



# Anticipating food price crises by reservoir computing

L. Domingo<sup>a,b,c</sup>, M. Grande<sup>c,d</sup>, F. Borondo<sup>a</sup>, J. Borondo<sup>d,e,\*</sup>

<sup>a</sup> Departamento de Química, Universidad Autónoma de Madrid, CANTOBLANCO, 28049 Madrid, Spain

<sup>b</sup> Instituto de Ciencias Matemáticas (ICMAT), Campus de Cantoblanco UAM, Nicolás Cabrera, 13-15, 28049 Madrid, Spain

<sup>c</sup> Grupo de Sistemas Complejos, Universidad Politécnica de Madrid, 28035, Madrid, Spain

<sup>d</sup> AgrowingData, 04001 Almería, Spain

<sup>e</sup> Departamento de Gestión Empresarial, Universidad Pontificia de Comillas ICADE, Alberto Aguilera 23, 28015 Madrid, Spain

## ARTICLE INFO

### Keywords:

Machine learning  
Reservoir Computing  
Time series prediction  
Food loss  
Sustainability  
Artificial Intelligence

## ABSTRACT

Anticipating price crises in the market of agri-commodities is critical to guarantee both the sustainability of the food system and to ensure food security. However, this is not an easy task, since the problem implies analyzing small and very volatile time series, which are highly influenced by external factors. In this paper, we show that suitable reservoir computing algorithms can be developed that outperform traditional approaches, by reducing the Mean Absolute Error and, more importantly, increasing the Market Direction Accuracy. For this purpose, the applicability of five variants of such method to forecast this market is explored, and their performance evaluated by comparing the results with those obtained with the standard LSTM and SARIMA benchmarks. We conclude that decomposing the time series and modeling each component with a separate RC is essential to successfully anticipate price trends, and that this method works even in the complex changing temporal scenario of the Covid-19 pandemic, when part of the data were collected.

## 1. Introduction

Understanding the agri-food industry [1–4] is becoming increasingly more important everyday, especially in connection with making food systems sustainable and more equitable as indicated by the United Nations development goals. This is even more urgent in the times of food insecurity that we are living, due to the climate change and other related factors [5]. Indeed, the agri-food industry is very relevant in the European global market [6], particularly in Spain [7] where it still remains a fundamental pillar of the national economy.

Seeking sustainability, many innovative technologies are being integrated in the sector [2,3,8]. Most of them focus on increasing productivity and developing solutions linked to precision farming, in order to make the production system more efficient. However, the food system cannot be sustainable without reducing food waste [9,10]. To achieve this for fruits and vegetables is key to understand the dynamics of the market, as the main reason of this waste are low prices leaving no sale margin. In fact, year in and year out we observe price crises in Europe where the commercialization of some products is not profitable, which automatically translates into tons of thrown away food.

Traditionally, research on agricultural prices forecast has heavily relied on Seasonal Autoregressive Integrated Moving Average (SARIMA)

models [11,12]; see, for example, Refs. [13–15]. Other studies, however, have focused in applying neural networks. Some examples can be found in Ref. [16] where such method was used to simulate China's food security early warning system, or in Ref. [17] where four such models were developed to predict China vegetable market prices [18]. More recently, ARIMA and a neural network were compared [19], concluding that overall the latter achieves higher performance. However, the issue if the models can anticipate or not the direction of the market it is not properly addressed in this type of studies.

The main challenge in the prediction of agri-commodity prices, which has severely limited previous research, is that markets (consider for example the European) are indeed highly connected networks, where prices are influenced by many factors, such as climate, crop productions, or exports, just to name a few of the most relevant. Thus, to obtain accurate models, all these factors have to be taken into account. At the same time, another important challenge to overcome is that the highest time resolution available for fruits and vegetables prices is daily, something that severely limits the size of the training samples.

In this respect, we believe that Reservoir Computing (RC), a recently introduced computational framework that has proven very useful in predicting the evolution of time series [23], such as weather forecast,

\* Corresponding author at: Departamento de Gestión Empresarial, Universidad Pontificia de Comillas ICADE, Alberto Aguilera 23, 28015 Madrid, Spain.

E-mail addresses: [laia.domingo@icmat.es](mailto:laia.domingo@icmat.es) (L. Domingo), [mar.grande@alumnos.upm.es](mailto:mar.grande@alumnos.upm.es) (M. Grande), [f.borondo@uam.es](mailto:f.borondo@uam.es) (F. Borondo), [jborondo@comillas.edu](mailto:jborondo@comillas.edu) (J. Borondo).

<https://doi.org/10.1016/j.chaos.2023.113854>

Received 10 May 2023; Accepted 15 July 2023

Available online 3 August 2023

0960-0779/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

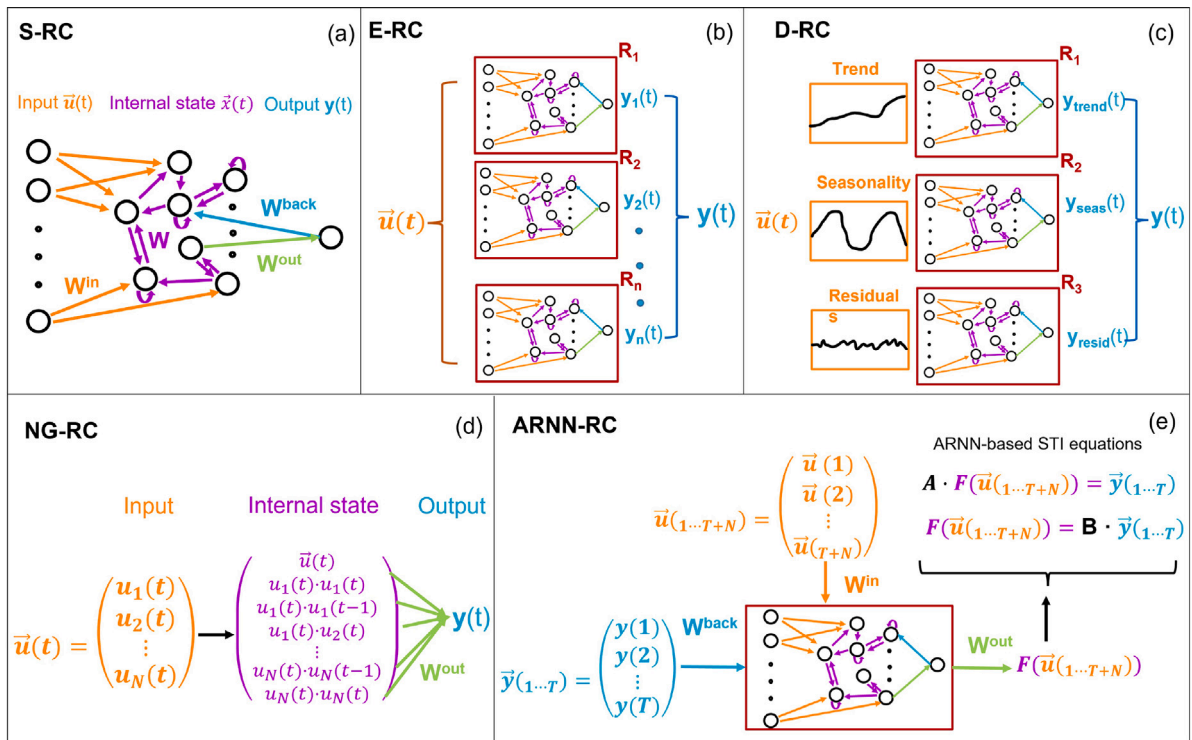


Fig. 1. Schematic architecture of the five reservoir computing based models studied in this work: (a) represents the original or standard reservoir computing (S-Rc) model introduced in Ref. [20], (b) shows an ensemble of reservoir computing (E-Rc) models, whose outputs are averaged to construct the final prediction, in (c) the time series is decomposed (D-Rc) into three parts: trend, seasonality and residuals, and then model E-Rc is used to predict each of the series components, (d) is the next generation reservoir computing (NG-Rc) model presented in Ref. [21], which only uses linear and non-linear functions of the output to predict its dynamics, finally, (e) represents the auto recurrent neural network reservoir computing (ARNN-Rc) model presented in Ref. [22], which obtains the predictions by mapping the future high-dimensional regressor variables to the associated time series prediction.

stock indices, or the dynamics of chaotic systems [24,25], is especially useful in this kind of settings and can outperform other more traditional approaches. For one thing, RC uses both the past value of the time series and regressor variables to extrapolate the time series. Moreover, the simple training framework of RC reduces the chances of overfitting when small training data, as it is the case here, are used. The concept of RC was introduced by Jaeger as Echo State Network [20] to improve the computationally expensive training of recurrent neural networks [26]. This author also proved [20,27] that as long as the reservoir fulfills certain properties, only training the output layer is enough to obtain excellent performance in many tasks.

Currently, price time series forecasting is an important and complex area of machine learning. The fact that the evolution of prices has a time component adds additional information, but it also implies dealing with new complexities, not present in other prediction tasks. In fact, this complexity not only involves the training process, but also, and more importantly, the evaluation of the quality of the predictions. Since prices always tend to be autocorrelated somehow in time, a model can output the price at time  $t-1$  as the prediction for the price at time  $t$  and still having a low error metric. Such models are obviously useless since they cannot anticipate changes in the direction of the market. Although this fact is extremely important when addressing price prediction, it has been often ignored in the scientific literature. In the present work, we overcome this problem by evaluating our models not only using the mean absolute error (MAE), but checking also their predictive power for the Market Direction Accuracy (MDA).

Anticipating the direction of the market is not only important in the food market, but also when forecasting many other markets, such as stocks [28,29] or cryptocurrencies [30,31], since the actors, i.e. the investors, operating in such markets always adopt trading strategies in which the return depends on the direction of the future foreseen motions of the market. Hence, a model rendering minimal MAE but low

MDA totally lacks interest. As an striking illustration of this, we will show that Long-Short Term Memory (LSTM), which does not consider external factors, is unable to anticipate the direction of the zucchini market, giving results which are only barely better than pure random guessing. In contrast, our most accurate RC based method [32,33] achieves an ~80% MDA, this making of it an excellent and profitable protocol to operate in this environment.

In this paper, we assess the performance of five variants of the recently introduced RC model in their applicability to the agri-food industry, by comparing their results with those obtained with two widespread benchmark algorithms in the field, namely SARIMA and LSTM. The main conclusion of this comparison is twofold. First, RC clearly outperforms the benchmark models on predicting price time series, especially in anticipating sudden changes in their tendency. Second, we provide an optimal RC architecture, based on decomposing the time series, to anticipate the evolution of food prices under the conditions prevailing in the agri-food market.

## 2. Results

In this section, we discuss the performance of different RC based models, whose operations are schematically illustrated in Fig. 1 (see also full description in section Methods below) compared with the two benchmark models, SARIMA and LSTM. In the training stage, a data set consisting of the daily prices at origin in Spain between 2013–2021 of three common vegetable varieties, i.e. zucchini, aubergine and tomato, is used. The final performance is evaluated using the test set campaign 2021–2022. (See section Dataset below.) All prices are given in €/kg throughout the paper.



Fig. 2. Prediction of the zucchini price time evolution in the test set 2021–2022 campaign for: (a) the five reservoir computing based methods considered in this work, and (b) the best reservoir computing method compared with SARIMA and LSTM. The actual price times series is shown in black line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 2.1. Qualitative analysis and main result

The corresponding results are shown in Fig. 2, where the predictions (colored lines) are compared with the actual time series (black line) for the zucchini prices in the data set. To aid in the interpretation, the computed results have been separated in two parts. In Fig. 2(a), we present a comparison among the results produced by our five RC based algorithms, while in Fig. 2(b), we present the comparison of the results obtained with our best RC based method and those rendered by the benchmark models.

A qualitative eye examination of the results in Fig. 2(a) shows that among the RC models, the D-RC (cyan line) is the one having the best performance. This method decomposes the signal into three parts: *trend*, *seasonality*, and *residuals* and then predicts each one separately (see Fig. 1(c) and Methods below), this being the key reason of its excellent performance. More interestingly, we also see in the comparison of Fig. 2(b) that D-RC clearly outperforms the standard LSTM and SARIMA algorithms (yellow and green lines). In particular: f (i) The D-RC algorithm is the one predicting more accurately the market falls. To emphasize this result, we have marked with red triangles those points in the data set (black line) where the price dropped to minimal values.

(ii) More importantly, in the regions near the minima the LSTM and the SARIMA models predict higher prices than the reals, while the D-RC is able to correctly capture the corresponding changes in tendency

(iii) In the region of high price volatility (highlighted as a shaded region in the plot), where the behavior of the time series significantly differs from the rest, the D-RC model is the only one able to make accurate predictions, while SARIMA forecasts prices with very small variations over time, and LSTM, despite predicting the highly volatility, appears largely uncorrelated with the actual prices.

As a result, we conclude that D-RC is the only forecasting model which captures the dynamics of the time series during the whole testing period.

### 2.2. Forecasting the market: Mean absolute error

Let us make now our analysis more quantitative by discussing the error metrics for each of the seven models and the three vegetables considered in this work. We begin by calculating the MAE of the predictions in the training, validation and test sets. To keep the narrative simpler, we focus on analyzing only the errors observed in the test sample, which results are given in Fig. 3(a). Nevertheless the metrics for training and validation are also provided in Table 1. As can be seen, the results clearly show that the best predicting model is D-RC, presenting this configuration significant lower errors in all cases. Actually, the MAE for the three vegetable varieties is always under 0.10, being the D-RC the only model breaking down this barrier.

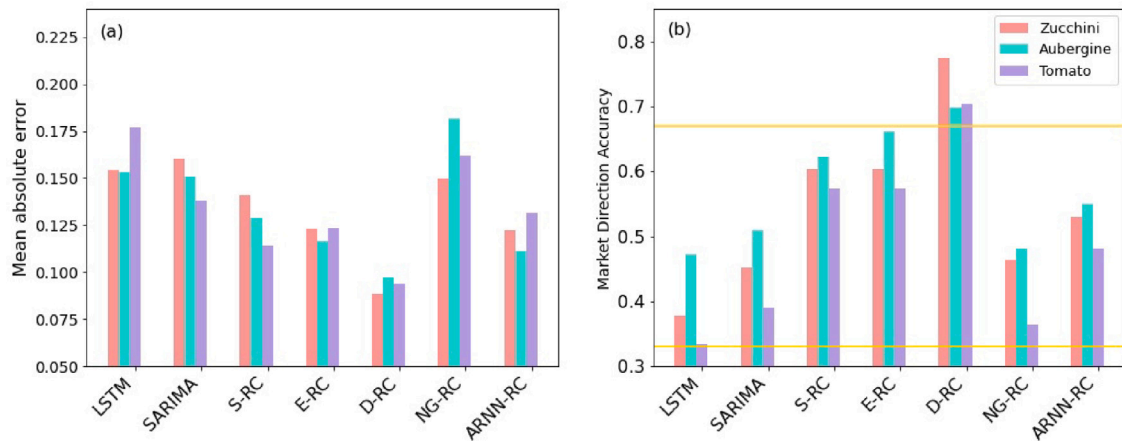
Next, we find that E-RC and S-RC both show a similar performance, having these two models slightly higher errors, ranging between 0.11–0.13. However, we have found that the E-RC is much more robust to the choice of the random matrices, and therefore its performance is much less prone to oscillations for different realizations of the algorithm.

Overall, the third best model is ARNN-RC, whose error in the test set is similar to the MAEs of E-RC and S-RC, even though for validation the errors are slightly higher. Note that by construction, in this method there is no training prediction, and thus the corresponding value does not appear in Table 1.

In contrast to the other RC configurations, the NG-RC exhibits a similar performance to that of the benchmark models, LSTM and SARIMA. Taking tomato as an example, NG-RC presents an error of 0.16, which is between those of SARIMA (0.14) and LSTM (0.18), and it is 78% higher than for the best model, i.e. D-RC.

### 2.3. Forecasting the direction of the market: Market direction accuracy

Most series describing the evolution of real data do not exhibit large variations between consecutive time steps, thus presenting a significant degree of autocorrelation. In fact, despite their high volatility, agri-food price time series are no exception. As a consequence, a common mistake



**Fig. 3.** Mean absolute error (a) and market direction accuracy (b) in the test set of zucchini, aubergine, and tomato time series for the seven models studied in this work. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Table 1**

Mean absolute error and market direction accuracy for the training, validation and test data sets for the seven machine learning models used in predicting the zucchini, aubergine and tomato prices series.

Model	Z U C C H I N I						A U B E R G I N E						T O M A T O					
	MAE			MDA			MAE			MDA			MAE			MDA		
	Train	Valid.	Test	Train	Valid.	Test	Train	Valid.	Test	Train	Valid.	Test	Train	Valid.	Test	Train	Valid.	Test
LSTM	0.217	0.199	0.154	0.414	0.455	0.377	0.185	0.194	0.153	0.427	0.364	0.372	0.092	0.127	0.177	0.502	0.500	0.333
SARIMA	0.124	0.213	0.16	0.505	0.523	0.453	0.175	0.214	0.151	0.481	0.5	0.509	0.107	0.124	0.138	0.454	0.545	0.389
S-RC	0.123	0.214	0.141	0.586	0.545	0.604	0.107	0.171	0.129	0.695	0.705	0.623	0.088	0.106	0.114	0.58	0.636	0.574
E-RC	0.128	0.166	0.123	0.617	0.705	0.604	0.134	0.157	0.117	0.644	0.636	0.66	0.086	0.105	0.123	0.58	0.727	0.574
D-RC	0.135	0.132	0.089	0.563	0.795	0.774	0.12	0.141	0.09	0.654	0.659	0.698	0.091	0.094	0.093	0.614	0.659	0.704
NG-RC	0.167	0.182	0.150	0.476	0.605	0.463	0.147	0.180	0.182	0.529	0.535	0.481	0.097	0.128	0.162	0.529	0.488	0.364
ARNN-RC	–	0.197	0.122	–	0.659	0.529	–	0.185	0.121	–	0.545	0.549	–	0.126	0.131	–	0.545	0.481

of many machine learning models in this scenario, that is not captured by indicators such as MAE,  $R^2$ , or the relative error, is to use the current value as a reasonable prediction for the next time step. Indeed, under these circumstances, just by copying the previous price a model can still achieve a low error metric. For this reason, we need to further evaluate our models using complementary metrics, that are robust against this autocorrelation. As one such adequate metrics, we propose here MDA, since it checks if a model can correctly predict a change of tendency in the price time series.

Our way to implement this metrics is the following. First, we consider that prices have increased or decreased within consecutive weeks if the price change is greater than  $\pm 0.05$ , respectively. Otherwise, we say that the prices have remained constant. Hence, the target function takes a value of 1 to indicate an uptrend,  $-1$  to indicate a downtrend, and 0 when the price remains constant. Next, we measure the MDA of the models predicting the new categorical target. The results are shown in Fig. 3(b) and Table 1. To analyze these results we have to take into account that our target function has three labels, and then assuming the same probability for all classes, a random model would guess correctly (on average) one third of the times, this leading to a 33% accuracy. This value (as well as 67% accuracy) has been added, as a yellow line, in Fig. 3(b) to guide the eye.

As can be seen, all results are above the random accuracy, and also here the D-RC is the one obtaining the best scores, rendering an accuracy of almost 80% for zucchini and 70% for tomato and aubergine. More interesting, is that in the case of zucchini, the D-RC method correctly anticipates changing trends in 77.4% of the cases, compared to the poor 45.3% of SARIMA or 37.7% for LSTM. Accordingly, we can conclude that the performance of this RC based method is quite impressive, since it gets 2.1 and 1.7 more hits than the widespread LSTM and SARIMA, respectively, for the prediction metric considered in this section. Moreover, it must be noticed that this effect is not

particular to zucchini, since similar results are obtained for aubergine and tomato, as can be ascertain from the results reported in Table 1.

The second-best models are again E-RC and S-RC, which have similar accuracy on the test set, even though, as expected, E-RC is slightly better. The MDA for both models are in general over 10 points behind the performance of D-RC.

According to this complementary MDA metric, the ARNN-RC and NG-RC perform also much better than LSTM, and slightly better than SARIMA. Also, ARNN-RC has higher accuracy than NG-RC, being the third-best model in terms of MDA as well. At this point, it should be emphasized that in the original paper for NG-RC [21], the authors evaluated the model for multi-dimensional time series. The predictor vectors contain time-delayed products of the time series components. In this work, the time series is just one-dimensional and therefore there are less time-delayed products to predict the time evolution, which may be the cause of having higher errors in the test set.

#### 2.4. Forecasting during the covid-19 pandemic

One final interesting point of our results is that all models perform worse, in terms of MAE, in the validation than in the test set. The most likely reason for this behavior is that the former contains the data from 2020, which are fairly different from the training data due to the start of the global Covid-19 pandemic. Indeed, prices present an abrupt high increment around March 2020, unrelated to any previous behavior of the series or the regressor variables. This explains why it is more difficult to predict the time series evolution during this period. However, we also noticed that in terms of MDA the RC models do not perform significantly worse in the validation set. Our interpretation for this divergence is that even though the reservoir cannot predict such extremely high prices (which produce a high MAE) since they never occurred in the past, it is still capable of following the trend of the signal, leading to high accuracy when predicting price rises and falls.



### 3. Discussion

Sustainability is one of the key goals in United Nations' Agenda 2030. However, our food system cannot be made sustainable [1–3] unless the current food loss and waste problem is first tackled. This is particularly critical for fresh products, which are not suitable for long-term storage. One of the biggest issues regarding food loss is that around 14% of the food produced in the world is lost between harvest and retail [10]. This is most often caused by price crises, where the price is so low that marketing the product is no longer profitable. Accordingly, to achieve food sustainability is key to anticipate prices, so that proper actions can be deployed before prices drop. To shed light into this problem, we have analyzed the evolution of zucchini, aubergine and tomato prices in the sud-east of Spain, which is the largest supplier region to the European market, during 2013–2022. Our results show that the novel RC framework [23–25] can contribute to adequately understand the dynamic of this market, certainly outperforming the widespread SARIMA and LSTM.

The time series studied in this work, that are paradigmatic of the agri-food market, present a high volatility, are influenced by external factors, and are small in size. This scenario makes our series hard to predict, since we need a model that captures the complex patterns underlying the time series, that adequately considers the effect of external factors, and, that is robust against overfitting. Indeed, not overfitting in this scenario is a very relevant challenge, since the number of learning parameters may be of the same order of magnitude as the training size. Thus, solving this challenge brings together a good added bonus to our work since, once solved,

In this work, we have compared different RC variants and showed the benefit of decomposing the time series and modeling each component separately. First, we have trained the original RC model presented in Ref. [20]. In our case, this configuration is limited by the small training set, as the S-RC model highly depends on the initialization of the random matrices and other hyperparameters. To reduce this effect, we introduced a second model, i.e. E-RC, consisting of an ensemble of reservoirs, that significantly increases the stability and to a less extent the performance. However, we found that the models is still limited: Indeed, since prices do not vary much from one week to another, the time series models are likely to copy the previous value of the time series in the prediction. To solve this problem, we next proposed another configuration of the RC, i.e. the D-RC, where we decompose the series into three parts: trend, seasonality and residuals, and model each one them by RC separately. We conclude, that separating the three components greatly helps to reduce the MAE and, more significantly, it increases the MDA. Indeed, this configuration clearly exhibits the lowest errors among all tested methods. In addition, we also considered the two novel RC approaches reported in Refs. [21,34]. The main advantage of the first one, the NG-RC, is that requires no random matrices and fewer hyperparameters. Although this configuration is shown to perform at the same level than S-RC when predicting the evolution of chaotic dynamical systems, we find that in the agri-food scenario it clearly underperforms. For example, for tomatoes it presents a MAE of 0.16 which is 45% larger than for S-RC. The second, ARNN-RC, method is based on predicting multiple-step-ahead predictions, using the regressor variables as the reservoir. For this variant, a similar performance to S-RC is found.

One of the main conclusions of our work is that the MDA indicator clearly and explicitly highlights the difference in performance between models. For example, LSTM, with a MAE of 0.15 for aubergine, could have been evaluated as a fairly efficient model; however its MDA drops to a low 37%, which is almost equivalent to making predictions by random guessing. In fact, the difference in performance between the best RC model and the benchmark models is significantly larger when evaluated under the prescription of MDA. To illustrate this fact, let us put some numbers taken from the case of zucchini. The estimated MAE here is 0.15 for LSTM and 0.16 for SARIMA, while D-RC presents

a value of just 0.09. This means, that the error is 73% larger for LSTM and 80% for SARIMA in comparison to our best RC. Moreover, and as mentioned above, this difference in percentage is more pronounced for the MDA indicator. In this case, we find that the error observed with respect to D-RC, is 176% higher for LSTM and 142% for SARIMA. Hence, we can argue that when proposing new methodologies to forecast time series, or comparing the existing methods, in order to properly evaluate them, it is not only necessary to look at typical error metrics, such as MAE or  $R^2$ , but also to consider their accuracy in anticipating the direction of the series. This issue, is specifically critical when forecasting markets, where the actors adopt a trading strategies whose return is primarily determined by the direction of the future motions.

### 4. Dataset

The present work is based on data concerning the prices at origin in Spain of three different vegetable varieties: zucchini, aubergine and tomato. These prices are published daily after the different auctions, and refer to the price per kilogram. In order to build up our time series, we have aggregated the prices of each variety weekly, coming up with the average price of the week. We do so, because weekly predictions are more representative, then valuable, than the daily ones for the agri-food market. Our dataset includes prices from March 2013 to March 2022. This time series is split into training, validation and test sets. The training set contains the prices until December 2019, resulting in 356 data points. The validation set contains the prices of 2020 (53 data points), and the test set contains the prices from January 2021 onwards (62 data points). Therefore, the training data has few number of data points, making hard the prediction without doing overfitting. When training the models, the time series is standardized to the  $[-1, 1]$  domain using a linear scale. The validation set is used to select the optimal hyperparameters of each model. The test set is then used to evaluate the performance of the selected models on unseen data. Apart from the time series, the RC models used in this work also use 16 regressor variables to predict the prices time evolution. Such variables gather information about production volumes and international trade that provide complementary information for the price prediction.

### 5. Methods

In this section, we present details of the machine learning models used in this study. In particular, we use two standard benchmark time series prediction models: a LSTM neural network [35] and a SARIMA [11], the standard RC (S-RC), and the four variations developed by us (E-RC, D-RC, NG-RC, and ARNN-RC). See descriptive schematic illustrations and acronym definitions in Fig. 1. We also report the numerical values of the hyperparameters used.

#### 5.1. Standard reservoir computing

In the S-RC framework [20], the learning complexity of the algorithm is reduced to performing a simple linear regression, the key point being to design a dynamical system able to learn the input–output dynamics. In this case, the dynamical system is a neural network, whose internal states are called *echo states*, since they can be thought of as echoes of their past [20]. The typical structure of the network is shown schematically in Fig. 1(a), where

- $W^{\text{in}} \in M(\mathbb{R})_{N \times K}$  represents the weights from the input units to the internal states,
- $W \in M(\mathbb{R})_{N \times N}$  represents the weights between the different internal states,
- $W^{\text{out}} \in M(\mathbb{R})_{L \times N}$  represents the weights from internal states to output units, and
- $W^{\text{back}} \in M(\mathbb{R})_{N \times L}$  represents the weights from the output units to the internal states,

also

- $\tilde{u}(t) = (u_1(t), u_2(t), \dots, u_K(t))$  is a  $K$ -dimensional vector giving the input units at time  $t$ ,
- $\tilde{x}(t) = (x_1(t), x_2(t), \dots, x_N(t))$  is an  $N$ -dimensional vector giving the internal states at time  $t$ , and
- $\tilde{y}(t) = (y_1(t), y_2(t), \dots, y_L(t))$  is an  $L$ -dimensional vector giving the output units at time  $t$ . In our setting  $L = 1$  and so, from now on, we will refer to the output as  $y(t)$ .

The matrices  $W^{\text{in}}$ ,  $W$  and  $W^{\text{back}}$  are fixed, and thus they do not change during the training phase. The only learnable parameters are the weights  $W^{\text{out}}$ . The steps to train the echo state network are the following:

1. Generate the reservoir matrices ( $W$ ,  $W^{\text{in}}$ ,  $W^{\text{back}}$ ) randomly.
2. Update the internal states by teacher forcing:

$$\begin{aligned}\tilde{x}(t) &= f[W^{\text{in}}\tilde{u}(t) + W\tilde{x}(t-1) + W^{\text{back}}y_{\text{teach}}(t-1)] \\ \tilde{x}(t) &= (1-\alpha)\tilde{x}(t-1) + \alpha\tilde{x}(t)\end{aligned}\quad (1)$$

where  $y_{\text{teach}}$  is the output that we want our network to predict,  $\alpha \in (0, 1]$  is the leaking rate and  $f$  is the activation function. Usually,  $f(\cdot) = \tanh(\cdot)$ , which is applied component-wise.

3. Discard a transient of  $t_{\text{min}}$  states to guarantee the convergence of the reservoir dynamics.
4. Find the readout matrix  $W^{\text{out}}$  by minimizing the mean squared error, (if necessary) with regularization,

$$\begin{aligned}MSE_r(y, y_{\text{teach}}) &= \frac{1}{T - t_{\text{min}}} \sum_{t=t_{\text{min}}}^T \left[ f^{\text{out}-1}(y_{\text{teach}}(t)) \right. \\ &\quad \left. - W^{\text{out}}\tilde{x}(t) \right]^2 + \gamma \|W^{\text{out}}\|^2.\end{aligned}\quad (2)$$

where  $f^{\text{out}}$  is the output activation function (usually  $f^{\text{out}} = Id$ ). Notice that this step only requires performing a ridge regression.

The steps to make the predictions after training the network are the following:

1. Given an input  $u(t)$ , update the state of the reservoir:

$$\begin{aligned}\tilde{x}(t) &= f[W^{\text{in}}\tilde{u}(t) + W\tilde{x}(t-1) + W^{\text{back}}y(t-1)] \\ \tilde{x}(t) &= (1-\alpha)\tilde{x}(t-1) + \alpha\tilde{x}(t)\end{aligned}\quad (3)$$

where  $y(t-1)$  is the prediction of the output at time  $t-1$ .

2. Compute the prediction  $y(t)$  as

$$y(t) = f^{\text{out}}[W^{\text{out}}\tilde{x}(t)].\quad (4)$$

Let us explain now these steps in more detail. To ensure that the RC model works, the internal network must fulfill the *echo state properties*. Roughly speaking, this means that the internal states must have the state and input forgetting property; that is, for large enough  $t$ ,  $\tilde{x}(t)$  should not depend on  $\tilde{x}(0)$ ,  $\tilde{u}(0)$  or  $y_{\text{teach}}(0)$ . For this reason, in step 2 of the training phase, we dismiss the initial values of the internal states, which could be influenced by the initial parameters of the reservoir. The spectral radius  $\rho(W)$  of the internal matrix  $W$  may also influence the performance of the method: Larger values of  $\rho(W)$  lead to a longer memory. In fact, it is a sufficient condition that  $\rho(W) < 1$  for having echo states, even though reservoirs with  $\rho(W)$  slightly larger than 1 may also give optimal results [20]. Also, matrix  $W$  should be sparse and inhomogeneous, so that the internal states contain a set of diverse trajectories. On the other hand, the leaking rate  $\alpha$  influences the velocity of the output dynamics. A fast-changing output should be trained with greater values of  $\alpha$ . Once the internal states have been computed, the learning phase only consists of training a linear model to find the mapping  $W^{\text{out}}$  from the internal states to the output. This linear model is usually obtained by ridge regression, which is used to reduce overfitting during training. Finally, notice that the RC model can also be used without the input layer when we aim to predict a time

**Table 2**

Optimal hyperparameters used to train the different Reservoir Computing models:  $\alpha$  is the leaking rate,  $N$  the number of neurons of the reservoir,  $\gamma$  is the regularization parameter,  $\rho(W)$  and  $D(W)$  the spectral radius and density of  $W$ , and  $N_e$  the number of reservoirs used in the ensemble.

Model	$\alpha$	$N$	$\gamma$	$\rho(W)$	$D(W)$	$N_e$
S-RC	0.81	120	2.6	0.52	0.012	–
E-RC	0.98	80	0.51	1.4	0.011	240
D-RC (trend)	0.58	60	0.011	1.1	0.016	80
D-RC (seasonality)	0.74	20	0.011	0.32	0.023	100
D-RC (residuals)	0.91	60	7.24	0.85	0.022	120
ARNN-RC	–	30	–	0.50	0.03	–

series without explanatory variables. In this case, the term  $W^{\text{in}}u(t)$  is removed from Eqs. (1) and (4).

The following subsections describe variations of the S-RC paradigm, which can provide an advantage in certain contexts.

### 5.2. Ensemble reservoir computing

When the size of the training data is small, the performance of the reservoir depends strongly on its actual realization. The choice of matrices  $W^{\text{in}}$ ,  $W^{\text{back}}$ , and  $W$ , even when maintaining the spectral radius and sparsity, can affect the results of the model. Therefore, different runs of the same algorithm may give different result, unless the exact same matrices are used. In this circumstances, the robustness of the model can be increased by training multiple reservoirs with the same hyperparameters (see Table 2). The different predictions are then averaged to produce the final output. The operation of this setting is schematically depicted in Fig. 1(b). The final model, in this case, produces more stable results, and is less likely to overfit the training data.

### 5.3. Trend-seasonality decomposition reservoir computing

When the consecutive values of a time series are too similar, many predictive models tend to (wrongly) predict the previous value of the series. That is, if  $y(t) = y(t-1) + \epsilon$  with  $|\epsilon| \ll 1$ , the model might predict  $\hat{y}(t) = y(t-1)$ , and still producing a small error  $\epsilon$ . One way to prevent the model from mimicking the previous value is to decompose the series into *trend*, *seasonality* and *residuals*, and then predict each component separately, as shown schematically in Fig. 1(c). Notice that this is very relevant for the price series that are being studied in this work, since they have a strong seasonal periodicity; therefore this decomposition is useful to remove temporal patterns. In this case, an additive time series decomposition is suitable to decompose the time series. Indeed, the Augmented Dickey–Fuller test on the residuals gives a  $p$ -value of  $3 \cdot 10^{-11}$ ,  $2 \cdot 10^{-14}$  and  $3 \cdot 10^{-14}$ , for the zucchini, aubergine and tomato prices time series, respectively (see section Results above), this meaning that we can reject the hypothesis of the series needing further differentiating. Once the decomposition is done, an ensemble RC model is trained with each of the time series components. The final prediction is obtained as the sum of the trend, seasonality and residuals predictions.

### 5.4. Next generation reservoir computing

In Ref. [21], a novel RC framework was presented, which requires no random matrices and fewer hyperparameters to define the model. Its performance was tested in predictions of the evolution of two benchmark chaotic dynamical systems: the Lorentz attractor and the double-scroll electronic circuit. The operation of the model is schematically depicted in Fig. 1(d). Instead of using a random complex network as the reservoir, the  $k$  time-delay observations of the dynamical system and nonlinear functions of these observations are used to predict the output. That is, to predict a dynamical system  $(x(t), y(t), z(t))$ ,

one would use linear terms of the form  $x(t), \dots, x(t-k), y(t), \dots, y(t-k), z(t), \dots, z(t-k)$  and non-linear terms of the form  $x(t)x(t-k), \dots, x(t)x(t-k), x(t)y(t), \dots, x(t)y(t-k), \dots, x(t)z(t-k)$  to perform a linear regression to predict the future values  $(x(t+1), y(t+1), z(t+1))$ . Therefore, the prediction is just a linear function of these time-delay observations, and there is no need to define the random matrices  $W^{\text{in}}, W^{\text{back}}$ , and  $W$ . In our case, the time series is one-dimensional, so the terms of the  $k$ -delay vector are all of the form  $y(t)y(t-k)$ .

### 5.5. Auto-reservoir computing

Another RC framework was developed in Ref. [34], which allows to make multiple-step-ahead predictions of high-dimensional time series with small training sizes. This is the ARNN-RC algorithm, which uses the regressor variables as the reservoir by obtaining a mapping from the future high-dimensional regressor variables to their associated time series prediction. This mapping is based on a spatiotemporal information transformation [22]. The algorithm is schematically illustrated in Fig. 1(e).

### 5.6. Hyperparameters

Hyperparameters for the seven considered models are set by using Bayesian optimization, minimizing the MAE of the validation set. The optimal values obtained in this way for the RC models are reported in Table 2. In all cases  $t_{\text{dismiss}} = 10$ ,  $f = \tanh$  and  $f^{\text{out}} = \text{id}$  were used. Notice that the NG-RC model does not appear in the Table, since random matrices are not used in this method for the reservoir. In this case, optimal parameters were  $k = 3$  and  $\gamma = 0.56$ . Regarding the benchmark models, LSTM used 128 neurons and 5000 training epochs, and the best SARIMA model was a SARIMA(0, 0, 1)(1, 0, 1, 45).

### CRedit authorship contribution statement

**L. Domingo:** Developed the idea and the theory, Performed the calculations and analyzed the data, Discussions and interpretations of the results, Wrote the manuscript. **M. Grande:** Developed the idea and the theory, Discussions and interpretations of the results, Wrote the manuscript. **F. Borondo:** Developed the idea and the theory, Discussions and interpretations of the results, Wrote the manuscript. **J. Borondo:** Developed the idea and the theory, Discussions and interpretations of the results, Wrote the manuscript.

### Declaration of competing interest

The authors declare no competing financial or non-financial interests.

### Data availability

Data will be made available on request.

### Acknowledgments

The project that gave rise to these results received the support of a fellowship from “la Caixa” Foundation (ID 100010434). The fellowship

code is LCF/BQ/DR20/11790028. This work has also been partially supported by the Spanish Ministry of Science, Innovation and Universities, Gobierno de España, under Contract No. PID2021-122711NB-C21; and by DG of Research and Technological Innovation of the Community of Madrid (Spain) under Contract No. IND2022/TIC-23716.

### References

- [1] Jacobi J, et al. *Sci Rep* 2020;10:19145.
- [2] Chaudhary A, Gustafson D, Mathys A. *Nature Commun* 2018;9(1):848.
- [3] Herrero M, et al. *Nat Food* 2020;1:266–72.
- [4] Charlotte-Bunge A, Wood A, Halloran A, Gordon LJ. *Nat Food* 2022;3:933–41.
- [5] Dasgupta S, Robinson EJZ. *Sci Rep* 2022;12:4709.
- [6] European Commission. Agri-food markets dashboards. 2022, [https://ec.europa.eu/info/food-farming-fisheries/farming/facts-and-figures/markets/overviews/market-overview-sector\\_en](https://ec.europa.eu/info/food-farming-fisheries/farming/facts-and-figures/markets/overviews/market-overview-sector_en).
- [7] Martos-Pedrero A, Cortés-García FJ, Jiménez-Castillo D. *Sustainability* 2019;11:6390.
- [8] Ben-Ayed R, Hanana M, Ercisli S, Karunakaran R, Rebai A, Moreau F. *Plants* 2022;11:1230.
- [9] Xue L, Liu X, Lu S, Cheng G, Hu Y, Liu J, et al. *Nat Food* 2021;2(7):519–28.
- [10] English A, et al. The state of food and agriculture. moving forward on food loss and waste reduction. Rome: Food and Agriculture Organization of the United Nations; 2019.
- [11] Chen P, Niu A, Liu D, Jiang W, Ma B. *IOP Conf Ser Mater Sci Eng* 2018;394:052024.
- [12] Wang C-C, Chien C-H, Trappey AJC. *Processes* 2021;9(7):1157, URL <https://www.mdpi.com/2227-9717/9/7/1157>.
- [13] Jadhav V, Chinnappa-Reddy B, Gaddi G. *J Agric Sci Technol* 2017;19:981–92.
- [14] Mgaya JF. *Cogent Food Agric* 2019;5(1):1607430.
- [15] Wang C, Sun Z. *PLoS One* 2021;16:e0251436.
- [16] Guan Z, Zhao Y, Geng G. *Comput Econ* 2021.
- [17] Yin H, Jin D, Gu YH, Park CJ, Han SK, Yoo SJ. *Agriculture* 2020;10(12):612.
- [18] Luo C, Wei Q, Zhou L, Zhang J, Sun S. In: Li D, Liu Y, Chen Y, editors. *Computer and computing technologies in agriculture IV*. CCTA 2010. IFIP advances in information and communication technology, vol. 346. Springer, Berlin, Heidelberg; 2011.
- [19] Weng Y, Wang X, Hua J, Wang H, Kang M, Wang F-Y. *IEEE Trans Comput Soc Syst* 2019;6(3):547–53.
- [20] Jaeger H. German national research center for information technology GMD technical report, vol. 148. 2001.
- [21] Gauthier D, Bollt E, Griffith A, Barbosa W. *Nature Commun* 2021;12:5564.
- [22] Ma H-F, Zhou T, Aihara K, Chen L. *Int J Bifurcation Chaos* 2014;24:1430033.
- [23] Gao R, Du L, Duru O, Yuen KF. *Appl Soft Comput* 2021;102:107111.
- [24] Pathak J, Lu Z, Hunt B, Girvan M, Ott E. *Chaos* 2017;27:121102.
- [25] Pathak J, Hunt B, Girvan M, Lu Z, Ott E. *Phys Rev Lett* 2018;120:024102.
- [26] Rumelhart DE, Hinton GE, Williams RJ. *Parallel distributed processing: explorations in the microstructure of cognition, volume 1: foundations*. Cambridge, MA: MIT Press; 1986, p. 318–62.
- [27] Jaeger H. *Scholarpedia* 2007;2:2330.
- [28] Levine R, Zervos S. *World Bank Econ Rev* 1996;10(2):323–39.
- [29] Cowles III A. *Econometrica* 1933;309–24.
- [30] Lucchini L, Alessandretti L, Lepri B, Gallo A, Baronchelli A. *Sci Adv* 2020;6(51):eabd2204.
- [31] Wei WC. *Econom Lett* 2018;168:21–4, URL <https://www.sciencedirect.com/science/article/pii/S0165176518301320>.
- [32] Jaurigue L, Lüdge K. *Nature Commun* 2022;13:227.
- [33] Vandoorne K, Mechet P, Van Vaerenbergh T, et al. *Nature Commun* 2014;5:3541.
- [34] Chen P, Liu R, Aihara K, Chen L. *Nature Commun* 2020;11:4568.
- [35] Hochreiter S, Schmidhuber J. *Neural Comput* 1997;9:1735.