if we consider a sigmoid function, then a neuron is perfectly analogous to a logistic regression (Rosenblatt, 1958). One can extend this basic model by combining multiple neurons into layers, which can be themselves connected to build an intricate network. Consequently, an incredibly complex system through the addition of layers is created– the more the layers, the higher the complexity. This convoluted mapping can virtually approximate any given function (Staudemeyer & Morris, 2019). The model obtained by this combination is sometimes called *the graph*, and it is conditional on all the parameters of each neuron (weights and biases), as well as on the different activation functions (Hastie et al., 2009: 403–458). The estimation of these parameters takes place during the *learning phase*, which refers to an optimization problem, where the objective loss function represents the error (or distance) between predictions and observed values. Due to the high complexity of the graph, and the large number of parameters, it is often infeasible to obtain an analytical solution by computing derivatives. Therefore, the above procedure is tackled through numerical methods, the most commonly adopted being (Stochastic) Gradient Descent (Ruder, 2016).

Depending on how the layers are connected, NNs can manifest various architectures. Each architecture modifies the NN to impart distinct characteristics, rendering it more or less suitable for specific tasks. For instance, the RNN architecture was initially developed to address certain limitations of the so-called "vanilla" NNs, particularly the inability to explicitly incorporate dynamic structures, which hindered their capacity to effectively model data with stochastic dependencies. Therefore, the basic NN model was extended with two additional functionalities. The first innovation consisted of the introduction of special hidden layers, commonly referred to as *memory cells*, containing recurrently connected neurons able to accumulate sequential information and maintain the knowledge acquired over time (Bandara et al., 2019). Second, the ability to iterate over the time series was introduced, thereby enabling NNs to feed the current step as part of the input for the next stage of the graph (Gers et al., 2002). Unfortunately, these two features brought up yet another problem that greatly frustrates the performance of RNNs in time series forecasting: the impossibility of retaining long-term dependency. In short, the information from the most recent observations overrides the oldest ones, since the model has no control over how to store and eventually use specific past information (Bengio et al., 1994; Hochreiter & Schmidhuber, 1997).

The LSTM-RNN, by re-designing the NN to include two additional features regulating the information flow along the network, is able to effectively handle the vanishing gradient problem as well as learn effectively the long-term dependencies (Gers et al., 2002; Hewamalage et al., 2021; Hochreiter & Schmidhuber, 1997; Sherstinsky, 2020; Staudemeyer & Morris, 2019). Figure 1 provides a graphical visualization of an LSTM-RNN, which is designed as a set of recurrently connected networks called memory cells (the block indicated by the letter A). The network is represented as linked sequentially over time with itself, to reflect its recurrent nature. The novelty of this model is that each of its cells is equipped with a Constant Error Carousel (CEC) and three gates (input, forget, and output gate), working together to govern the push forward of past information (h_t), to be combined with the inputs (X_t), thereby safeguarding long-term dependencies (Staudemeyer & Morris, 2019). This is achieved by a series of computations involving first the decision of the values to be updated via sigmoid (σ) or arctangent ($tanh$) functions, and then their modification via pointwise addition ($\oplus$) or multiplication ($\otimes$). See Annex A in the Supplementary material for a brief explanation of LSTM-RNN functioning.

As previously introduced, all ML techniques require regularization to avoid overfitting the data, for instance

[2] A well-known example is a random walk process, where the best one-step ahead forecasts are provided by the previous observation.

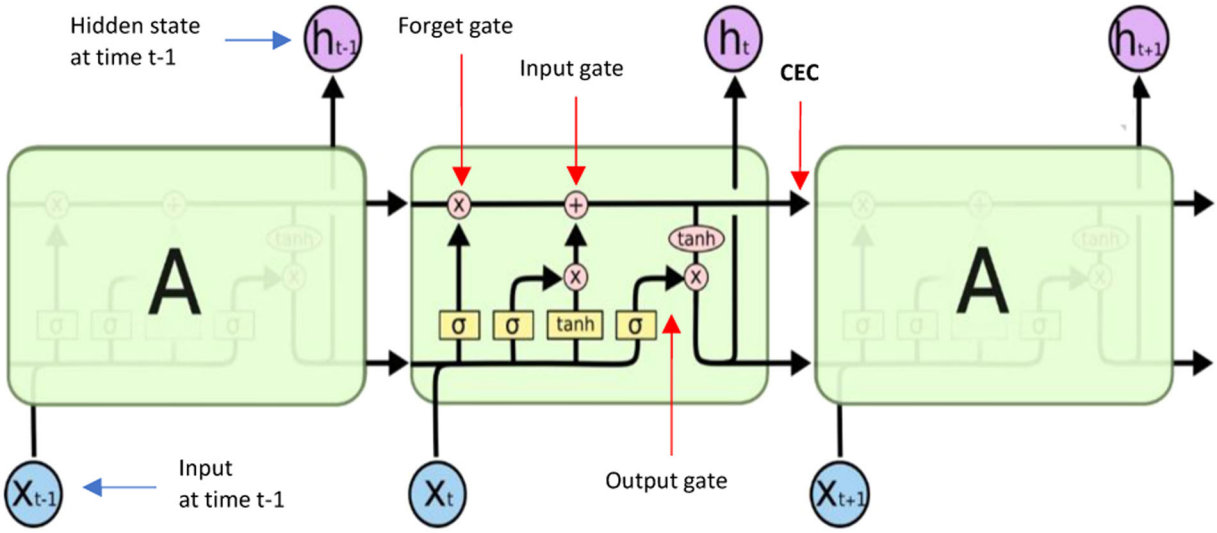

FIGURE 1 Graphical representation of a LSTM Network, unfolded. *Source:* Gers et al., 2002.

limiting the number of epochs or including penalty terms in the loss function (Bishop, 2005; Hastie et al., 2009). However, there exist other approaches that are specific to the data structure at hand. In time series problems, for example, the use of rolling-windows can be thought of as a surrogate regularization strategy, where parameters' learning is smoothed over different time intervals thereby lowering the influence of problematic observations and reducing the risk of overfitting (see section 4.2 for details). In other words, this technique should limit, in a similar fashion to the aforementioned practices, the influence that a particular variable may exert over a given period of time. We introduce these strategies in sections 3 and 4, where we describe pre-modeling data transformations and the LSTM-RNN forecasting algorithm that is used in this paper.

3 | DATA

3.1 | Data description

Our dataset consists of 13 time series measured at daily frequency between January 2000 and June 2020, for a total of 5174 time-steps for each series. We opted to analyze data at a daily frequency for two reasons: on the one hand, we strived to strike a balance[3] between the number of observations and computational constraints; on the other, this level of granularity is necessary to produce short-term forecasts, as detailed in section 4. Indeed, many situations including the stipulation of crop insurance programs and when informing financial investment strategies require forecasts over relatively narrow time windows, thus the need for highly disaggregated data (i.e., daily, or weekly observations) (Andreasson et al., 2016; Sanders & Irwin, 2016; Zhao, 2021). Therefore, our interest in daily data is not new in agricultural price forecasting problems. As an early example, Leuthold et al. (1970) employ daily hog prices and quantities to compare the functioning of several econometric forecasting models. More recently, Ouyang et al. (2019) considered daily futures prices of 12 agricultural commodities, while RL & Mishra (2021) used daily spot prices to predict 5 different agricultural products.

Our data represents six futures contracts traded on different markets: grains, energies, animal products, gold, currencies, and financial indexes. Table 1 presents summary statistics. Additionally, Annex B includes the corresponding plots for all series.

We choose corn futures prices[4] as our primary outcome of interest—that is, the variable that we will use to test the predictive performances of the various methods—while the 12 remaining times series serve as covariates in either the LSTM-RNNs or the VAR/VECM models. Our decision to focus on corn is twofold. Besides being a major animal fodder crop worldwide and, for this, having major economic relevance (Xu & Zhang, 2021a), corn is also

[3] The use of weekly or monthly data would have resulted in series too short for the application of neural networks, while employing a higher frequency would have required significantly more computational power, especially when considering the optimization of multiple neural networks.

[4] An alternative target for exercises like ours could be spot prices, as futures prices tend to converge to spot prices upon maturation. Spot prices can serve as a good proxy for uncertainty in the market. However, in our specific case, we chose to focus on futures prices primarily due to data availability reasons.

TABLE 1 Summary statistics.

Variable	*N*	Mean	Std. Dev.	Min	Max	Kurtosis	Skewness
Corn	5174	378	149	175	838	3.365002	.9688616
Soybean	5174	912	308	418	1772	2.274633	.2832183
HardW	5174	525	187	270	1322	3.206699	.8704956
SoftW	5174	568	203	286	1943	4.910356	1.0099422
SP	5174	1614	624	675	3392	2.858940	.9705892
Nasdaq	5174	3205	2135	810	10952	3.637341	1.2284344
Gold	5174	957	474	255	1900	1.656952	.1021518
WTI	5174	61	26	13	145	2.311986	.4000311
Dollar	5174	91	12	72	121	2.694651	.6468928
Milk	5174	15	3.4	8.6	25	2.662383	.2368158
Fcattle	5174	125	36	73	244	3.644849	.9408071
Lcattle	5174	102	24	60	173	2.501645	.4646261
Gasoline	5174	1.8	.72	.49	3.6	2.161262	.3077619

Note: We run normality tests for all the series in the table (Xu & Zhang, 2022). Specifically, we perform Jarque-Bera, Kolmogorov-Smirnov, Cramer von-Mises and Anderson-Darling tests. None of the tests suggest that the series follow a normal distribution.

largely used in biofuel production, for example in the US (US Department of Energy, 2023). Moreover, by concentrating on corn prices, we can conveniently evaluate the models' capacity to handle complex time features such as unit roots, seasonality, trends, and, importantly, structural breaks (Egelkraut et al., 2007). Concerning the latter, it is widely recognized that corn prices experienced a significant upsurge in the mid to late 2000s, primarily influenced by legislative measures aimed at boosting corn ethanol production (Tyner, 2010). We can therefore use this knowledge to analyze how abrupt changes in the data-generating process impact our predictions (see section 5.1).

All the data were retrieved from Barchart (2020). Since futures contracts have a finite expiration date, unlike stocks that trade perpetually, the nearest series were used to construct continuous series rolling[5] from one contract to the next for the whole period considered.

To evaluate the methods' performances as the number of available variables varies, we consider different model specifications utilizing different groups of variables. Since training an LSTM-RNN on a set of heterogeneous time series could potentially produce low quality predictions, Bandara et al. (2019) propose to initially cluster the available time series and take advantage of any similarities between subsets of data. Although clustering should in principle be based on the properties of the time series, such as frequency, seasonality and persistence, the authors suggest that clustering can also be based on domain knowledge and researchers' familiarity with the data. In particular, the latter approach might be more appropriate when the problem involves few variables with clear conceptual linkages, whereas algorithmic clustering could provide a reasonable compromise in large sets of unstructured time series. For this reason, and to leverage cointegrating relationships involving corn, we do not perform algorithmic clustering but rather seek to compromise between economic knowledge and formal testing. First, we run each model on the full variable set. Second, we select the series belonging to the same domain (commodity and financial prices) and we estimate a cointegrating relationship, retaining only the variables that appear in the long-term structural component. The cointegration analysis identifies three distinct subgroups that align with our prior domain knowledge (Serra et al., 2011; Suh & Moss, 2017). The first cluster consists of eight variables encompassing futures prices of grains, livestock futures prices associated with grains used as feed, and energy prices. The second cluster focuses solely on grains and livestock futures prices. Last, the third cluster comprises only grains' futures prices (Table 2). Through this procedure, we can evaluate the extent to which the various methods benefit from incorporating a larger number of variables.

3.2 | Data pre-processing

In order to ensure the correct functioning of LSTM-RNNs, data need to be pre-processed. Different time series with different scales might prevent the model from learning effectively, by making the parameters diverge during computations. As stated in the introduction, pre-processing

[5] Rolling was done 15 days before the expiration month, based on total volume and open interest (Barchart, 2020).

TABLE 2 Variables clusters.

Models	Variables
LSTM_tot	*All variables are considered*
VAR_tot	
LSTM_8	*Considering only:*
VAR_8	Corn—Soybean—Hard Wheat—Spring Wheat—Feeder Cattle—Live Cattle—WTI Oil—Gasoline
VEC_8	
LSTM_6	*Considering only:*
VAR_6	Corn—Soybean—Hard Wheat—Spring Wheat—Feeder Cattle—Live Cattle
VEC_6	
LSTM_4	*Considering only:*
VAR_4	Corn—Soybean—Hard Wheat—Spring Wheat
VEC_4	
ARIMA	*Only Corn is considered*

Source: own elaboration.

is also fundamental in dealing with some time series properties, such as seasonality, trends, unit roots and, more in general, non-stationarity.

For the comparison between classical models and LSTM-RNNs, we utilized two pre-processing procedures: First Differencing (FD) and the Full Adjustment (FA) approach. Whereas FD is very common in most conventional forecasting tasks, FA is widely used in the ML literature, and it is particularly suited to forecasting with RNNs and LSTM-RNNs (Bandara et al., 2019; Hewamalage et al., 2021; Smyl, 2020). Within this method, there are three elements to take care of: the stabilization of the variance, seasonality, and trend. The stabilization of the variance is performed first, through a logarithmic transformation of the raw data. In practice, this transformation reduces the scale of the data. The de-seasonalization is carried out by dividing the series into time windows of 1 year each and removing the seasonal component via a Locally Estimated Scatterplot Smoothing (LOESS) (Cleveland et al., 1990). Last, the trend is eliminated from the series following the same approach as in the de-seasonalization. The advantage of this technique, besides providing more accurate estimates of local trends and seasonal components, is that it does not introduce information from the training set to the test set, affecting the actual performances of the NN. However, most of the common prediction tasks involving the use of ML techniques, such as the ones we present in this paper, deal with data entailing little or no persistence. Specifically, when strong dependencies are present, these are typically the result of seasonal or (possibly nonlinear) trend components for which the FA approach works particularly well (e.g., Bandara et al., 2020; Mtibaa et al., 2020). On the other hand, economic variables often exhibit complex dynamics even when periodic components have been removed, thereby leading researchers to resort to FD. Although some authors have successfully applied the FA technique to times series of the second type (Bandara et al., 2019), we train our LSTM-RNN on both FD and FA transformed data and keep record of the model's performance across the two representations.

In order to evaluate the models' capability to handle structural breaks, we compare their performances when trained on either the full dataset (henceforth FULL) or on a post-structural break dataset (henceforth post-SB) defined as the collection of all the observations beyond a certain breakpoint. To identify such a threshold, we conducted a battery of Chow tests using a VECM with eight variables (i.e., all the grain, livestock, and energy prices) as the reference model in all the time steps via the *strucchange* R Package (Zeileis et al., 2002, 2003). The resulting post-SB dataset includes 3084 daily observations from April 28, 2008 to June 2020 (see Annex C for more details).

For the assessment of the LSTM-RNN ability to pick up time elements, we compare the performances of our NNs on the datasets obtained from the Full Adjustment (FA) dataset, first-difference (FD) dataset, and three other datasets where we applied either de-seasonalization (Season adj., SA) or de-trending (Trend adj., TA), or no transformation at all (Untransformed, UN) (see Table 3).

4 | ESTIMATION AND FORECASTING

4.1 | Estimation and forecasting with the classical models

To construct the VAR and ARIMA models, both on the FA and the FD transformed datasets, the following procedure was followed (see Table 4). We started by checking

TABLE 3 Pre-processing procedures and resulting datasets.

Dataset	Details	Identifier
	Pre-processing procedure	
Full adjustments	*Logarithmic transformation* *Seasonality removed* *Trend removed*	FA
First difference	*Unit root removed*	FD
Trend adj.	*Trend removed* *Standardization*	TA
Season adj.	*Seasonality removed* *Standardization*	SA
Untransformed	*Standardization only for LSTM*	UN
	Dimension	
Full dataset	*From January 2000 to June 2020* *5174 time-steps*	FULL
Post-structural break	*From the May 2008 to June 2020* *3084 time-steps*	post-SB

Source: own elaboration.

TABLE 4 Classical models specifications.

	Specifications	
	Full adjustment dataset	**First differenced dataset**
VAR_tot	1 lag, constant	1 lag, no constant
VAR_8	1 lag, constant	1 lag, no constant
VAR_6	2 lags, constant	2 lags, no constant
VAR_4	3 lags, constant	3 lags, no constant
ARIMA	AR(1), I(0), constant	AR(1,3), I(0), MA(7,9), no constant
Unprocessed dataset		
VEC_8	1 lags, 3 cointegrating vectors, constant	
VEC_6	2 lags, 2 cointegrating vectors, constant	
VEC_4	2 lags, 1 cointegrating vector, constant	

Source: own elaboration.

the stationarity of each series, using the Augmented Dickey-Fuller (Dickey & Fuller, 1979) test as well as the KPSS test (Kwiatkowski et al., 1992). The univariate model was determined by using the Autocorrelogram and Partial Autocorrelation graph. The choice for autoregressive processes or moving averages was guided by the obtainment of white noise errors, as well as parsimony (Verbeek, 2017). For the multivariate cases, we proceeded with a VAR model, choosing the number of lags using Wald tests (Lütkepohl, 2005). To construct the VEC models on the untransformed dataset, cointegration was tested through the Johansen test (Johansen, 1995). The optimal number of lags was based on AIC, BIC, and Likelihood Ratio tests.

Once the models were fit, predictions for 30 and 7 days ahead were produced. As discussed in sections 1 and 3, planning may require different forecasting horizons depending on the market, the period of interest, the actors involved and the purpose of the prediction. In many cases, the focus is not solely on single or few-step-ahead predictions, but rather on obtaining forecasts for an entire time window. Taking this factor into consideration, we have chosen to use both a 7-day and a 30-day forecast window, during which we make predictions for both the entire sequence of time steps and the averages. The selection of these two forecast windows is based on their frequent utilization in practical applications within financial contexts (Xu, 2020).

Predictions for the classical models are calculated *dynamically* in the sense that the predictions for each time step are used recursively to predict the next step. The reason we chose this option is that it is the most common prediction method when using ARIMA models, despite the

risk of carrying over errors in consecutive predictions given an error in the beginning.

4.2 | Training and forecasting with LSTM-RNNs

Standard NN algorithms are generally trained by k-fold cross-validation,[6] which cannot be applied in a time series context since it does not preserve the order of the observations.

To overcome this issue, we use the Rolling-Window (RW) approach described in Bandara et al. (2019) and Smyl (2020). A simple way to visualize this method is by considering a zipper mechanism, where each tooth on the string can be considered as a time step. The first part of this procedure is designing a window covering a certain number of steps. The window is trained over a fixed proportion of those steps and validated over the remaining part. Starting from the beginning of the sequence, the window is applied, and the first set of parameters is computed. Then, this window is rolled over the sequence, each time updating the parameters so to optimize for forecasting accuracy. The parameters computed in the last window of data are the final parameters for the algorithm. In this way, it is possible to cross-validate the parameters while preserving the temporal state of the data.

Therefore, the first step for training LSTM-RNN models is to define the span of the rolling-window. As discussed in the previous section, we provide both 7-day and 30-day ahead forecasts. Considering the 30 days forecast horizon, we start from a window consisting of a total of 395 time steps, 365 of which are input steps and 30 output steps. Based on the 1-year input window, the NN produces the forecasts for the 30 days ahead. These forecasts are compared with the actual observations and, based on the prediction error, the parameters are updated backwardly optimizing for forecasting accuracy. This procedure is then repeated for the whole length of the series, each time rolling the window input of 180 time steps, to obtain partial overlapping between two consecutive windows. The same reasoning goes for the 7-day forecast horizon, where the rolling-window has 7 output steps.

A key feature of producing forecasts through LSTM-RNN consists of how predictions are calculated over the time horizon of interest. Specifically, the RW window approach to LSTM-RNN generates *static* forecasts, meaning that the algorithm directly outputs predictions over the whole stretch of the forecasting range (Taieb et al., 2010). Conversely, classical models produce forecasts in a *dynamic* way, meaning that they generate single one-step ahead forecasts, and then repeat the process for the whole forecasting horizon, iteratively. The advantage of this last method is that it allows to consider an increasing number of observations while generating the forecasts. However, it also implies that the errors made in previous steps are carried over to the following forecasts (Taieb et al., 2010).

The last step for assessing the performance of the LSTM-RNN is a Bayesian Optimization (BO) of the hyperparameters (Bandara et al., 2019; Snoek et al., 2012). This procedure allows to efficiently find the combination of hyperparameters that maximizes the accuracy of the NN's predictions using fewer iterations than with a standard grid search algorithm. Shahriari et al. (2015) provide an accurate introduction of BO, explaining the importance of probabilistic optimization in the presence of a highly complex function with millions of possible hyperparameter combinations. Tables 5 and 6 report respectively the chosen hyperparameters value for each LSTM-RNN model in our analysis trained on the FA or the FD datasets.

4.3 | Accuracy measures

We measure the performances of classical models and LSTM-RNNs, via three distinct accuracy measures: the Mean Absolute Error (Equation 1), the Root Mean Square Error (Equation 2), and the Mean Absolute Percentage Error (Equation 3).

$$MAE = \frac{\sum |y - \hat{y}|}{N} \tag{1}$$

$$RMSE = \sqrt{\frac{\sum (y - \hat{y})}{N}} \tag{2}$$

$$MAPE = \frac{\sum \left(|y - \hat{y}|\right)}{|y|} * 100 \tag{3}$$

This choice is mainly dictated by the large popularity they have in the forecasting literature (Goodwin, 2020). The first two measures are usually employed in evaluating different models on the same dataset, while the third is more versatile (and scale-independent). Since the RMSE is more sensitive to outliers (Hyndman & Koehler, 2006), the discussion focuses on the MAE, while the remaining measures are reported in Annex D in the Supplementary Material.

[6] K-fold cross-validation consists in randomly splitting the training data into k non-overlapping folds, fitting the model to each subset, and then averaging the resulting predictions.

TABLE 5 Chosen hyperparameters value for each LSTM-RNN model trained on the FA dataset.

FA	30-day forecast horizon				7-day forecast horizon			
Hyperparameters	**LSTM_tot**	**LSTM_8**	**LSTM_6**	**LSTM_4**	**LSTM_tot**	**LSTM_8**	**LSTM_6**	**LSTM_4**
Learning rate	5.6e-04	.0002	1e-01	1e-01	2.3e-06	.002	2.6e-03	1e-02
Number of LSTM layers	5	5	2	2	2	5	2	2
Number of nodes per layers	333	512	9	9	344	230	344	324
Lambda regularizer	1e-01	.001	1e-03	1e-03	1.2e-02	.1	1e-03	5.4e-03

TABLE 6 Chosen hyperparameters value for each LSTM-RNN model trained on the FD dataset.

FD	30-day forecast horizon				7-day forecast horizon			
Hyperparameters	**LSTM_tot**	**LSTM_8**	**LSTM_6**	**LSTM_4**	**LSTM_tot**	**LSTM_8**	**LSTM_6**	**LSTM_4**
Learning rate	1e-06	1e-06	1e-06	1e-06	1e-06	7.68e-05	1e-06	1e-06
Number of LSTM layers	4	2	4	4	2	1	5	5
Number of nodes per layers	5	5	5	5	5	512	5	5
Lambda regularizer	.001	.001	.1	.1	.001	.1	.001	.001

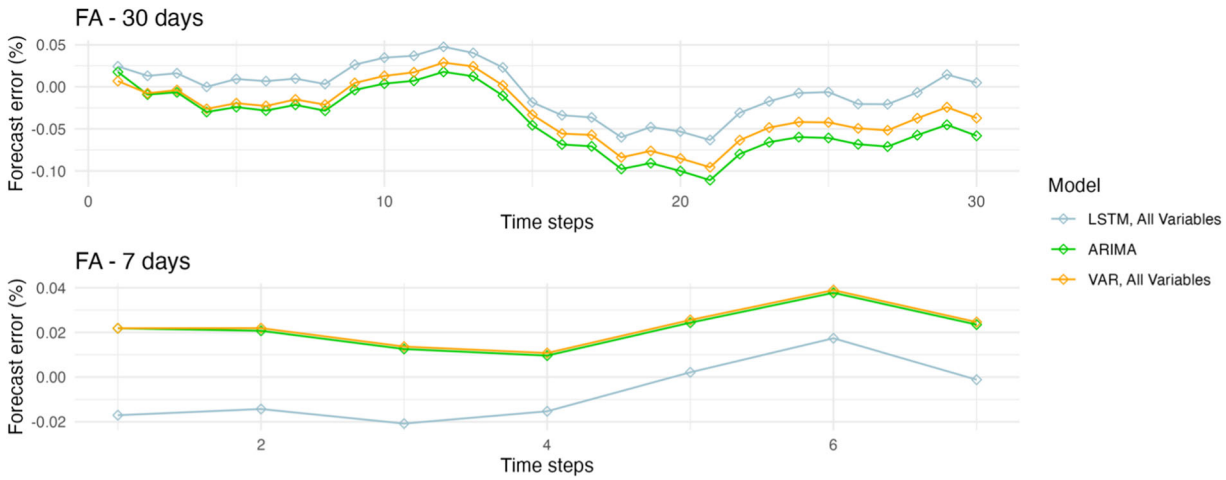

FIGURE 2 Performance comparison of LSTM versus classical models, on the FA dataset for 30 and 7 day forecast horizon. *Source:* own elaboration.

5 | RESULTS

5.1 | Performance and structural break comparison

Figures 2 and 3 report both the 30-day and 7-day forecasts obtained through the LSTM-RNN trained on the FA and FD data, respectively, as well as the forecasts generated by classical models using the whole set of variables. Figure 4 reports the comparison between LSTM-RNN and VAR models trained on the FD dataset versus the VECMs fit on the untransformed data for both the 30- and 7-day forecast horizon. This visual inspection provides a first rough assessment of all the methods presented throughout the paper, showing mixed results depending on the dataset and forecast horizon considered. However, since graphic representations are prone to subjectivity and possible misinterpretations, the next section investigates the different performances through the MAE accuracy measure.

Table 7 provides the performance measures for the models based on the FA dataset, considering both the FULL and post-SB datasets, for both 30-day and 7-day forecast horizons. When examining the longer forecast horizon, the MAEs indicate that LSTM-RNNs trained using the FA approach consistently outperform the classical models, with values 10% lower on average. However, switching

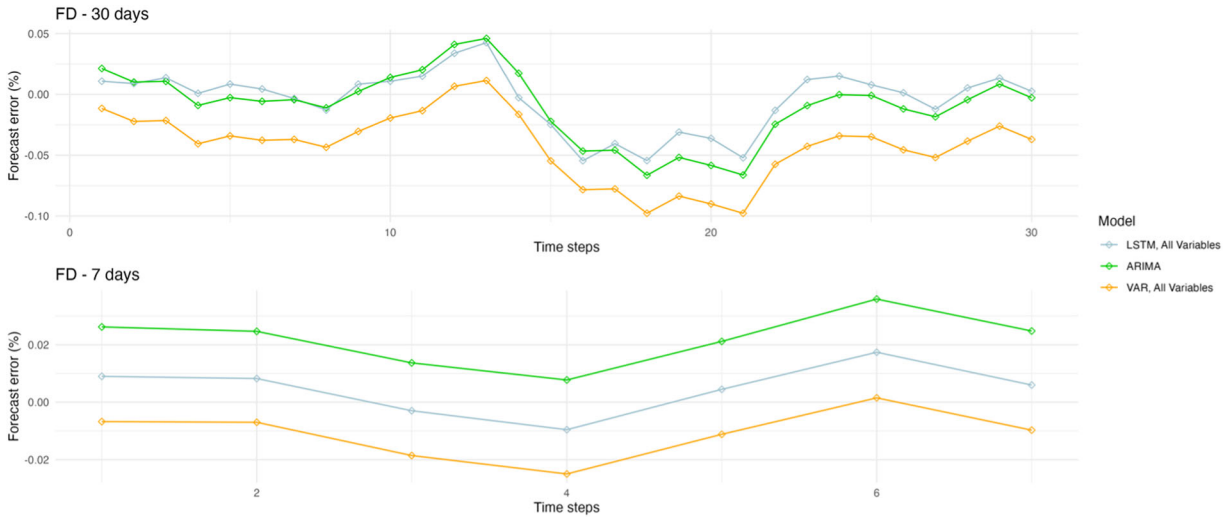

FIGURE 3 Performance comparison of LSTM versus classical models on the FD dataset for 30 and 7 day forecast horizon. *Source:* own elaboration.

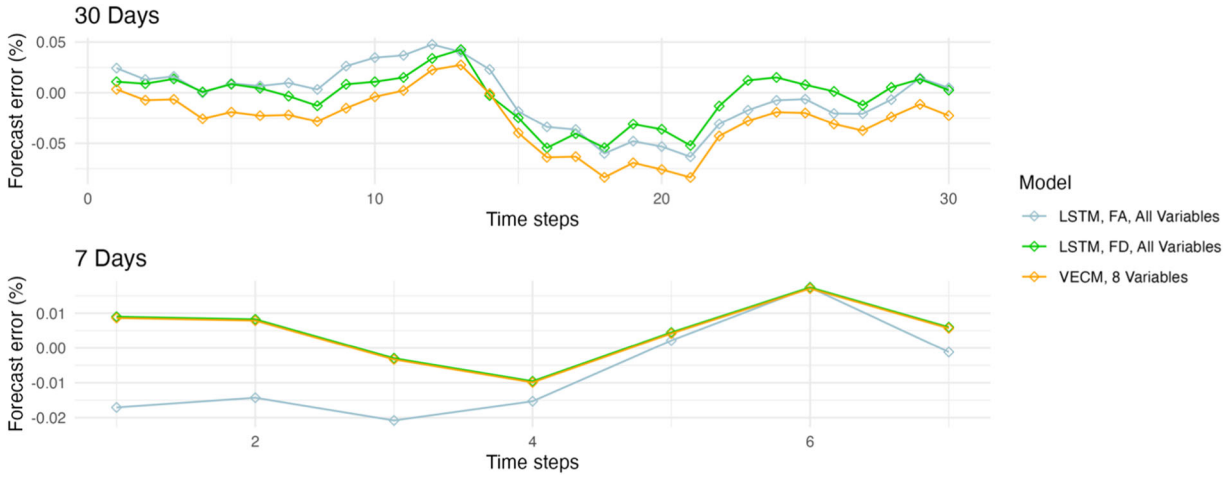

FIGURE 4 Performance comparison of LSTM versus VECMs for 30 and 7 day forecast horizon. *Source:* own elaboration.

TABLE 7 Mean absolute error for all models and forecast horizons on both FULL and post-SB FA dataset.

	7-day forecast horizon		30-day forecast horizon	
Model—FA	**post-SB**	**FULL**	**post-SB**	**FULL**
LSMT_tot	29.63	**4.22**	12.44	8.30
LSTM_8	23.93	17.98	13.75	10.59
LSMT_6	22.19	3.92	12.35	8.41
LSTM_4	19.72	5.66	11.53	**8.25**
VAR_tot	5.14	7.50	16.22	12.50
VAR_8	5.18	5.77	15.08	12.92
VAR_6	5.61	6.42	14.80	13.45
VAR_4	5.72	6.47	14.02	14.34
UNI	11.52	7.17	13.46	15.66

Source: own elaboration.

TABLE 8 Mean absolute error for all models and forecast horizon on both FULL and post-SB FD datasets.

	7-day forecast horizon		30-day forecast horizon	
Model—FD	**post-SB**	**FULL**	**post-SB**	**FULL**
LSMT_tot	3.37	**2.75**	10.46	**6.29**
LSTM_8	2.29	**2.75**	10.46	10.46
LSMT_6	2.37	**2.75**	10.46	6.42
LSTM_4	2.23	**2.75**	10.88	7.13
VAR_tot	2.46	3.83	10.68	14.76
VAR_8	2.67	2.78	10.47	10.39
VAR_6	2.68	3.83	10.47	14.19
VAR_4	2.70	3.95	10.68	14.25
UNI	3.03	7.35	10.43	7.46

Source: own elaboration.

to the 7-day forecast horizon yields mixed results. On the one hand, restricting the dataset to post-SB does impact the performance of the LSTM-RNNs rather severely, whereas the MAE of the VAR models improves slightly when focusing on the observations beyond the estimated cut-off point. In this setting, we find that classical econometric models largely outperform LSTM-RNNs. On the other hand, when LSTMs are trained on the FULL dataset, they consistently yield better predictive performances than VARs and the univariate model. Overall, Table 7 seems to suggest that the longer the time series, the better the results NN can produce, regardless of the presence of potential structural breaks. Conversely, if the distribution of the data abruptly changes at some point in time, fitting VAR models to a restricted observation range can be beneficial, especially when the forecast horizon is short.

Next, Table 8 displays the performance measures for the models based on the FULL and post-SB FD datasets, considering both 30-day and 7-day forecast horizons. When utilizing FD data, both LSTM-RNNs and classical models exhibit far superior forecasting performances, as evidenced by lower MAE values. Focusing on the 30-day prediction exercise, the performance of the ML methods is comparable to that of classical models when trained on the post-SB restricted dataset. In this setting, the best performance is achieved by the univariate models, although the results from the other methods are almost just as good. However, LSTM-RNNs prove once again better when one utilizes the entire time series, with MAE values largely lower than those obtained through either VAR or univariate models.

Shifting the focus to the 7-day forecasts, ML models once again yield the best predictive metrics, regardless of whether the training was performed on the post-SB or the FULL dataset. This time, however, the MAE is on average lower in the restricted sample, although the difference with respect to the models estimated on the whole series is minimal. When it comes to the VAR and the univariate specifications, however, the improvement of fitting classical models on the post-SB dataset appears more evident. Once again, however, LSTM-RNNs achieve the best overall forecast accuracy, whether we consider the presence of a structural break during the learning step or we do not. When using FD data, however, the differences are not as striking.

Finally, Table 9 presents the comparison between the LSTM-RNNs and the VAR models trained on the FD dataset versus the VECMs estimated on the raw dataset. On the longer forecast horizon, the LSTM-RNN and VAR models perform better than the VECM. Despite the additional structure that these models incorporate (i.e., the cointegrating relationship), compared to reduced-form specifications such as VARs, there is no improvement in terms of predictive performance. However, since VEC models primarily focus on estimation rather than forecasting, one might think of these tools as reasonable compromises between interpretability and accuracy. This line of reasoning has already emerged in the literature, where some authors argue that theory-driven methods might not be the best choice when prediction is the main objective (Breiman, 2001).

When it comes to 7-day forecast, we witness a convergence between LSTM-RNNs and VEC models, with the latter exhibiting slightly better performances than the corresponding VARs. Moving from the post-SB to the FULL dataset, the performances of the VECMs on the longer forecast horizon are on average comparable, while they slightly worsen on the shorter forecast horizon.

5.2 | Composite forecasts

In addition to analyzing each model individually, we also explore the combination of forecasts from both classical and ML models, as outlined in Xu and Zhang (2021b),

TABLE 9 Mean absolute error comparison for best performing LSTMs and VECMs on both FULL and post-SB FD datasets.

	7-day forecast horizon		30-day forecast horizon	
Model	**post-SB**	**FULL**	**post-SB**	**FULL**
LSTM_8	**2.29**	2.75	10.46	10.46
LSMT_6	**2.37**	2.75	10.46	**6.42**
LSTM_4	**2.23**	2.75	10.88	7.13
VAR_8	2.67	2.78	10.47	10.39
VAR_6	2.68	3.83	10.47	14.19
VAR_4	2.70	3.95	10.68	14.25
VECM_8	2.46	6.63	11.14	12.45
VECM_6	2.46	6.38	12.81	11.26
VECM_4	2.55	2.70	12.26	10.55

Source: own elaboration.

TABLE 10 Mean absolute error for equal-weight composite forecasts across FA and FD datasets for the 30 and 7 day forecast horizon.

Model	**FA**	**FD**
	30-day forecast horizon	
LSTM_tot	**10.46**	12.44
VAR_tot	10.68	16.22
ARIMA	13.46	10.43
Combination	10.6	14.1
	12.05	
	7-day forecast horizon	
LSTM_tot	3.37	29.63
VAR_tot	**2.46**	5.14
ARIMA	11.52	3.03
Combination	3.53	2.98
	2.96	

TABLE 11 Mean absolute error for inverse-MSE composite forecasts across FA and FD datasets for the 30 and 7 day forecast horizon.

Model	**FA**	**FD**
	30-day forecast horizon	
LSTM_tot	**10.46**	12.44
VAR_tot	10.68	16.22
ARIMA	13.46	10.43
Combination	10.5	14.7
	7-day forecast horizon	
LSTM_tot	3.37	29.63
VAR_tot	**2.46**	5.14
ARIMA	11.52	3.03
Combination	2.89	4.08

Trujillo-Barrera et al. (2016), and Timmerman (2006). We consider two types of combinations: an equal weight composition (Table 10) and an inverse-Mean-Squared-Error (inverse-MSE) combination (Table 11), where we treat the latter as a robustness check for the former. The equal-weighting scheme, which does not require the estimation of weights to average the forecasts, has been shown to outperform more sophisticated aggregation methods (Timmermann, 2006). This technique is also particularly interesting for our work because it allows to combine models trained on the same dataset but pre-processed in different ways. The inverse-MSE combination scheme, on the other hand, does not allow for such flexibility, because MSE is a scale-dependent measure. Since we only have a limited number of models to combine, no models were excluded when compositing the forecasts, and no trimming was carried out. For simplicity, we focus on the subset of models trained on the full variable set, while we leave the discussion on the remaining model set for Annex E. The performance metrics of the composite forecasts are presented in Tables 10 and 11.

Our results show that combining forecasts in our model set-up does not bring any substantial advantage in terms of predictive accuracy.

5.3 | Trend and seasonality

In this section, we wish to emphasize how important pre-processing is to obtain satisfying predictive performances when working with ML methods. Although the standard scaling procedures discussed in the previous sections yield very good forecast performances, training LSTM-RNNs on unprocessed data with seasonal and trend persistence components might result in poorer results. This issue is discussed in Bandara et al. (2019) who suggest that time series with a trend component might be challenging to forecast for off-the-shelf LSTM-RNN algorithms.

TABLE 12 Performance comparison of LSTM across dataset containing different time elements.

Models	MAE				
	FA	FD	Untransformed	Trend adjusted	Season adjusted
LSTM_tot	8.93	6.29	442.72	348117.9	125.64

Source: own elaboration.

TABLE 13 Percentage bias in the weekly and monthly average forecast on the FA dataset.

Model—FA	7-day forecast horizon	30-day forecast horizon
LSMT_tot	−.088	−.026
LSTM_8	−.071	−.032
LSMT_6	−.066	−.025
LSTM_4	−.059	**−.020**
VAR_tot	**.015**	−.046
VAR_8	.015	−.042
VAR_6	.017	−.040
VAR_4	.017	−.037
UNI	.034	−.032

TABLE 14 Percentage bias in the weekly and monthly average forecast on the FD dataset.

Model—FD	7-day forecast horizon	30-day forecast horizon
LSMT_tot	−.005	−.028
LSTM_8	.007	−.028
LSMT_6	.007	−.028
LSTM_4	.004	−.029
VAR_tot	**.003**	−.028
VAR_8	.004	−.028
VAR_6	.004	−.029
VAR_4	.004	−.028
UNI	.006	**−.027**

TABLE 15 Percentage bias on the weekly and monthly average forecast for VECMs.

Model	7-day forecast horizon	30-day forecast horizon
VECM_8	**.003**	−.034
VECM_6	**.003**	−.036
VECM_4	**.003**	**−.030**

Table 12 shows that, when fed with data adjusted for seasonality but not trend, the LSTM-RNN produces predictions that are completely out of scale. Similarly, training the NN on trend-adjusted but seasonality-unadjusted time series yields nonsensical forecasts, as the literature seems to suggest. Finally, estimating the LSTM-RNN on the raw data (i.e., data where neither the trend nor the seasonal component was removed), leads to predicted prices that are still completely out of bounds. Overall, this empirical exercise seems to suggest that adjusting for at least seasonality is critical to improving the forecasting accuracy of LSTM-RNNs. However, partialing out both seasonality and trend components remains a necessary stage in time series forecasting using RNN.

5.4 | Weekly and monthly forecasting averages

In order to complete the picture, and given their relevance to several economic operators (Hoffman et al., 2015), we also analyze forecasting of monthly and weekly average prices. In this case, the best performances are observed when using the FD dataset, where there are no significant differences between the classical models and the ML models (Tables 13–15). On the FA dataset (Table 14), the classical models demonstrate better performance in the weekly forecasts, while the ML models outperform in the monthly forecasts. Comparing VECMs to VARs trained on the FD dataset, VECMs perform similarly to VARs for the weekly forecast but show inferior performance for the monthly one.

5.5 | Discussion

The results presented in the previous sections demonstrate that LSTM-RNNs outperform classical econometric models in forecasting lengthy and intricate time series, though with certain caveats. Since the predictions generated through classical models often tend to plateau over longer forecast horizons, the fact that ML techniques yield better performances comes with little surprise. However, the argument is different when it comes to short-term (i.e., 1 week) forecasts. We expected that classical econometric models, due to their greater rigidity and thus reduced sensitivity to noise, could still deliver competitive results, compared to ML models. Nevertheless, this was only observed in the case of the VARs on the FA dataset.

In general, we observe a significant reduction in prediction error when transitioning from a 30-day to a 7-day forecast horizon across all models and pre-processing techniques. This phenomenon is likely attributed to the issue of error propagation in dynamic forecasts, which becomes more pronounced due to the inflexibility of classical models, resulting in a more substantial decline in accuracy. Classical models may indeed yield accurate predictions over short timeframes, provided that the observations are

not too distant in the future. However, as we extend the forecasting horizon, disparities about future observations may increase due to the limited flexibility (e.g., linearity in lag parameters) of these models, leading to higher MAE over a 30-day horizon. In this regard, the nonlinearities inherent in LSTM-RNNs, combined with their capacity to regulate complexity dynamically and locally through the RW approach, may render these methods better tools for precise future forecasting.

One obvious concern with any forecasting exercise is the presence of structural breaks. When the distribution of the variables of interest changes abruptly at some $t \in \{1, \dots, T\}$, the ability to forecast future events can potentially deteriorate if such discontinuity is not properly accounted for. This problem is evident when VARs and VECMs are used to perform the predictions: when these modes are trained on a post-structural-break dataset, they typically yield lower forecasting errors than the corresponding models estimated using the full series.

On the other hand, our empirical analysis demonstrates that LSTM-RNNs can automatically handle structural breaks, thus obviating the need for the researcher to explicitly search for potential breakpoints. This is a very important property since testing for structural breaks typically presents several challenges, including the need for critical assumptions, multiple testing, and modeling choices. Currently, there is a lack of clear guidance on whether it is more advantageous to just use raw data or select a specific model (such as ARIMA, VECM, or VAR) when looking for structural breaks. In case the latter option is preferred, further complications arise from the need to accurately specify the model, which entails navigating through a wide array of potential modeling choices.

Additionally, attempting to deal with structural breaks via dummy variables can be highly sensitive to even slight inaccuracies in determining the time step at which the break occurs. Such a misstep can significantly impact estimation, leading to potential model misspecification and false detections. Therefore, the automatic handling of structural breaks by LSTM-RNNs offers a promising alternative, allowing researchers to leverage the full extent of the data without the complexity and potential pitfalls associated with traditional approaches to structural break detection. However, it is still unclear whether this ability can be fully attributed to the network structure itself or the rolling-window approach. To this extent, further research remains necessary.

Conversely, we find that LSTM networks were not able to pick up trends or seasonality when trained on a set of similar time series. Differently from what is suggested in Bandara et al. (2019), our NNs struggled to produce sensible predictions when trained on trend-adjusted (but not seasonality-adjusted) data. When, on the other hand, the adjustment concerns seasonality and not trends, the LSTM-RNNs yield lower prediction error, although their performance remains unimpressive. Interestingly, using trend-adjusted series performed worse than fitting the model with the raw dataset where both the trend and seasonal components are still present. While removing the trend should help to achieve more accurate predictions, our results seem to suggest that it instead prevents the NN from correctly learning the underlying data-generating process, thereby leading to a great decrease in precision. This poor performance could be explained by the training procedure via rolling-window, where the size of the sliding training set could be insufficient for the algorithm to provide a full representation of these deterministic components.

Finally, we find that the best performances for both the longer and the shorter forecast horizon are obtained on the FD dataset. Full Adjustment might be more suited for applications as seen in the M-competitions (Hyndman, 2020) rather than forecasting agricultural futures prices. Indeed, these series are typically characterized by having defined time series characteristics,[7] and therefore do not need such heavy pre-processing. Moreover, unlike FD, FA will not make the series stationary in the presence of a unit root.

5.6 | Limitations

Our work has also some important limitations, both concerning technical aspects but also issues that require further exploration. First, when evaluating the performances reported in this section, we have to emphasize that choosing among the different ways by which the different models produce their forecasts is not trivial. The LSTM-RNN that we implemented is a sequence-to-sequence model, meaning that it directly generates a sequence of outputs from an input sequence. In contrast, the classical models make forecasts in a dynamic (or autoregressive) way, meaning that each forecast is used as additional information for the following forecast. There are obviously consequences and trade-offs among these different approaches: on the one hand, one may argue that the classical models perform worse on the longer forecast horizon because they carry the errors from the previous forecasts to the end, while they perform better on the shorter forecast horizon where the potential forecasting error builds up to a lesser extent. On the other hand, the sequence produced by the LSTM-RNN can be sensitive to the choice

[7] For instance, the frequency is dictated by the futures market, while the seasonality is dictated by nature.

of the rolling-window, which is also a critical modeling aspect. Alternative approaches might include splitting the data into train and test datasets, but even in this case defining which observations belong in either subset can be challenging.

Another limitation concerns the amount of data available. What makes machine learning models interesting is their ability to handle both a larger number of variables and many more observations than traditional econometric models (Efron, 2020; Storm et al., 2020). However, since neural networks need sufficiently long time series to be trained effectively (Bandara et al., 2019), what would happen if we trained them on a shorter series? This could be done by either truncation, decreasing the frequency of the data or focusing on problems where fewer data is available. Such exercise would also allow us to determine how neural networks perform when faced with data at different frequencies or with different noise-to-signal ratios.

Finally, at the time of writing this paper, there are no available methods to compute the Shapley values for the specific model we have implemented. Technically, there is currently no existing approach that supports a model producing a 3D output (features-steps-batch, a technical detail of the LSTM-RNN implementation in Python) while also integrating the rolling-window approach, due to compatibility constraints. Consequently, we are unable to identify which time series demonstrates the highest predictive capability.

The possible ways forward move in multiple directions. The first is more practical and concerns extending the comparison to different datasets using different pre-processing procedures, to (possibly) confirm the results of this analysis. The second is more theoretical, and it is directed at answering more technical questions such as the relationship between regularization and non-stationarity. It is our opinion that regularization could be considered as a strategy implemented by neural networks to deal with non-stationarity, by allowing more flexibility in the forecasts with respect to the past.

6 | CONCLUSIONS AND IMPLICATIONS

This paper evaluates the degree to which machine learning techniques (particularly LSTM-RNNs) can improve classical econometric time series models in futures prices forecasting. Our investigation encompassed various data pre-processing procedures and diverse forecast horizons. In general, our findings indicate that LSTM-RNNs perform better than classical models, although the extent of such improvements depends on the specific pre-processing techniques applied and the chosen forecast horizons.

Notably, when trained on the complete series, machine learning models consistently outperform classical models in all comparisons, particularly for longer forecast horizons. This suggests that when the presence of structural breaks may pose challenges in using longer datasets or require complex and subjective modeling efforts, LSTM-RNNs are the preferable choice for long forecast horizons. On the other hand, we stress that this important advantage can potentially be offset by the limited capacity of these techniques to capture either trend or seasonality components when applied to unprocessed data, contrary to what the literature has suggested (Bandara et al., 2019). This implies that the accuracy of forecasts generated by these neural networks is heavily dependent on the pre-processing procedure used. Interestingly, the more traditional First-Difference procedure resulted in lower prediction errors than the modern Full Adjustment procedure. Therefore, to effectively employ advanced forecasting techniques like LSTM-RNNs for non-seasonally, non-trend stationary data, such as those employed in this paper, it is crucial to accurately pre-process all the series using tailored transformations. In our view, this trade-off leans in favor of the machine learning approach as simple transformations such as de-trending, deseasonalization and either first-differencing or full adjustment are typically less demanding than controlling for structural breaks.

Moreover, without prior knowledge of the data-generating process, methods like neural networks can incorporate a larger and more finely tuned number of autoregressive components, moving averages, and other time elements into their structure, which would be exceedingly challenging to achieve with simple linear models. With (futures) price data becoming available at higher frequency rates, allowing for such flexibility could advance the frontier of forecasting methods (e.g., de Boer et al., 2022).

Despite the rather technical nature of the paper, providing reliable forecasts for agricultural futures prices remains an important area of research, given its relevant industry and policy implications. Building a consensus on the most appropriate forecasting methodologies can increase trust, not only by agricultural, food, and financial industries, but also by policymakers and other (non-)governmental actors. This would contribute to a more proper use of forecasting results, especially when decisions must be taken quickly.

Indeed, accurate forecasts can not only inform strategic business decisions, such as the design of insurance instruments and financial investments, but can also support policy analysis (Storm et al., 2023) and guide policy planning by adjusting forecast windows to meet the needs of relevant stakeholders (FAO, 2017). In this respect, one of the main takeaways of our research suggests

that LSTM-RNN can provide a crucial edge in forecasting under turbulent commodity markets, especially in times of increasingly recurrent geopolitical and climatic disruptions. Specifically, the ability of ML forecasting techniques to autonomously handle structural breaks provides a built-in safeguard mechanism against severe model misspecification that would otherwise occur in traditional econometric models. As the frequency of these shocks and their interplay greatly complicate the underlying data-generating process, forecasting through explicit modeling may become a seriously complex exercise.

REFERENCES

Andreasson, P., Bekiros, S., Nguyen, D. K., & Uddin, G. S. (2016). Impact of speculation and economic uncertainty on commodity markets. *International Review of Financial Analysis*, *43*, 115–127. https://doi.org/10.1016/j.irfa.2015.11.005

Bandara, K., Bergmeir, C., & Hewamalage, H. (2020). LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(4), 1586–1599. https://doi.org/10.1109/TNNLS.2020.2985720

Bandara, K., Bergmeir, C., & Smyl, S. (2019). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert Systems with Applications*, *140*, 112–896. https://doi.org/10.1016/j.eswa.2019.112896

Barchart. (2020). Historical data futures. Retrieved from https://www.barchart.com/

Bayona-Oré, S., Cerna, R., & Hinojoza, E. T. (2021). Machine learning for price prediction for agricultural products. *WSEAS Transactions on Business and Economics*, *18*, 969–977. https://doi.org/10.37394/23207.2021.18.92

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 156–166. https://doi.org/10.1109/72.279181

Bishop, C. (2005). *Neural Networks for pattern recognition*. Oxford University Press.

Bojer, C. S., & Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, *37*(2), 587–603. https://doi.org/10.1016/j.ijforecast.2020.07.007

Brandt, J. A., & Bessler, D. A. (1981). Composite forecasting: An application with US hog prices. *American Journal of Agricultural Economics*, *63*(1), 135–140. https://doi.org/10.2307/1239819

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231. https://doi.org/10.1214/ss/1009213726

Brooks, C., Prokopczuk, M., & Wu, Y. (2013). Commodity futures prices: More evidence on forecast power, risk premia and the theory of storage. *The Quarterly Review of Economics and Finance*, *53*(1), 73–85. https://doi.org/10.1016/j.qref.2013.01.003

Carter, C. A., & Mohapatra, S. (2008). How reliable are hog futures as forecasts? *American Journal of Agricultural Economics*, *90*(2), 367–378. https://doi.org/10.1111/j.1467-8276.2007.01122.x

Chen, S. H., Chiou-Wei, S. Z., & Zhu, Z. (2022). Stochastic seasonality in commodity prices: The case of US natural gas. *Empirical Economics*, *62*, 1–22. https://doi.org/10.1007/s00181-021-02094-4

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., & Newey, W. (2017). Double/debiased/neyman machine learning of treatment effects. *American Economic Review*, *107*(5), 261–265. https://doi.org/10.1257/aer.p20171038

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on LOESS. *Journal of Official Statistics*, *6*, 3–73. https://www.nniiem.ru/file/news/2016/stl-statistical-model.pdf

Colino, E. V., Irwin, S., & Garcia, P. (2011). Improving the accuracy of outlook price forecasts. *Agricultural Economics*, *42*(3), 357–371. https://doi.org/10.1111/j.1574-0862.2010.00519.x

Colino, E. V., & Irwin, S. H. (2010). Outlook vs. futures: Three decades of evidence in hog and cattle markets. *American Journal of Agricultural Economics*, *92*(1), 1–15. https://doi.org/10.1093/ajae/aap013

de Boer, T. A., Gardebroek, C., Pennings, J. M., & Trujillo-Barrera, A. (2022). Intraday liquidity in soybean complex futures markets. *Journal of Futures Markets*, *42*(7), 1189–1211. https://doi.org/10.1002/fut.22325

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, *74*, 427–431. https://doi.org/10.1080/01621459.1979.10482531

Efron, B. (2020). Prediction, estimation, and attribution. *Journal of the American Statistical Association*, *115*, 636–655. https://doi.org/10.1111/insr.12409

Egelkraut, T. M., Garcia, P., & Sherrick, B. J. (2007). The term structure of implied forward volatility: Recovery and informational content in the corn options market. *American Journal of Agricultural Economics*, *89*(1), 1–11. https://doi.org/10.1111/j.1467-8276.2007.00958.x

FAO – Food and Agricultural Organization of the United Nations. (2017). *What did we learn from the bout of high and volatile food commodity prices (2007-2013)? FAO Commodity and Trade Policy Research Working Paper Series*. No. 54. https://www.proquest.com/working-papers/what-did-we-learn-bout-high-volatile-food/docview/1932256342/se-2

Gers, F., Schraudolph, N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, *3*, 115–143. https://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf

Gilliland, M. (2020). The value added by machine learning approaches in forecasting. *International Journal of Forecasting*, *36*(1), 161–166. https://doi.org/10.1016/j.ijforecast.2019.04.016

Goodwin, P. (2020). Performance measurement in the M4 competition: Possible future research. *International Journal of Forecasting*, *36*(1), 189–190. https://doi.org/10.1016/j.ijforecast.2019.02.015

Greene, W. (2020). *Econometric analysis* (8th ed.). Pearson.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The elements of statistical learning: Data mining, inference, and prediction. *Springer series in statistics* (2nd ed.). Springer New York, NY.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Hoffman, L. A., Etienne, X. L., Irwin, S. H., Colino, E. V., & Toasa, J. I. (2015). Forecast performance of WASDE price projections for

U.S. corn. *Agricultural Economics*, *46*, 157–171. https://doi.org/10.1111/agec.12204

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, *36*(1), 7–14. https://doi.org/10.1016/j.ijforecast.2019.03.015

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Johansen, S. (1995). *Likelihood-based inference in cointegrated vector autoregressive models*. Oxford University Press.

Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of Econometrics*, *54*, 159–178. https://doi.org/10.1016/0304-4076(92)90104-Y

Leuthold, R. M., MacCormick, A. J. A., Schmitz, A., & Watts, D. G. (1970). Forecasting daily hog prices and quantities: A study of alternative forecasting techniques. *Journal of the American Statistical Association*, *65*(329), 90–107. https://doi.org/10.1080/01621459.1970.10481064

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer.

Mahto, A. K., Alam, M. A., Biswas, R., Ahmad, J., & Alam, S. I. (2021). Short-term forecasting of agriculture commodities in context of Indian market for sustainable agriculture by using the artificial neural network. *Journal of Food Quality*, *2021*, 1–13. https://doi.org/10.1155/2021/9939906

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4), 1346–1364. https://doi.org/10.1016/j.ijforecast.2021.11.013

Mtibaa, F., Nguyen, K. K., Azam, M., Papachristou, A., Venne, J. S., & Cheriet, M. (2020). LSTM-based indoor air temperature prediction framework for HVAC systems in smart buildings. *Neural Computing and Applications*, *32*(23), 17569–17585. https://doi.org/10.1007/s00521-020-04926-3

Mullainathan, S., & Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, *31*(2), 87–106. https://doi.org/10.1257/jep.31.2.87

OECD. (2000). *Income risk management in agriculture*. OECD Publishing.

Ouyang, H., Wei, X., & Wu, Q. (2019). Agricultural commodity futures prices prediction via long-and short-term time series network. *Journal of Applied Economics*, *22*(1), 468–483. https://doi.org/10.1080/15140326.2019.1668664

RL, M., & Mishra, A. K. (2021). Forecasting spot prices of agricultural commodities in India: Application of deep-learning models. *Intelligent Systems in Accounting, Finance and Management*, *28*(1), 72–83. https://doi.org/10.1002/isaf.1487

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408. https://psycnet.apa.org/doi/10.1037/h0042519

Ruder, S. (2016). An overview of gradient descent optimization algorithms. ArXiv, abs/1609.04747. https://doi.org/10.48550/arXiv.1609.04747

Sanders, D. R., & Irwin, S. H. (2016). The "Necessity" of new position limits in agricultural futures markets: The verdict from daily firm-level position data. *Applied Economic Perspectives and Policy*, *38*(2), 292–317. https://doi.org/10.1093/aepp/ppv019

Serra, T., Zilberman, D., Gil, J. M., & Goodwin, B. K. (2011). Nonlinearities in the US corn-ethanol-oil-gasoline price system. *Agricultural Economics*, *42*(1), 35–45. https://doi.org/10.1111/j.1574-0862.2010.00464.x

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, *104*(1), 148–175. https://doi.org/10.1109/JPROC.2015.2494218

Sheikh, R., & Jahirabadkar, S. (2018). An insight into theory-guided climate data science—A literature review. *Advances in Data and Information Sciences: Proceedings of ICDIS-2017*, *1*, 115–125. https://doi.org/10.1007/978-981-10-8360-0_11

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, *404*, 132–306. https://doi.org/10.1016/j.physd.2019.132306

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*(1), 75–85. https://doi.org/10.1016/j.ijforecast.2019.03.017

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems, *25*, 106–120. https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM–a tutorial into Long Short-Term Memory Recurrent Neural Networks. arXiv, abs/1909.09586. https://doi.org/10.48550/arXiv.1909.09586

Storm, H., Baylis, K., & Heckelei, T. (2020). Machine learning in agricultural and applied economics. *European Review of Agricultural Economics*, *47*(3), 849–892. https://doi.org/10.1093/erae/jbz033

Storm, H., Heckelei, T., Baylis, K., & Mittenzwei, K. (2023). Identifying farmers' response to changes in marginal and average subsidies using deep learning. *American Journal of Agricultural Economics*, https://doi.org/10.1111/ajae.12442

Suh, D. H., & Moss, C. B. (2017). Decompositions of corn price effects: Implications for feed grain demand and livestock supply. *Agricultural Economics*, *48*(4), 491–500. https://doi.org/10.1111/agec.12350

Taieb, S. B., Sorjamaa, A., & Bontempi, G. (2010). Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing*, *73*(10-12), 1950–1957. https://doi.org/10.1016/j.neucom.2009.11.030

Timmermann, A. (2006). Forecast combinations. *Handbook of Economic Forecasting*, *1*, 135–196.

Trujillo-Barrera, A., Garcia, P., & Mallory, M. L. (2016). Price density forecasts in the US hog markets: Composite procedures. *American Journal of Agricultural Economics*, *98*(5), 1529–1544. https://doi.org/10.1093/ajae/aaw050

Tyner, W. E. (2010). The integration of energy and agricultural markets. *Agricultural Economics*, *41*, 193–201. https://doi.org/10.1111/j.1574-0862.2010.00500.x

United States Department of Energy. (2023). Alternative Fuels Data Centre, Maps and Data. https://afdc.energy.gov/data/10355

Verbeek, M. (2017). *A guide to modern econometrics* (5th ed.). Wiley.

Wang, D., Yue, C., Wei, S., & Lv, J. (2017). Performance analysis of four decomposition-ensemble models for one-day-ahead agricultural commodity futures price forecasting. *Algorithms*, *10*(3), 108. https://doi.org/10.3390/a10030108

Wei, A., & Leuthold, R. M. (2000). Agricultural Futures Prices and Long Memory Processes. *OFOR Working Paper.* No. 00.04. https://dx.doi.org/10.2139/ssrn.229795

Xu, X. (2020). Corn cash price forecasting. *American Journal of Agricultural Economics*, *102*, 1297–1320. https://doi.org/10.1002/ajae.12041

Xu, X., & Zhang, Y. (2021a). Individual time series and composite forecasting of the Chinese stock index. *Machine Learning with Applications*, *5*, 100035. https://doi.org/10.1016/j.mlwa.2021.100035

Xu, X., & Zhang, Y. (2021b). Corn cash price forecasting with neural networks. Computers and Electronics in Agriculture, *184*. https://doi.org/10.1016/j.compag.2021.106120

Xu, X., & Zhang, Y. (2022). Commodity price forecasting via neural networks for coffee, corn, cotton, oats, soybeans, soybean oil, sugar, and wheat. *Intelligent Systems in Accounting, Finance and Management*, *29*(3), 169–181. https://doi.org/10.1002/isaf.1519

Zeileis, A., Kleiber, C., Krämer, W., & Hornik, K. (2003). Testing and dating of structural changes in practice. *Computational Statistics & Data Analysis*, *44*(1–2), 109–123. https://doi.org/10.1016/S0167-9473(03)00030-6

Zeileis, A., Leisch, F., Hornik, K., & Kleiber, C. (2002). strucchange: An R package for testing for structural change in linear regression models. *Journal of Statistical Software*, *7*(2), 1–38. https://doi.org/10.18637/jss.v007.i02

Zhao, H. (2021). Futures price prediction of agricultural products based on machine learning. *Neural Computing and Applications*, *33*, 837–850. https://doi.org/10.1007/s00521-020-05250-6

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Brignoli, P. L., Varacca, A., Gardebroek, C., & Sckokai, P. (2024). Machine learning to predict grains futures prices. *Agricultural Economics*, *55*, 479–497. https://doi.org/10.1111/agec.12828