

# Forecasting Trends in Food Security: a Reservoir Computing Approach

Joschka Herteux<sup>a</sup>, Christoph R  th<sup>b</sup>, Amine Baha<sup>c</sup>, Giulia Martini<sup>a</sup>, and Duccio Piovani<sup>a</sup>

<sup>a</sup>World Food Programme, Research, Assessment and Monitoring Division (RAM), Via Cesare Giulio Viola 68, 00148 Rome, Italy

<sup>b</sup>Deutsches Zentrum f  r Luft- und Raumfahrt (DLR), Institute for AI Safety and Security, Wilhelm-Runge-Stra  e 10, 89081 Ulm, Germany

<sup>c</sup>World Food Programme Innovation Accelerator, Buttermelcherstrasse 16, 80469 Munich, Germany

## ABSTRACT

Early warning systems are an essential tool for effective humanitarian action. Advance warnings on impending disasters facilitate timely and targeted response which help save lives, livelihoods, and scarce financial resources. In this work we present a new quantitative methodology to forecast levels of food consumption for 60 consecutive days, at the sub-national level, in four countries: Mali, Nigeria, Syria, and Yemen. The methodology is built on publicly available data from the World Food Programme's integrated global hunger monitoring system which collects, processes, and displays daily updates on key food security metrics, conflict, weather events, and other drivers of food insecurity across 90 countries (<https://hungemap.wfp.org/>). In this study, we assessed the performance of various models including ARIMA, XGBoost, LSTMs, CNNs, and Reservoir Computing (RC), by comparing their Root Mean Squared Error (RMSE) metrics. This comprehensive analysis spanned classical statistical, machine learning, and deep learning approaches. Our findings highlight Reservoir Computing as a particularly well-suited model in the field of food security given both its notable resistance to over-fitting on limited data samples and its efficient training capabilities. The methodology we introduce establishes the groundwork for a global, data-driven early warning system designed to anticipate and detect food insecurity.

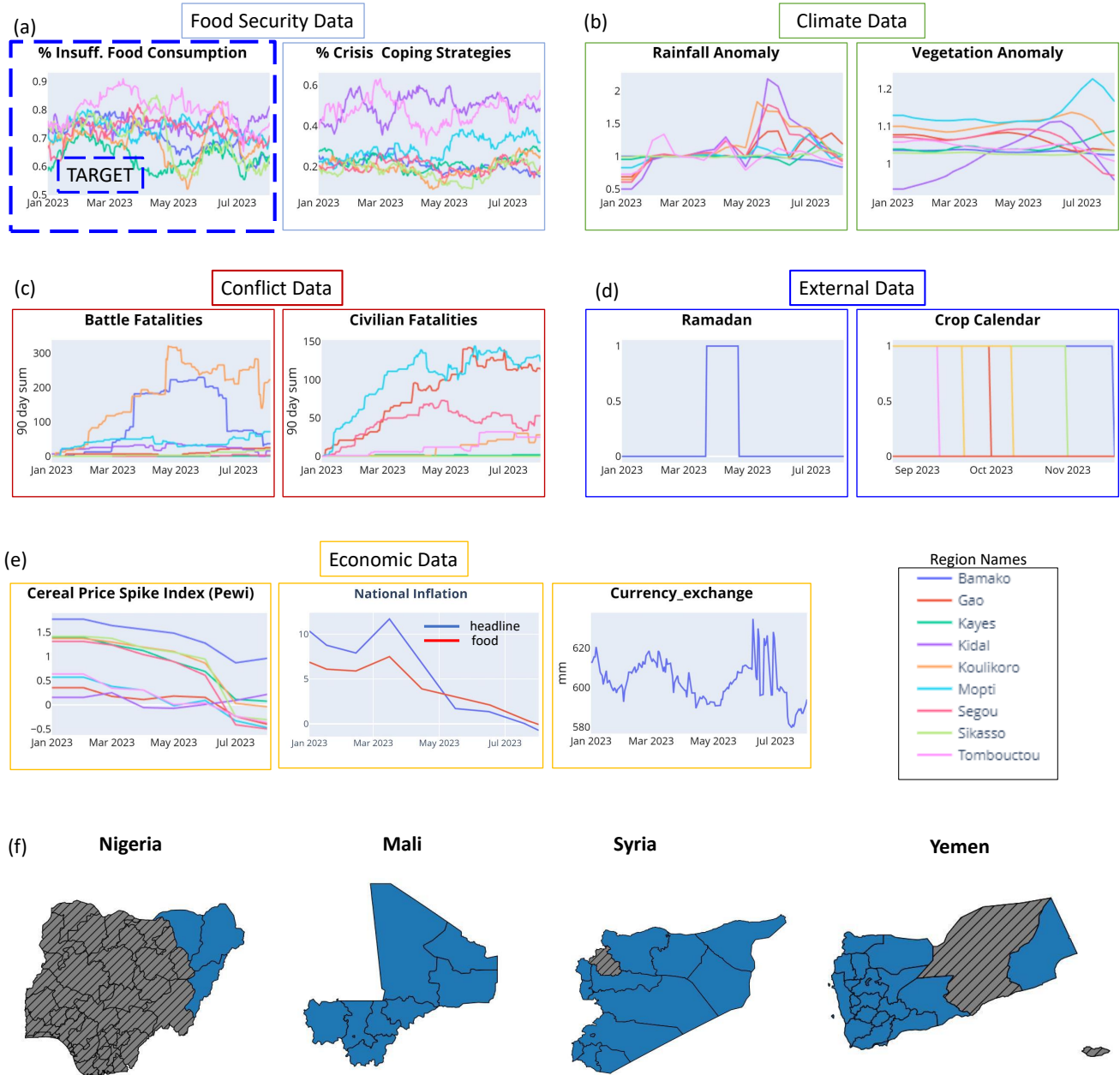
## Introduction

Conflict<sup>1,2</sup>, climate extremes, and soaring food, fertilizer and energy prices<sup>3</sup> on the heels of an incomplete recovery from the COVID-19 pandemic<sup>4</sup> have created a food crisis of unprecedented proportions<sup>5</sup>. The war in Ukraine further complicated the situation with millions of people a step away from famine. In such an uncertain environment it is incumbent upon humanitarian agencies such as the World Food Programme to deploy effective early warning systems that monitor food security conditions in the most vulnerable countries<sup>6</sup>. Early warning systems serve as the foundation for preparedness and quick response to potential food crises, whether man-made or natural disasters. They help humanitarian organizations better target assistance to where it is needed the most, hence minimizing duplication and waste. Forecasting systems are an integral part of early warning systems, allowing them to anticipate potential hazards and issue alerts in a timely and effective manner.

A robust approach to creating accurate forecasting tools is to leverage modern time series prediction methods that are based on Machine Learning (ML)<sup>7</sup>. These methods can be applied to a growing number of available data streams, and have proven to be successful across a range of diverse fields. Examples of these fields include monitoring epidemics<sup>8–13</sup>, predicting financial markets<sup>14–17</sup>, tracking energy consumption<sup>18</sup> and forecasting weather patterns and climate change impacts<sup>19,20</sup>. The scientific community has been studying and modeling food security for many decades, and recent research has shown that ML can also be used in this context<sup>21–23</sup>. By using ML, it is possible to develop complex, data-driven models with minimal feature engineering, thereby avoiding the need for an in-depth understanding of the underlying processes, provided that enough training data is available. This makes ML an attractive candidate for the food security context, where the problem is complex and reliant on local conditions, making it challenging to generalize any claims that may lead to a knowledge-based model.

For instance, the Food and Agriculture Organization of the United Nations (FAO) has investigated long-term forecasts at the country level for several years<sup>24,25</sup>, while nowcasting food security from secondary data has been attempted for households in Uganda<sup>26,27</sup> and for sub-national units in multiple at-risk countries<sup>22</sup> and comparable work has also been carried out in Malawi<sup>28</sup>. Studies carried out by the World Bank<sup>29,30</sup> and Westerveld et al.<sup>31</sup> have focused on forecasting transitions between different phases of food security, while a recent study from the French National Centre for Scientific Research (CNRS) has investigated use of ML for the prediction of food consumption<sup>32</sup>. Researchers from UCLA recently used remotely sensed soil moisture data and food prices to predict changes in food security<sup>33</sup>. Even the use of relevant text-based features from news

## Input Data Examples



**Figure 1.** (a)-(e): the figure shows the time series of the data used in constructing the forecasting methodology. The target variable, highlighted by the blue dashed curve in (a), is the regional prevalence of insufficient food consumption extracted from the Food Consumption Score (FCS). In addition to the historical values of the target, the methodology incorporates predictors coming from another food security indicator, climate, conflict, and economic data. External datasets with known future values, including crop calendars and Ramadan days, are also considered (e). While the figures are based on data from Mali, the framework remains consistent across all countries. A detailed breakdown of the data for each country can be found in Table 1, with comprehensive information on data sources available in the methods section. (f) The map displays the first administrative level boundaries in the four tested countries, where grey dashed polygons indicate regions that were excluded due to data unavailability.

streams has been successfully attempted to forecast food insecurity as defined by the Integrated Phase Classification (IPC) up to 12 months ahead<sup>23</sup>. Notably in<sup>34</sup> authors used a panel vector-auto regression to model and forecast IPC 1, 2 and 3+ phases in 15 countries, and then applied a LASSO to identify the most important drivers. Finally in<sup>21</sup> authors used XGBoost to build a 30-day forecaster on a similar data set up used in this work.

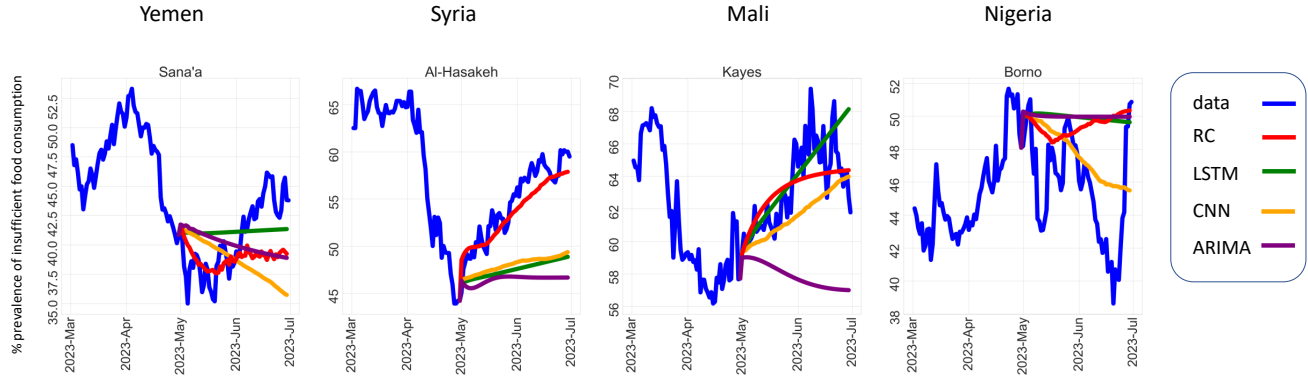
The World Food Programme regularly collects data on food security in countries where it operates<sup>35</sup>. The methodology we present in this manuscript builds upon data sourced from the Real-Time Monitoring (RTM) system, developed by the organization over the last decade and publicly available via HungerMap<sup>LIVE36</sup>. Collected through Computer Assisted Telephone Interviews (CATI)<sup>37</sup>, the data provides daily updates on food security indicators at the household level in near 40 countries. One such indicator is the Food Consumption Score (FCS), a composite measure that assesses the frequency of consumption of different food groups in the seven days prior to the survey, categorizing households as having poor, borderline, or acceptable food consumption. Households with insufficient food consumption refers to those with poor or borderline food consumption and is the target variable for our study. In contrast to other food security monitoring initiatives like IPC, CH, or FEWSNet, WFP's RTM program stands out due to several significant distinctions. Other programs release country updates at intervals of six months or more while the RTM program provides daily updates throughout the entire year. This unique feature enables real-time monitoring of the dynamic changes in food security levels, setting it apart as a more responsive and timely tool for assessing and addressing evolving challenges. Furthermore, the underlying process to generate the indicators in the RTM is purely quantitative, and does not rely on domain expert consensus or qualitative observations.

The main goal of this research is to develop a forecasting methodology capable of predicting insufficient food consumption, as presented in WFP's RTM system, over a span of 60 consecutive days. This has been attempted in four countries, Mali, Nigeria, Syria, and Yemen, at the sub-national level, using secondary data that represent the key-drivers of food insecurity: conflict, extreme weather events and economic shocks<sup>6</sup>. A successful forecasting methodology developed using the RTM data, building on the unique properties of the monitoring system, would therefore be complementary to the reported approaches, and prove valuable for quantitative timely insights at any given point in time. This includes anticipating levels of food insecurity following natural disasters, political or economic shocks, and while the lean season is unfolding when other information may be scarce. The versatility and flexibility of such methodology would ensure its utility all year around, and would not be limited to the predefined release dates of monitoring reports. Moreover, in other methodologies that extend their forecasts over longer periods, the longer duration is typically a result of the increased time granularity of the data. In such cases, a 12-month forecast corresponds to predicting only one or two data points into the future. Our methodology is designed to replicate the complete 60-day time series of insufficient food consumption, thereby estimating the curve's behavior throughout the entire time window and going beyond offering a single numerical output at the conclusion of the period. In order to increase the transparency and explainability of the model, an aspect that has been identified as crucial in the literature<sup>38–40</sup>, we further implement a simple method for dynamic feature selection. This allows us to assess the importance of different variables in the model and their impact on the prediction.

We compare a broad range of approaches, covering both classical statistical methods and state-of-the-art deep learning techniques. The model we identify to be the most suitable for the problem of forecasting trends of food insecurity is based on a Reservoir Computing (RC)<sup>41–43</sup>, a type of recurrent neural network (RNN), where only the last readout layer is trained. This approach, recently attracting considerable attention and application across various domains, offers a fast and efficient training procedure while mitigating challenges such as the vanishing gradient problem. Despite its simplicity RC has proven to be a valid alternative to advanced machine learning prediction techniques particularly in data-scarce contexts as can be found in<sup>44–46</sup> when applied to synthetic data. Specifically our methodology relies on an ensemble of RC models, exploiting the low resource requirements of training RC and stabilizing predictions. This was also recently attempted in<sup>47</sup> where RC is shown to outperform LSTMs on forecasting food prices. To assess the goodness of our approach, we conducted a comprehensive evaluation by comparing the Root Mean Squared Error (RMSE) of various methodologies: ARIMA, CNNs and LSTMs. Additionally, we referred to the results presented in<sup>21</sup>, where the authors developed an XGBoost forecaster. The findings presented in this manuscript highlight Reservoir Computing (RC) as a robust option for forecasting applications in the field of food security. Notably, RC outperformed the other tested algorithms, particularly in situations of more dramatic deterioration.

## Results

The target variable of our modelling is the time series of the sub-national prevalence of households with insufficient food consumption. An example is shown in the left panel of figure 1, together with time series examples of a sub sample of the secondary data. Our choice is inspired by the approach used in previous works<sup>21,22</sup> and makes use of available datasets describing conflict, economic shocks, and weather events. More specifically these consist of rainfall and vegetation levels, spikes in market food prices, food and headline inflation, currency exchange rates and number of fatalities due to different kinds of conflict. A detailed description of the input data is provided in the Methods section. Since Islam is the main religion in all four analysis countries, the holy month of Ramadan (fasting) is expected to significantly influence insufficient food

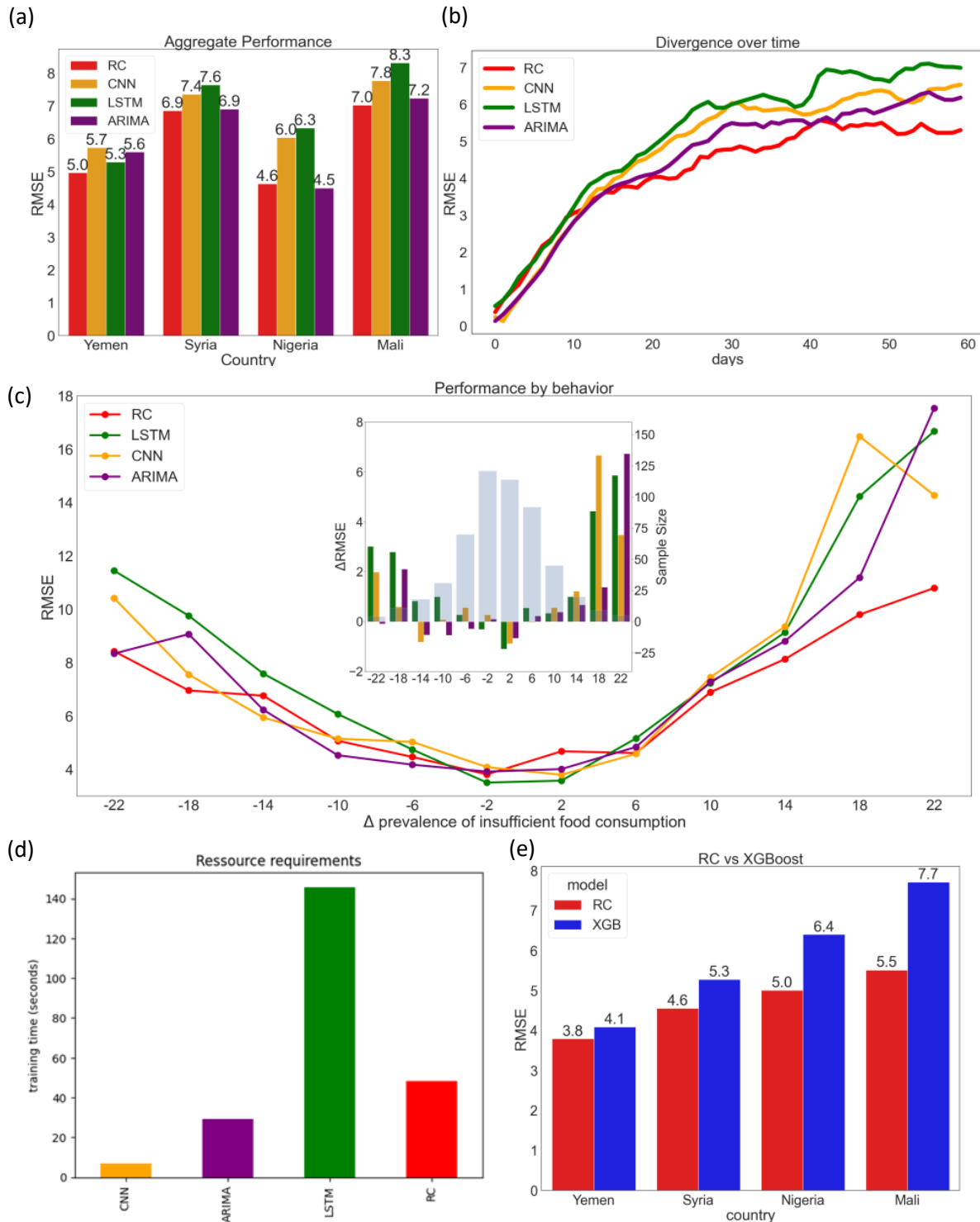


**Figure 2.** 60-day forecasts were generated using the RC, CNN, LSTM, and ARIMA models for four specific sub-national regions: Yemen, Syria, Mali, and Nigeria. In the visual representation, the blue curve represents the actual data, while each of the other curves depicts the prediction of one of the models. These predictions cover the time period starting on the first of May 2023, with each model trained on all available data up to that point.

consumption (we can see this in figure 5 in the Methods section). Therefore, the dates for Ramadan are used as an external input in the RC model, since it is available not only for the past, but also for the prediction period. To standardize all input variables to the same format of sub-national daily time series we performed some basic pre-processing steps. These include linear interpolation of variables which are only available on monthly basis, aggregating information to the desired sub-national level, averaging intermittent variables over a rolling window as well as de-noising and smoothing all data. Given the stochastic nature of the RC algorithm, to generate our output we repeat the prediction procedure 100 times with different seeds, with the median of all results then considered the final prediction.

A comparison between the predicted curves obtained with the different tested methodologies and the actual data is shown in the plots in figure 2 where the blue curves are the actual data representing the sub-national curves of insufficient food consumption, the colored curves representing the forecasts. The objective of this work is to test the goodness of the predictions for 60 consecutive days. To ensure a comprehensive comparison among all considered algorithms and to assess their robustness across all seasonal trends, we have evaluated their performances in 12 60-day splits. The forecasts for each split starter at the beginning of every month, covering the period from June 2022 to May 2023. To simulate the use of such methodology in a production environment we employed a walk-forward validation scheme, meaning the model was retrained for every new forecast adding the data of the previous month, therefore expanding the training moving forward in time. At this stage retroactive delays in data updates were not factored in and the models were trained on all available data at present (refer to Supplementary Material for additional details). For every split the hyper-parameters of the different algorithms were updated and selected through an extensive grid search of the performances on previous splits. Furthermore the algorithms were allowed to choose the subset of input variables that were found to be more informative, i.e. minimized the errors similarly to what was measured in<sup>22</sup>. This approach represents a first step towards adding to the forecasting methodology a method to explain the role of the input variables and their importance in generating the forecasts. In figure 4 of the Methods section the subsets of input variables are reported along with the frequency of their selection. It is clear that most of the predictive power comes from the auto-regressive element, the rCSI indicator and the seasonal external variables. A detailed description of the training procedure is found in the Methods section.

The aggregate errors are shown in figure 3 and are the result of ca. 70.000 CPU hours of calculations accumulated over several 10.000 runs for each country and model on the German Aerospace Agency's High Performance Cluster. This extensive computation period was essential to thoroughly assess the performances achieved with a vast array of hyper-parameter combinations across all tested splits and countries, ensuring a fair comparison between methodologies. It's noteworthy that the actual training times, on a local standard machine, for an individual hyper-parameter combination, illustrated in Figure 3(d), typically fall under 1 minute. As illustrated in the figure, our analysis of the performances includes a comprehensive study from multiple angles, emphasizing systematic differences in the outputs. To achieve this, we aggregated errors by country, by classifying the single splits of the underlying data according to their behavior, and explored the temporal evolution of errors. Testing the methodologies on different countries is a way to assess the algorithms' robustness to different sets of available input data sets as well as diverse underlying relationships with the target variable. A complete reference of the available input data sets per country is found in table 1. In figure 3a we can see how RC clearly outperforms the two deep learning algorithms



**Figure 3.** (a)-(c) Performances of the LSTM, CNN and ARIMA models measured in median RMSE, aggregated in three different ways. (a) country (b) prediction lag (c) behavior of the target time series. In (c) the inset plot shows a comparative analysis of each model's performance against the RC model based on the same metric, and the distribution of the samples over the different types of behaviour. (d) Average time cost for training each of the models used. The time was recorded during creation of the forecasts used in the main analysis. The RC time corresponds to the training of 100 separate models, as done to produce the forecasts and measure the performances (e) Comparison of the RC model's performance to the XGBoost model<sup>21</sup> on five 30-day time windows between October 2021 and February 2022 as shared by the authors.



LSTMs and CNNs in all countries, and is either better or comparable to ARIMA. Perhaps surprisingly the simple ARIMA algorithm outperforms the two more sophisticated LSTMs and CNNs. This type of results is in line with the findings in other studies<sup>47–51</sup> where ARIMA outperforms or yields comparable results to more advanced algorithms, and is likely to depend on the high amount of noise present in the data as well as its scarcity. The temporal evolution of the error is shown in Figure 3b: initially CNNs and ARIMA exhibit strong performance in the early time steps of the predictions, however over time RC distinctly outperforms all benchmarks, showcasing its increasing advantage as the prediction horizon extends. Additionally, Figure 3c illustrates the third and arguably the most crucial type of aggregation. The figure demonstrates a widening difference in errors among the models as the target variable exhibits more extreme variations, with the RC consistently outperforming all benchmarks by larger margins. The reason for the increase in errors across all models for larger variations in the data is attributed to the rarity of such behavior. The training dataset is indeed unbalanced and skewed toward stable curves as shown in the inset of figure 3c. Remarkably RC's performance, although in relative terms, experiences a substantial increase. This property is exceptionally valuable, and optimizing the methodology for these less common curve types will be the focus of future work. In figure 3d, we observe the training time for all tested algorithms on a single split, using the hyperparameters selected from the grid search. The comparison reveals that, despite employing an ensemble of 100 models for Reservoir Computing (RC), its training time remains comparable to ARIMA and is significantly more efficient than LSTMs. While the CNN achieves the shortest training time, this advantage is counterbalanced by a considerably inferior performance. The relatively long training times obtained using ARIMA depend on the necessity to train a distinct ARIMA model for each region in a country. In contrast, other methodologies enable simultaneous training, allowing for concurrent forecasting across all regions.

To further evaluate our proposed approach, we conducted a benchmark on the same dataset employed and shared in<sup>21</sup>. In their study, the authors implemented an XGBoost forecaster for a consecutive 30-day period, targeting the same variable as addressed in our manuscript. It's worth noting that XGBoost inherently lacks built-in support for multi-output regression, as its primary design focuses on single-output tasks such as regression and classification. Consequently, to achieve a 30-day prediction horizon, the authors were forced to train and evaluate thirty distinct models. In figure 3e we can clearly see how the RC outperformed the XGBoost in all tested countries. Given XGBoost's reliance on multiple models, rendering its use in a production environment challenging in terms of maintenance and updates, and the clear results obtained in comparing the performances on a horizon of 30 days, extending the approach to a 60-day prediction horizon was considered impractical and unlikely to provide substantial additional value.

## Discussion

This manuscript applies several purely quantitative time series prediction methodologies to the problem of forecasting insufficient food consumption, based on the data collected by the World Food Programme's real-time monitoring system. In direct comparison with LSTM, CNN and ARIMA we find Reservoir Computing to be the most suitable for this task in terms of performance, resource requirements, robustness and ease of use. We further proved this approach to surpass a previous attempt using XGBoost<sup>21</sup>. The Reservoir Computing algorithm has demonstrated to be a solid framework for the construction of a purely quantitative early warning system, owing to several advantageous features. Its ability to process multidimensional time series and learn complex nonlinear relations in target and features gives it the advantage over simple statistical models like ARIMA. The iterative prediction scheme allows the use of exogenous variables like seasonal and religious calendars where future values are already known as well as make implicit forecasts for those where this is not the case. The relative simplicity of the approach mitigates the danger of overfitting when compared to the more sophisticated Deep Learning methodologies CNN and LSTM, in line with what was shown in<sup>47</sup>. The limited resources needed for the training process allow an ensemble approach, the size of which can be finely tuned and adjusted based on the available computational resources and accuracy requirements of the field and context of application. Furthermore, we saw how the RC methodology was able to accommodate a degree of class imbalance in the training data. Indeed the above benchmark performance shown by the RC model does not depend on the majority of stable curves that form the training set, but is actually more evident in the subset of curves that represent dramatic increases in the levels of insufficient food consumption. These findings suggest the potential advantages of RC over models of this kind in data-scarce contexts, even when confronted with high-dimensional, noisy, and authentic datasets. The ensemble architecture appears to perform well without requiring specific modifications for large input dimensions, as seen in approaches like Local States<sup>52</sup>.

Future improvements of the forecasting methodology will concentrate on four main pillars: extending the forecasting horizon, tailoring the methodology to detect severe deterioration's of food security levels, developing methods to explain the role of the input variables and implementing the methodology in a production environment. Initial results obtained on 90-day forecasts confirm the RC algorithm as the preferred choice with respect to other benchmarks, despite a general increase of the errors (see Supplementary Material). Aggregating the underlying data to weekly or monthly time granularity, reducing the number of data points that need to be forecasted and smoothing uninformative fluctuations, seems like a promising possibility.

Increasing the target's time granularity would also eliminate the need for interpolation and other pre-processing steps thereby facilitating the deployment of the approach. Introducing a custom weighting system in the training process and duplicating data splits which show severe deterioration will be attempted to focus the methodology on those curves that are more important to anticipate. Also testing other and more modern architectures of Reservoir Computing like the Next Generation RC<sup>53</sup> or the Trend-Seasonality decomposition RC<sup>47</sup> variation might represent a valid choice to improve overall performances and extend the horizon. Finally, allowing the algorithms to autonomously select the most informative subset of features for each forecast, as done in this manuscript, marks an initial effort to comprehend the underlying drivers influencing the predictions. A comprehensive analysis of the results and the continued development of this approach will be crucial for enhancing the explanatory power of the methodology.

In conclusion, this research lays the groundwork for a dedicated quantitative tool for modeling and forecasting food insecurity. Our approach complements existing methodologies with its year-round versatility by offering real time updates on future levels of food insecurity. By leveraging the advantageous features of real-time monitoring, the methodology proves particularly valuable for foreseeing and quantifying the impact of events that occur between the releases of other monitoring systems. Our work presents the potential to furnish humanitarian organizations with the capability to rapidly institute new early warning systems to preemptively help vulnerable communities. The utilization of extant and expanding datasets in conjunction with the low requirement of computational power renders the proposed approach simple and economical to implement. As such, it could offer much-needed technical support to humanitarian actors in their efforts to anticipate, quantify and promptly address impending food crises.

## Methods

In this section, we provide an in-depth exploration of the forecasting methodologies employed in our study, along with a comprehensive overview of the input datasets utilized to generate the results discussed in the preceding sections. Particular emphasis is given to the selected Reservoir Computing framework, because of it being less utilised and its central role in this work. In addition, a detailed overview of the training procedures and the hyperparameter selection process for each methodology is shown.

### Algorithms

For this study we have investigated a number of forecasting methodologies ranging from the simple statistical model ARIMA (Autoregressive Integrated Moving Average) to the Deep Learning techniques CNN (Convolutional Neural Network) and LSTM (Long Short-Term Memory) where Reservoir Computing (RC) can be understood as a hybrid between the two extremes.

**Ensemble Reservoir Computing Model** Reservoir Computing is a simplified Recurrent Neural Network(RNN) algorithm known to be easy to train and to be well-suited for problems with limited amounts of data, where it outperforms other state-of-the-art prediction algorithms<sup>44–46</sup>. It is able to process multidimensional inputs as well as past values through its untrained hidden layer called the reservoir and use them to iteratively predict multidimensional time series. As an RNN it is well suited for temporal, sequential data. Because only the last readout layer is explicitly trained through a simple linear regression, it is still considered less complex than typical Deep Learning methodologies like LSTM. To utilize the low resource requirements of RC and stabilize any random fluctuations caused by the initialization of the untrained network weights we implemented a simple ensemble scheme for the model by training 100 separate models with the same hyperparameters similar to what has been applied in<sup>47</sup>. For each single model only the seed for the random creation of the reservoir network is changed. The final output of the ensemble model is derived by calculating the median of the compilation of single outputs.

The variant of RC we implement for each single model is based on the original version of RC known as Echo State Network (ESN)<sup>42</sup>. An ESN is a Recurrent Artificial Neural Network consisting of three layers: an input layer, a hidden reservoir layer and a read-out. While the input layer is simply responsible for distributing the incoming information in the network and the function of the readout is to shape the final output, the reservoir's purpose is firstly the projection of the data to a high-dimensional, nonlinear space and secondly to serve as the memory of the model. The dynamics of the ESN center around the reservoir state  $\mathbf{r}_t \in \mathbb{R}^N$ , which develops according to the update equation

$$\mathbf{r}_{t+1} = \tanh(\mathbf{A}\mathbf{r}_t + \mathbf{W}_{in}\mathbf{x}_t) \quad (1)$$

where the adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  represents the network,  $\mathbf{x}_t \in \mathbb{R}^{d_x}$  is the input fed into the reservoir and  $\mathbf{W}_{in} \in \mathbb{R}^{d_x \times N}$  is the input matrix. The hyperbolic tangent is a standard choice for the activation function. Since the state of the reservoir is a function of all the past inputs it received, it retains some information about the past if they are provided in sequential order. This is what makes ESN and RC in general well suited for time series analysis and prediction. We create  $\mathbf{A}$  as a sparse random network since low connectivity has been found to be advantageous<sup>54</sup>. The weights of the network are then drawn uniformly

from  $[-1, 1]$  and afterwards rescaled to fix the spectral radius  $\rho$  to some fixed value.  $\rho$  is a free hyper-parameter. We chose  $\mathbf{W}_{in}$  to be also sparse, in the sense that every row has only one nonzero element. This means every reservoir node is only connected to one degree of freedom of the input<sup>55</sup>. The nonzero elements are drawn uniformly from the interval  $[-1, 1]$  and then rescaled with a factor  $s_{input}$ , which is another free hyperparameter called input strength.

From this we can then compute the output  $\mathbf{y}_t \in \mathbb{R}^{d_y}$ . For our predictive model we train the ESN to approximate  $\mathbf{y}_t \approx \tilde{\mathbf{x}}_{t+1}$ . Where  $\tilde{\mathbf{x}}$  is not just the target data, but also all of the secondary data that is not known for the future. This way the model also learns to estimate the secondary variables and project them into the future. The readout is characterized by

$$\mathbf{y}_t = \mathbf{W}_{out} \tilde{\mathbf{r}}_t \quad (2)$$

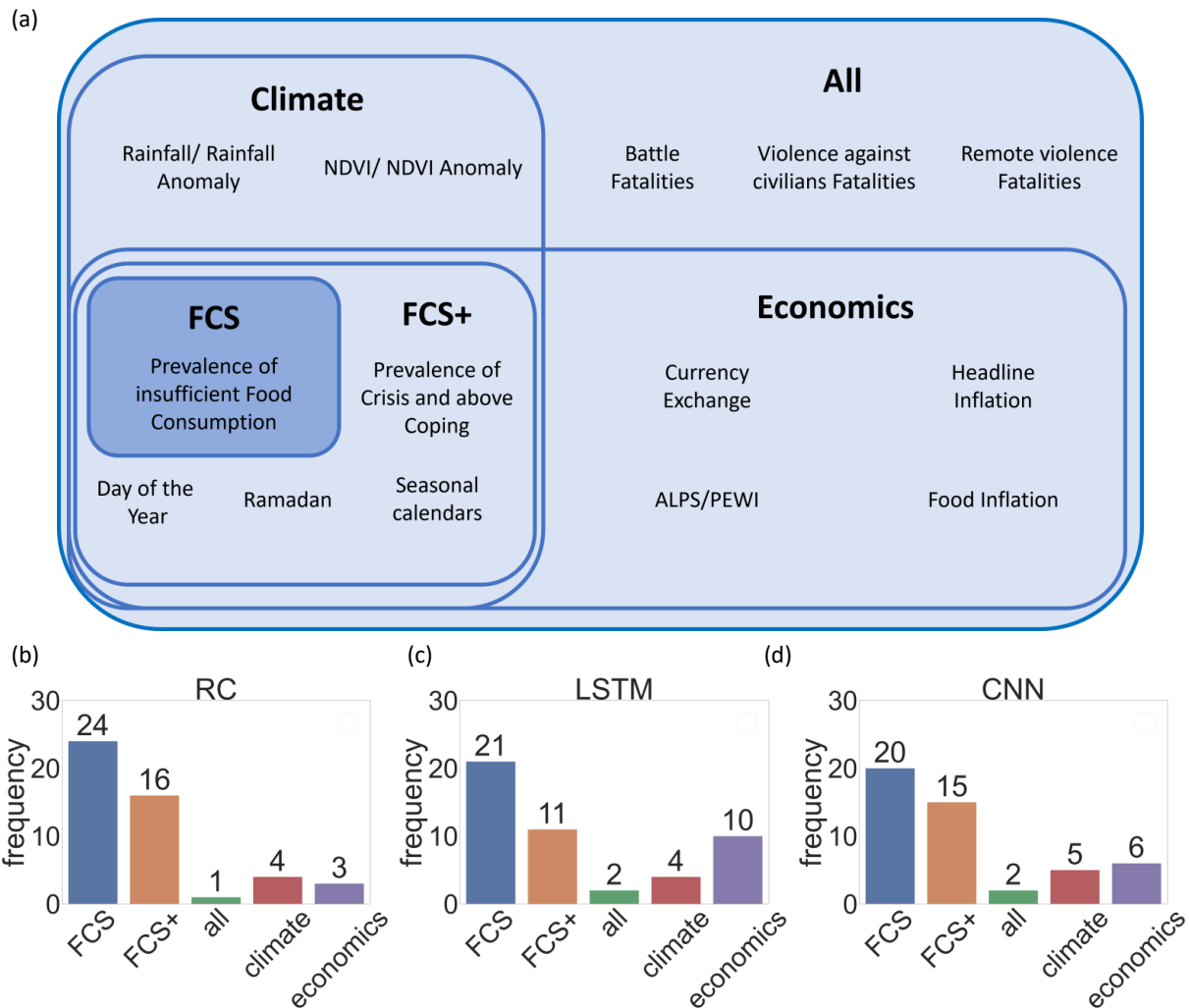
where  $\tilde{\mathbf{r}} = \{r_1, r_2, \dots, r_N, r_1^2, r_2^2, \dots, r_N^2\}$ . This is a commonly used<sup>56</sup> nonlinear transformation of  $\mathbf{r}_t$  which additionally serves to break the antisymmetry the equations would otherwise have<sup>57</sup>. The readout matrix  $\mathbf{W}_{out} \in \mathbb{R}^{d \times \tilde{N}}$  is the only part of the ESN that is trained. This is done via simple Ridge Regression<sup>58</sup> controlled by the regularization parameter  $\beta$ . The resulting model is as a first step capable of forecasting the target variable as well as the required secondary variables one day into the future. To do so the time series of past input data is fed into the reservoir sequentially. During this process the reservoir stores information about past and present values of these variables. The readout can then be used to map from the reservoir to the prediction. For the purpose of predicting time series of any length the model is used as a "closed loop". The output is fed back into the model allowing the sequence of input data to continue. This way the forecast can be extended to any length, although the error tends to grow with every iteration, thus effectively limiting the reasonable time window. Secondary variables have to be projected into the future in the same way for the procedure to work. Since this creates an additional source of error, we developed a way to include future values of the data in case they are known. For some features like Ramadan or growth seasonality, we do not train the model to forecast the values. Instead, the known future values are used as input during the respective prediction steps. An avenue for future research and an improved feature selection could be the inclusion of independent forecasts for secondary variables like rainfall and conflict, since the RC model is not optimized to predict their behavior.

**ARIMA:** The Autoregressive Integrated Moving Average (ARIMA) is a simple statistical model designed to predict future behavior of a time series solely based on its past values, without incorporating additional variables. In this model, each time step is presumed to have a linear dependence on the preceding  $p$  values of the time series, along with  $q$  error terms representing white noise. To address non-stationary time series, the data undergoes differencing  $d$  times. While these assumptions limit the model's ability to capture complex nonlinear relationships, ARIMA is recognized for its robustness, efficiency, and ease of training. Consequently, it remains widely applied in various domains such as financial time series<sup>50</sup> and epidemiology<sup>59</sup>.

**Convolutional Neural Networks (CNN):** a widely recognized category of Artificial Neural Network (ANN) particularly renowned for image recognition tasks. 1-dimensional CNNs have been successfully applied in time series prediction tasks even though they are not primarily designed for sequential data. These networks can incorporate multiple hidden layers and are able to learn temporal dependencies between the target data, its past values and secondary data, which they achieve via an advanced backpropagation-based training scheme. Our model is based on the keras implementation of CNN and consists of a variable number of 1D Convolution layers with ReLu activation functions each followed by a 1D Max Pooling layer. The number of filters, the kernel size and the pool size are among the free hyper-parameters of the architecture which are selected through the grid search. The readout is comprised of two Dense layers, the first employing ReLu activation functions and a variable number of units followed by a linear layer that maps to the output. In contrast to the iterative scheme utilized by RC, the model takes a variable number of past values of the employed features as input, and directly outputs a vector of predicted values. During training, the model parameters are trained by minimizing the Mean Squared Error (MSE) using the Adam optimizer for 200 epochs with a variable learning rate. Early Stopping was employed, using 20% of the training data for validation, to halt the training once the validation error fails to improve for five consecutive epochs.

**Long Short-Term Memory (LSTM):** known as the most widely adopted variant of Recurrent Neural Network, it is part of the larger category of ANNs where it stands out by its sophisticated mechanism to deal with long-range temporal dependencies through selectively retaining past information with the help of its forget gate<sup>60</sup>. The parameters of this model are trained via backpropagation through time (BPTT). While this is the most sophisticated algorithm we tested in this study, it is also the hardest to train and to fine-tune and carries the highest risk of overfitting. Similarly to our approach with CNNs we built our LSTM model using keras. The architecture consists of two LSTM layers connected by a Repeat Vector Layer. The activation functions used are of the type ReLU while the number of units in both layers is a free hyperparameter subject to tuning in the grid search. The same is true for the dropout rate which we apply to both layers on the input as well as the recurrent connections. The readout of the model consists of a Time Distributed Dense layer mapping to a vector of predicted values. We only apply the model to differenced data as opposed to making this a part of the hyperparameter tuning as in the RC and CNN model due to the fact that we saw a complete inability to learn the original data without differencing with the LSTM in preliminary trials. The training of the LSTM model mirrors that of the CNN model using an Adam optimizer and Early Stopping.





**Figure 4.** Feature groupings that were considered in the grid search for the RC, CNN and LSTM models to implement a dynamic feature selection. (a) Schema of how the features were classified into 5 different feature groupings. (b-d) frequency with which each feature grouping was selected for in (a) the RC model, (b) the LSTM model and (c) the CNN model.

## Data

As mentioned in the Introduction and Results sections, the models were trained on data representing the prevalence of food consumption and coping strategies as well as the global key-drivers of food insecurity. The following paragraphs introduce each of the input data sources and the features extracted from them. In table 1 we can see the availability of the data for each country.

**Prevalence of people with insufficient food consumption:** This measure is built on the Food Consumption Score (FCS), a commonly used food security indicator, and is obtained estimating the percentage of households in a region with an FCS below a country-dependent threshold indicating poor or borderline food consumption. This way a single value between 0 and 100 per region is gained. The Food Consumption Score is collected by WFP both by face to face surveys, which happen approximately once a year and through CATI (Computer Assisted Telephone Interviews) on a daily basis. The score evaluates the amount as well as the nutritional value of a household's food consumption. For some regions not enough data is available and they were thus excluded from the study. This is the target variable of our modelling.

**Prevalence of people using crisis or above crisis food based coping:** Similar to insufficient food consumption, the prevalence of people using crisis or above crisis food based coping is based on a food security indicator, the reduced Coping Strategy Index (rCSI), which represents a different dimension of the concept. The rCSI is calculated from the results of the same surveys as the FCS and therefore the same methodology is used. Again a prevalence or percentage is defined by counting the number of people in a region scoring below a certain threshold. The rCSI is in particular a measure of the coping strategies adopted by the

Variable	Yemen	Syria	Mali	Nigeria	frequency	spatial resolution
insufficient food consumption	✓	✓	✓	✓	daily	sub-national
food based coping	✓	✓	✓	✓	daily	sub-national
rainfall	✓	✓	✓	✓	dekad	sub-national
rainfall 1 month anomaly	✓	✓	✓	✓	dekad	sub-national
rainfall 3 months anomaly	✓	✓	✓	✓	dekad	sub-national
NDVI	✓	✓	✓	✓	dekad	sub-national
NDVI 1 month anomaly	✓	✓	✓	✓	dekad	sub-national
ALPS/PEWI	✓	✓	✓	✓	monthly	sub-national
food inflation			✓	✓	monthly	national
headline inflation			✓	✓	monthly	national
currency exchange official	✓	✓	✓	✓	daily	national
currency exchange unofficial	✓				daily	national
battle fatalities	✓	✓	✓	✓	daily	sub-national
violence against civilians fatalities	✓	✓	✓	✓	daily	sub-national
remote violence/explosions fatalities	✓	✓	✓	✓	daily	sub-national
Seasonal calendars	✓	✓	✓	✓	dekad	sub-national
Ramadan	✓	✓	✓	✓	daily	national
day of the year	✓	✓	✓	✓	daily	national
Number of input variables	266	160	114	360		

**Table 1.** Variables used in the model for each of the four pilot countries. Time resolution of the data is indicated under frequency. The level of spatial aggregation is shown under spatial resolution. The last row shows the total number of input variables used in each model, which is summed over all sub-national time series available for the specified features.

victims of food insecurity. Examples include limiting the sizes of food portions or the amount of meals per day, borrowing food or relying on help from friends or relatives, and choosing less preferred or less expensive food. Since both indicators stem from the same source, the availability of the data is the same.

**Rainfall and Vegetation:** Five rainfall and vegetation features were constructed, sourced from WFP’s Seasonal Explorer<sup>61</sup> and are originally provided by CHIRPS<sup>62</sup> (rainfall) and MODIS<sup>63</sup> (vegetation) satellite imagery. These include the amount of rainfall in mm, the 1- and 3-month anomaly of rainfall compared to historical seasonal behavior. Similarly, the NDVI value and its 1-month anomaly are used. The NDVI (Normalized Difference Vegetation Index) is a measure for the amount of vegetation in a region. From these WFP derives regional seasonal calendars which mark agriculturally relevant seasons. They can be considered a binary time series of either 0 or 1. All of these variables are given in 10-day intervals ("dekads") for every region.

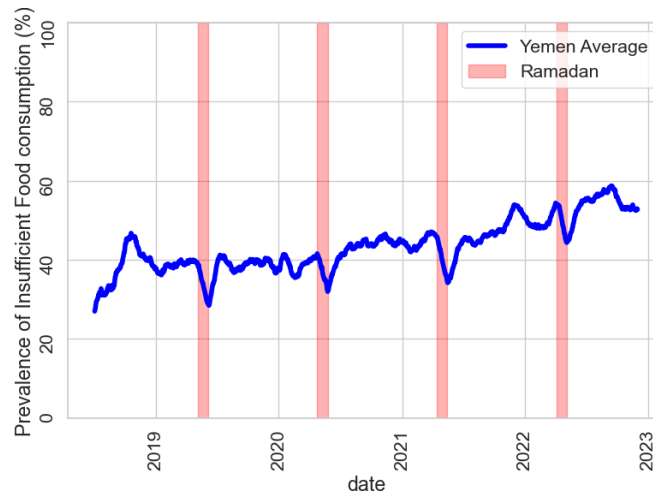
**Conflict:** To represent conflict as a driver of food insecurity we use data from the Armed Conflict Location & Event Data Project (ACLED)<sup>64</sup> which is a publicly available repository providing real-time and historical data on political violence and protest events in nearly 100 countries<sup>65</sup>. We build our features from the fatalities reported for events in an administrative region of the categories Battle, Violence against Civilians and Explosions/Remote Violence by summing them over the last 90 days in order to create daily time series.

**PEWI Index:** Food prices are an important factor for food security as they affect how much food a household can purchase. The forecasting power of this kind of data has been shown empirically<sup>66</sup>. The ALPS indicator (Alert for Price Spikes)<sup>67</sup> measures relative spikes in monthly prices by comparing them to estimated values and is calculated monthly as follows

$$ALPS = (Price_t - \hat{Price}_t) / \sigma_\epsilon \quad (3)$$

where  $t$  gives the month and  $\sigma_\epsilon$  describes the historic standard deviation of the residuals ( $Price_t - \hat{Price}_t$  for previous months). WFP monitors commodity prices in local markets monthly and makes them publicly available through its Economic Explorer<sup>68</sup>. For our model, we only use commodities in the "cereals and tubers" category. The data is averaged over all commodities within this category and across all markets in a given region. This aggregates the data to single time series per first-level administrative unit. The ALPS indicator is sometimes also referred to as PEWI (Price Early Warning Indicator).

**Macro-Economics:** To gain a better understanding of the overall situation in a country we also look at several economic indicators, which are available at the country level. We include food and headline inflation to complement our monthly data on food commodity prices. We also use daily effective exchange rates for local currency. All of these indicators can be found in WFP’s Economic Explorer<sup>68</sup>.



**Figure 5.** Prevalence of people with insufficient food consumption averaged over all available regions of Yemen in blue. Time periods of Ramadan shown by areas shaded red.

**Ramadan:** Ramadan is the month of fasting in the Islamic faith, which is the predominant religion in all four countries considered in this study. The traditional practice of fasting for the whole month can clearly be expected to have a strong influence on food consumption and in the way it is self-reported. The exact relationship is likely complex, but the impact can empirically be seen in the data. In figure 5 we can see how the target data averaged over sub-national units is affected. The target time series consistently starts to go down during the time of Ramadan and recovers to roughly the same level afterwards in a similar time frame. These dips are among the most noticeable features of the time series. To accommodate this, we include the occurrence of Ramadan as a binary variable. If a date lies during Ramadan, this feature has a value of 1, otherwise it is 0. Since we know even for future dates if they occur in Ramadan, this feature is processed separately as external data. It is available during prediction and does not need to be inferred.

**Day of the year:** To give the models the ability to learn seasonal effects as well as possible we include a feature to represent the day of the year as a number ranging from 1 to 365. Similarly to Ramadan this feature is of course known for the future and can be used as an external variable.

## Training Procedure

To ensure a fair comparison of all models and determine their optimal configurations, we conducted an extensive grid search on a high-performance cluster for each model and country. In order to simulate real-life scenarios, we implemented a walk-forward validation scheme across the 12 splits under consideration covering a full year. As a preprocessing step all input data was resampled to a daily resolution and to eliminate isolated missing values linear interpolation was used. A trailing 10-day moving average was applied to smooth the data and reduce the amount of noise. For every prediction, we record the RMSE and the utilized hyperparameters. The result is a rich dataset that enables an in-depth analysis of our models' behavior and the significance of various hyperparameters.

Results directly derived from the grid search are shown in figure 6. The importance of the spectral radius and the input strength in RC is confirmed in agreement with the literature. For CNN we can see that differencing improves the the performance significantly, while other hyperparameters like e.g. the number of filters only have a small impact. For LSTM we identify the learning rate, the number of units, the dropout rate and the length of lookback as having a significant impact. We can see that it is generally impossible to define a single hyperparameter as good or bad since they are usually interdependent. Nevertheless, the fact that high dropout rates are favored in some cases in LSTM indicates a tendency towards overfitting. Overall, it is observed that RC stands out as the model for which it is easiest to identify generally reasonable hyperparameter configurations that yield satisfactory performance across all splits. This implies that satisfactory forecasts could even be made with less frequent retuning of the model's hyperparameters.

Within this framework we implement a walk-forward optimization procedure. This technique is commonly used in time series analysis and forecasting to assess the performance of a predictive model. The idea is to train and test the model sequentially over time, moving forward in a step-by-step manner. This process allows the model to be evaluated on new data that becomes available as time progresses. The process can be separated into the following steps, which are carried out for each model on each country:

Parameter	range
$\rho$	0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1
$\beta$	$10^{-5}$ , $10^{-3}$ , $10^{-1}$ , 10, 100
$s_{input}$	0.1, 0.3, 0.5, 1.0, 1.5
features	FCS, FCS+, economics, climate, all
differencing	True, False

**Table 2.** Ranges of RC hyperparameters to be searched during tuning procedure.

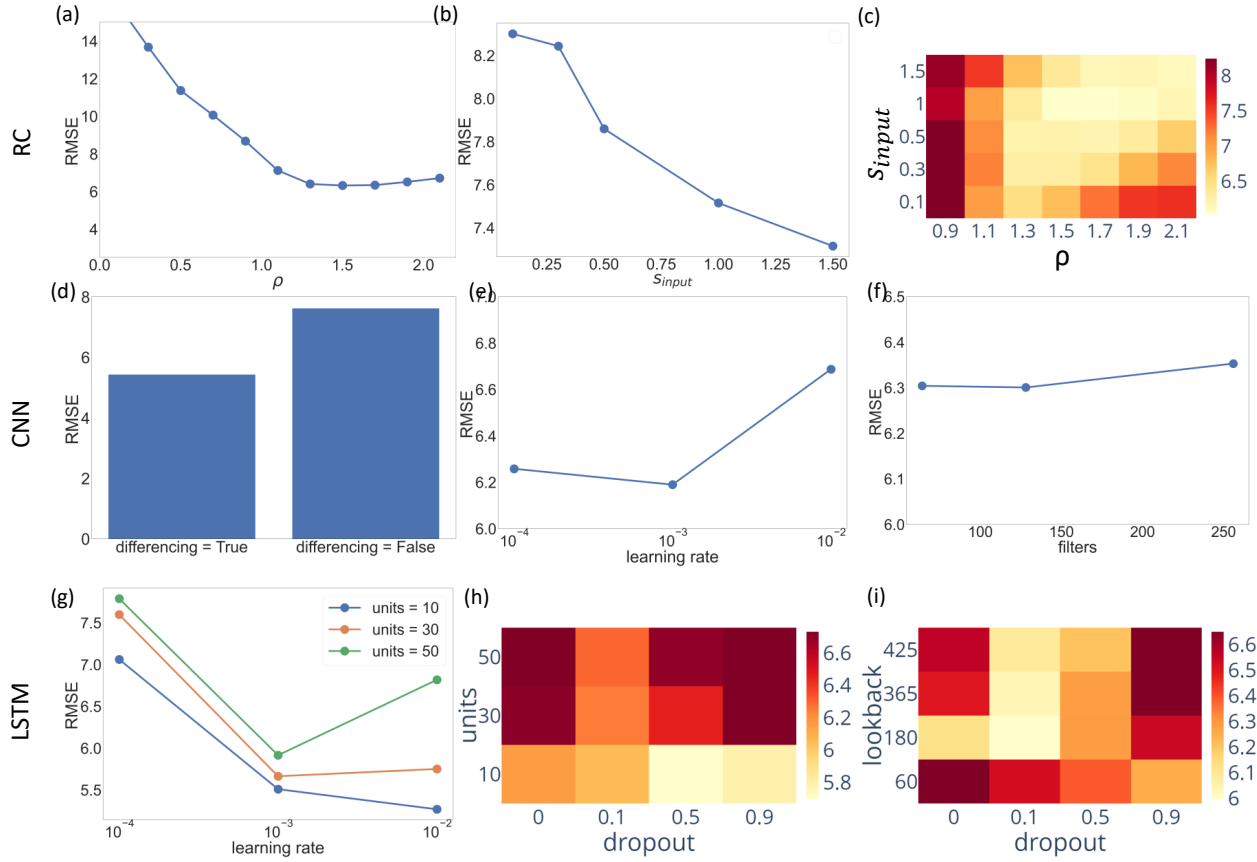
Parameter	range
learning rate	0.0001, 0.001, 0.01
lookback	60, 180, 365, 425
units	10, 30, 50
dropout rate	0., 0.1, 0.5, 0.9
features	FCS, FCS+, economics, climate, all

**Table 3.** Ranges of LSTM hyperparameters to be searched during tuning procedure.

1. **Selecting a Split:** A point in the the time series is chosen as the cutoff between training and evaluation. The data before this point is considered to be the training data. The data after this point is considered test data.
2. **Training the Model:** The model is trained on the training data.
3. **Evaluating the Model:** A forecast is created with the trained model on the time period after the cutoff and compared to the test data where the RMSE is calculated between forecast and test data.
4. **Walk Forward:** A new split with a new cutoff point further ahead in time is selected and the process is repeated.

This strategy can be used to evaluate the model with the best hyperparameters in the following way: the process is carried out for each combination of hyperparameters considered. To evaluate the performance and simulate an real-life application with regular retraining of the model hyperparameters for each split are chosen based on their aggregate performance across all preceding splits, where the prediction period ended before the beginning of the current one. This prevents any mixing of training and test set. The metric used to evaluate this aggregate performance is the median RMSE over these splits. The choice of the median RMSE over the mean RMSE was motivated by the fact that we found usage of the latter to be more likely to lead to mean-reverting, flat predictions. Conversely, using only a smaller amount of previous splits to select the hyperparameters proved to be too unstable given the heterogenous and noisy nature of the data where the best performance in a single split can often be an outlier. With the optimal knowable configuration for a given split we can properly evaluate by looking at the performance of the respective version of the model on the respective time period. Using the differences or returns of the prevalence of insufficient food consumption as target instead of the raw values proved to be particularly important for the LSTM and CNN models which were otherwise mostly unable to successfully forecast. For this reason we added this as an option for RC as well while for ARIMA differencing is a already a part of the regular setup.

Beyond the configuration of the algorithms themselves we also investigated a noteworthy modification of the input data. Because the importance of each feature as well as their information content and noise-levels can vary drastically from country to country and even between splits, we implemented a method to select the most useful features for each prediction in a way that could be used in application. In order to achieve this we introduced a hyperparameter for different groups of features in the grid search which makes it possible to optimize it together with the model configuration. The five groups we define are called "FCS", "FCS+", "climate", "economics" and all ranging from including only the target data in "FCS" to the full set of features in "all". Exactly which features are part of the different groups is visualized in figure 4a. In figure 4b-d we present the frequency with which each of the groupings is selected by the three algorithms RC, LSTM and CNN. The smaller groups of "FCS" and "FCS+" are selected with the highest frequency indicating that the autoregressive component is the most generally predictive one. However, the larger groups especially "climate" and "economics" do make their appearance showing that they are indeed informative at specific instances. This method was not applied for ARIMA, which does not use secondary data. For all models a large preliminary grid search was carried out on the example of Yemen to identify the most relevant hyperparameters. Based on the results the ranges and fixed parameters were selected as shown in tables 2,3,4 and 5. While the hyperparameters and their ranges generally differ for each model, our selection focused on those with the highest impact on performances while keeping the overall resource requirements of the grid search similar for RC, CNN and LSTM.



**Figure 6.** Additional results on the influence of different hyperparameters derived from the grid search on Yemen. The value shown is always the median RMSE over all splits and unspecified hyperparameters. **(a-b):** RC results for spectral radius and input strength, which are often considered the most important hyperparameters in RC. Results are separated into runs with and without differencing to showcase varying behavior. **(d-f):** CNN results on differencing, learning rate and number of filters. **(g-i):** LSTM results on learning rate, number of units, dropout rate and length of lookback.

Parameter	range
learning rate	0.0001,0.001,0.01
lookback	60, 180, 365, 425
kernel size	2, 5, 10, 15
kernel size	64, 128, 256
pool size	2, 4, 6
layers	1, 2
features	FCS, FCS+, economics, climate, all
differencing	True, False

**Table 4.** Ranges of CNN hyperparameters to be searched during tuning procedure.

Parameter	range
d	0, 1
p	1, 2, 3, 4
q	1, 3, 5, 7, 9

**Table 5.** Ranges of ARIMA hyperparameters to be searched during hyperparameter tuning.



## Data Availability

The data that support the findings of this study are available from the corresponding author, D.P.

## Code Availability

The code that supports the findings of this study is available from the corresponding author, D.P.

## References

1. Jagtap, S. *et al.* The russia-ukraine conflict: Its implications for the global food supply chains. *Foods* **11**, DOI: [10.3390/foods11142098](https://doi.org/10.3390/foods11142098) (2022).
2. Ben Hassen, T. & El Bilali, H. Impacts of the russia-ukraine war on global food security: Towards more sustainable and resilient food systems? *Foods* **11**, DOI: [10.3390/foods11152301](https://doi.org/10.3390/foods11152301) (2022).
3. Alexander, P. *et al.* High energy and fertilizer prices are more damaging than food export curtailment from ukraine and russia for food prices, health and the environment. *Nat. Food* 1–12 (2022).
4. Picchioni, F., Goulao, L. F. & Roberfroid, D. The impact of covid-19 on diet quality, food security and nutrition in low and middle income countries: A systematic review of the evidence. *Clin. Nutr.* **41**, 2955–2964 (2022).
5. WFP. War in ukraine drives global food crisis (2022). <https://www.wfp.org/publications/war-ukraine-drives-global-food-crisis>.
6. WFP. Global report on food crises - 2022 (2022). <https://www.wfp.org/publications/global-report-food-crises-2022>.
7. Lim, B. & Zohren, S. Time-series forecasting with deep learning: a survey. *Philos. Transactions Royal Soc. A* **379**, 20200209 (2021).
8. Polyvianna, Y., Chumachenko, D. & Chumachenko, T. Computer aided system of time series analysis methods for forecasting the epidemics outbreaks. In *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, 1–4 (IEEE, 2019).
9. Kraemer, M. U. *et al.* Past and future spread of the arbovirus vectors aedes aegypti and aedes albopictus. *Nat. microbiology* **4**, 854–863 (2019).
10. Tizzoni, M. *et al.* Real-time numerical forecast of global epidemic spreading: case study of 2009 a/h1n1pdm. *BMC medicine* **10**, 1–31 (2012).
11. Scarpino, S. V. & Petri, G. On the predictability of infectious disease outbreaks. *Nat. communications* **10**, 1–8 (2019).
12. Ardabili, S. F. *et al.* Covid-19 outbreak prediction with machine learning. *Algorithms* **13**, 249 (2020).
13. Prakash, K. B., Imambi, S. S., Ismail, M., Kumar, T. P. & Pawan, Y. Analysis, prediction and evaluation of covid-19 datasets using machine learning algorithms. *Int. J.* **8**, 2199–2204 (2020).
14. Timmermann, A. Forecasting methods in finance. *Annu. Rev. Financial Econ.* **10**, 449–479 (2018).
15. Elliott, G. & Timmermann, A. Forecasting in economics and finance. *Annu. Rev. Econ.* **8**, 81–110 (2016).
16. Sezer, O. B., Gudelek, M. U. & Ozbayoglu, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. soft computing* **90**, 106181 (2020).
17. Sirignano, J. & Cont, R. Universal features of price formation in financial markets: perspectives from deep learning. *Quant. Finance* **19**, 1449–1459 (2019).
18. Deb, C., Zhang, F., Yang, J., Lee, S. E. & Shah, K. W. A review on time series forecasting techniques for building energy consumption. *Renew. Sustain. Energy Rev.* **74**, 902–924 (2017).
19. Schultz, M. G. *et al.* Can deep learning beat numerical weather prediction? *Philos. Transactions Royal Soc. A* **379**, 20200097 (2021).
20. Mudelsee, M. Trend analysis of climate time series: A review of methods. *Earth-science reviews* **190**, 310–322 (2019).
21. Foini, P., Tizzoni, M., Martini, G., Paolotti, D. & Omodei, E. On the forecastability of food insecurity. *Sci. Reports* **13**, 2793 (2023).
22. Martini, G. *et al.* Machine learning can guide food security efforts when primary data are not available. *Nat. Food* **3**, 716–728 (2022).

23. Balashankar, A., Subramanian, L. & Fraiberger, S. P. Predicting food crises using news streams. *Sci. Adv.* **9**, eabm3449 (2023).
24. Organization, W. H. *et al.* *The state of food security and nutrition in the world 2020: transforming food systems for affordable healthy diets*, vol. 2020 (Food & Agriculture Org., 2020).
25. Wanner, N., Cafiero, C., Troubat, N. & Conforti, P. Refinements to the fao methodology for estimating the prevalence of undernourishment indicator. *Documento de trabajo* **2014**, 14–05 (2014).
26. Mwebaze, E., Okori, W. & Quinn, J. A. Causal structure learning for famine prediction. In *2010 AAAI Spring Symposium Series* (2010).
27. Okori, W. & Obua, J. Machine learning classification technique for famine prediction. In *Proceedings of the world congress on engineering*, vol. 2, 4–9 (2011).
28. Lentz, E. C., Michelson, H., Baylis, K. & Zhou, Y. A data-driven approach improves food insecurity crisis prediction. *World Dev.* **122**, 399–409 (2019).
29. Andree, B. P. J., Chamorro, A., Kraay, A., Spencer, P. & Wang, D. Predicting food crises. *The World Bank* (2020).
30. Wang, D., Andree, B. P. J., Chamorro, A. F. & Girouard Spencer, P. Stochastic modeling of food insecurity. *The World Bank* (2020).
31. Westerveld, J. J. *et al.* Forecasting transitions in the state of food security with machine learning using transferable features. *Sci. Total. Environ.* **786**, 147366 (2021).
32. Deléglise, H. *et al.* Food security prediction from heterogeneous data combining machine and deep learning methods. *Expert. Syst. with Appl.* **190**, 116189 (2022).
33. Krishnamurthy R, P. K., Fisher, J. B., Choularton, R. J. & Kareiva, P. M. Anticipating drought-related food security changes. *Nat. Sustain.* **5**, 956–964 (2022).
34. Wang, D., André, B. P. J., Chamorro, A. F. & Spencer, P. G. Transitions into and out of food insecurity: A probabilistic approach with panel data evidence from 15 countries. *World Dev.* **159**, 106035 (2022).
35. WFP. Food security analysis (2022). <https://www.wfp.org/food-security-analysis>.
36. WFP. Hungermap live (2022). <https://hungermap.wfp.org>.
37. WFP. The world food programme's real-time monitoring systems: Approaches and methodologies (2021). <https://docs.wfp.org/api/documents/WFP-0000135070/download/>.
38. McBride, L. *et al.* Predicting poverty and malnutrition for targeting, mapping, monitoring, and early warning. *Appl. Econ. Perspectives Policy* **44**, 879–892 (2022).
39. Baylis, K., Heckeley, T. & Storm, H. Machine learning in agricultural economics. In *Handbook of Agricultural Economics*, vol. 5, 4551–4612 (Elsevier, 2021).
40. Zhou, Y., Lentz, E., Michelson, H., Kim, C. & Baylis, K. Machine learning for food security: Principles for transparency and usability. *Appl. Econ. Perspectives Policy* **44**, 893–910 (2022).
41. Maass, W., Natschlaeger, T. & Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **14**, 2531–2560, DOI: [10.1162/089976602760407955](https://doi.org/10.1162/089976602760407955) (2002).
42. Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Ger. Ger. Natl. Res. Cent. for Inf. Technol. GMD Tech. Rep.* **148**, 13 (2001).
43. Lukoševičius, M. & Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149 (2009).
44. Chattopadhyay, A., Hassanzadeh, P. & Subramanian, D. Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Process. Geophys.* **27**, 373–389, DOI: [10.5194/npg-27-373-2020](https://doi.org/10.5194/npg-27-373-2020) (2020).
45. Vlachas, P. R. *et al.* Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics (2019). [1910.05266](https://arxiv.org/abs/1910.05266).
46. Shahi, S., Fenton, F. H. & Cherry, E. M. Prediction of chaotic time series using recurrent neural networks and reservoir computing techniques: A comparative study. *Mach. learning with applications* **8**, 100300 (2022).
47. Domingo, L., Grande, M., Borondo, F. & Borondo, J. Anticipating food price crises by reservoir computing. *Available at SSRN 4457897* (2023).

48. Han, J., Lin, H. & Qin, Z. Prediction and comparison of in-vehicle co2 concentration based on arima and lstm models. *Appl. Sci.* **13**, 10858 (2023).
49. Menculini, L. *et al.* Comparing prophet and deep learning to arima in forecasting wholesale food prices. *Forecasting* **3**, 644–662 (2021).
50. Kobiela, D., Krefta, D., Król, W. & Weichbroth, P. Arima vs lstm on nasdaq stock exchange data. *Procedia Comput. Sci.* **207**, 3836–3845 (2022).
51. Frausto-Solís, J., Hernández-González, L. J., González-Barbosa, J. J., Sánchez-Hernández, J. P. & Román-Rangel, E. Convolutional neural network–component transformation (cnn–ct) for confirmed covid-19 cases. *Math. Comput. Appl.* **26**, 29 (2021).
52. Baur, S. & R  th, C. Predicting high-dimensional heterogeneous time series employing generalized local states. *Phys. Rev. Res.* **3**, 023215, DOI: [10.1103/PhysRevResearch.3.023215](https://doi.org/10.1103/PhysRevResearch.3.023215) (2021).
53. Gauthier, D. J., Bollt, E., Griffith, A. & Barbosa, W. A. Next generation reservoir computing. *Nat. communications* **12**, 1–8 (2021).
54. Griffith, A., Pomerance, A. & Gauthier, D. J. Forecasting chaotic systems with very low connectivity reservoir computers. *Chaos: An Interdiscip. J. Nonlinear Sci.* **29**, 123108 (2019).
55. Lu, Z., Hunt, B. R. & Ott, E. Attractor reconstruction by machine learning. *Chaos: An Interdiscip. J. Nonlinear Sci.* **28**, 061104 (2018).
56. Lu, Z. *et al.* Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos: An Interdiscip. J. Nonlinear Sci.* **27**, 041102 (2017).
57. Herteux, J. & R  th, C. Breaking symmetries of the reservoir equations in echo state networks. *Chaos: An Interdiscip. J. Nonlinear Sci.* **30**, 123142 (2020).
58. Hoerl, A. E. & Kennard, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**, 55–67 (1970).
59. Benvenuto, D., Giovanetti, M., Vassallo, L., Angeletti, S. & Ciccozzi, M. Application of the arima model on the covid-2019 epidemic dataset. *Data brief* **29**, 105340 (2020).
60. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).
61. WFP. Seasonal explorer (2022). [https://dataviz.vam.wfp.org/seasonal\\_explorer/rainfall\\_vegetation/visualizations](https://dataviz.vam.wfp.org/seasonal_explorer/rainfall_vegetation/visualizations).
62. (CHC), C. H. C. Chirps: Rainfall estimates from rain gauge and satellite observations (2022). <https://www.chc.ucsb.edu/data/chirps>.
63. NASA. Moderate resolution imaging spectroradiometer (modis) (2022). <https://modis.gsfc.nasa.gov/>.
64. ACLED. The armed conflict location & event data project (2022). <https://acleddata.com>.
65. Raleigh, C., Linke, A., Hegre, H. & Karlsen, J. Introducing acled: an armed conflict location and event dataset: special data feature. *J. peace research* **47**, 651–660 (2010).
66. Araujo, C., Araujo-Bonjean, C. & Brunelin, S. Alert at maradi: preventing food crises by using price signals. *World Dev.* **40**, 1882–1894 (2012).
67. WFP. Calculation and use of the alert for price spikes (alps) indicator, technical guidance note, analysis and nutrition service (2014). [https://documents.wfp.org/stellent/groups/public/documents/manual\\_guide\\_proced/wfp264186.pdf](https://documents.wfp.org/stellent/groups/public/documents/manual_guide_proced/wfp264186.pdf).
68. WFP. Economic explorer (2022). [https://dataviz.vam.wfp.org/economic\\_explorer/prices](https://dataviz.vam.wfp.org/economic_explorer/prices).

## Acknowledgements

We would like to thank Kyriacos Koupparis and Arif Husain for the continuous support. We are also grateful to Elisa Omodei and Pietro Foini for sharing their data and their knowledge on the XGBoost approach. We also want to express our gratitude to Will Fein from Microsoft AI for Good for his suggestions that improved our training procedure. This work was supported by the World Food Programme Innovation Accelerator.

## Author contributions

C.R. conceived and supervised the research. J.H. and D.P. designed the study. J.H. and D.P. conducted the calculations. All authors interpreted and evaluated the findings. All authors wrote and reviewed the manuscript.

## **Competing interests**

All authors declare no competing interests.

## **Correspondence**

Correspondence and requests for materials should be addressed to Duccio Piovani

# Supplementary Material: Forecasting Trends in Food Security: a Reservoir Computing Approach

Joschka Herteux      Christoph Räth      Amine Baha  
Giulia Martini      Duccio Piovani

December 21, 2023

## Results on 90-Day Forecast

In addition to our main analysis which was carried out on a 60 day window, we also tried forecasts on a 90 day window in the context of Yemen. As we can see in the aggregated results in figure 1 RC is confirmed as the best performing methodology even for this longer forecast horizon. Even though the numerical performances are good, we selected the 60-day window as our focus since the forecasts with all methodologies show a mean-reverting tendency for this window. This, despite not influencing the performances, limits the actual usability of the forecasts, failing to signal deteriorating curves.

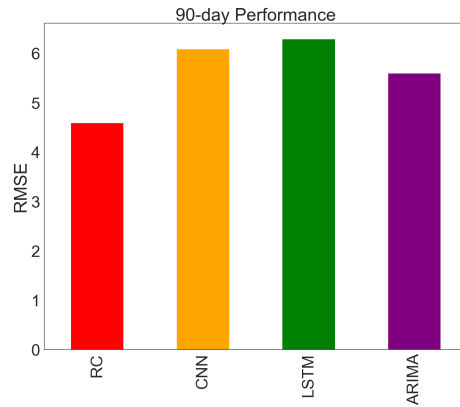


Figure 1: Aggregated RMSE of the four methodologies on 90 day forecasting windows for the regions of Yemen



## Preliminary Results from Application

While the main text compares different methodologies on historical data where some aspects of a real-life application are not fully reproduced, the RC methodology developed in this work has already been applied as a prototype in collaboration with several WFP Country Offices. First evaluations of these forecasts can be found in figure 2 for the context of Yemen and Haiti. In these cases delayed data sources have to be extrapolated with the last available value or shifted for training and prediction. In extreme cases we drop features that are not available from the analysis. While these analyses are carried out on a small

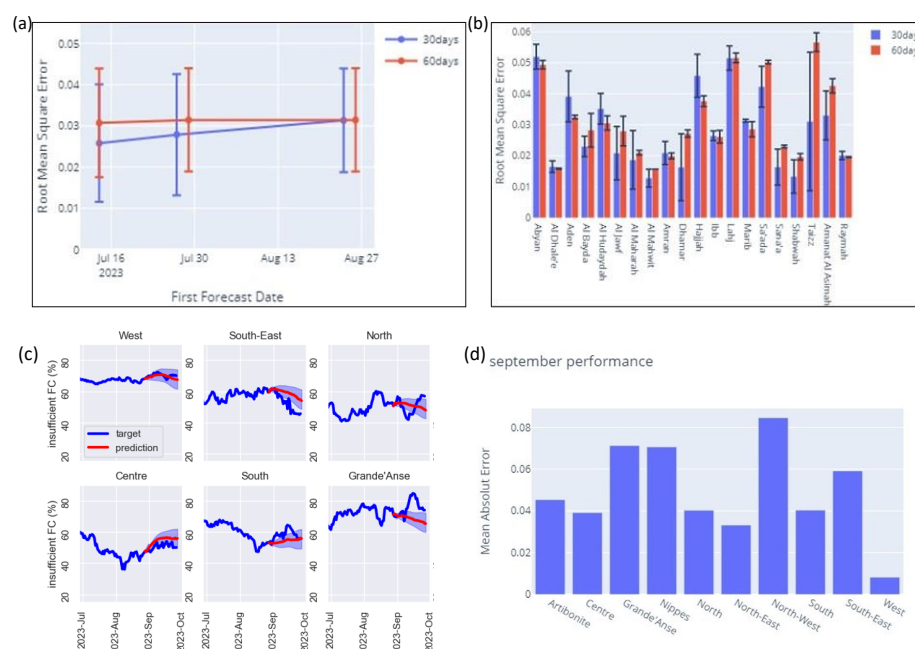


Figure 2: The RMSE obtained when producing the forecasts for Yemen Country Office **(a)-(b)** and Haiti **(d)**. These correspond the error of the Forecasting methodology in actual use and not on historical data. In **(c)** We show an example of the forecasted values (red curves) vs the actual data (blue curves). The confidence intervals represent the errors obtained on historical data.

sample they do not seem to indicate any significant performance drop which could have been caused by data delay. This is likely related to the importance of the auto-regressive component of the model we report in the manuscript. Another explanation is the importance of long term trends over short term fluctuations in time series like food- and headline inflation, which are also the most likely to experience a delay in reporting.

## RC Hyperparameters

The grid search we have carried out has given us a number of insights on the impact of hyperparameters beyond what has been reported in the manuscript. In particular we have seen that in the case of RC there are hyperparameter configurations that seem to lead to systematically good performances without needing adjustment for each split. In particular the hyperparameters of the spectral radius  $\rho$ , the input strength  $s_{input}$  and the differencing lead to strong results and show very promising regimes. We show some of these results in figure 3.

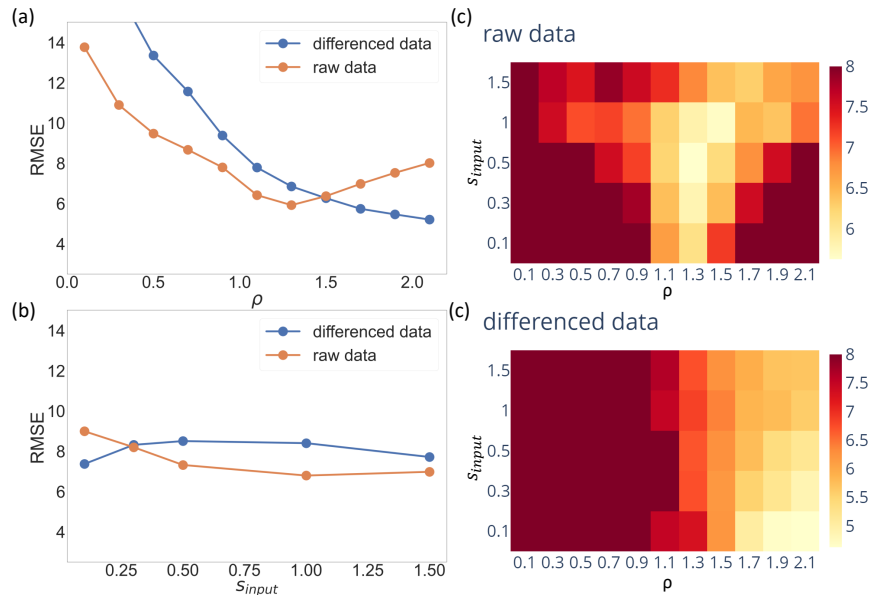


Figure 3: Influence of the hyperparameters spectral radius  $\rho$ , input strength  $s_{input}$  and differencing on the RC model as extracted from the grid search results. The value is always the median RMSE over all splits and hyperparameters that are not specified. **(a)**: The RMSE for different values of the spectral radius separated by application on differenced or raw data. **(b)**: The RMSE for different values of the input strength separated by application on differenced or raw data. **(c)**: Heatmap of the RMSE for different values of both hyperparameters applied on raw data. **(d)**: Heatmap of the RMSE for different values of both hyperparameters applied on differenced data.

As we can see the RC responds differently to the hyperparameters  $\rho$  and  $s_{input}$  depending on the use of differenced or raw data. While both approaches can be tuned to give good results, the differencing of the data seems to allow higher performances in particular in a regime of high  $\rho$  and low  $s_{input}$ . This corresponds to a regime where the memory of the RC is particularly pronounced

in contrast to recently incoming input. This indicates that the RC is making use of longer term trends instead of short term fluctuations.

The hyperparameters indicated here can be used as a starting point for application if an in depth tuning is too expensive.

## Performance of Ensemble RC with different Ensemble Sizes

Since RC is built on randomly selected parameters and is fast and efficient in its training we have used it as a building block for an ensemble model. This is meant to increase the forecasting power by stabilizing any fluctuations introduced through the influence of the stochasticity and making use of the extremely low training time of a single RC model.

To validate our choice of using an ensemble of 100 we have tested the performance against ensembles of 1, 10, 20 and 30 on the the splits introduced in the main section using the hyperparameter configurations derived from the grid search. The results are shown figure 4.

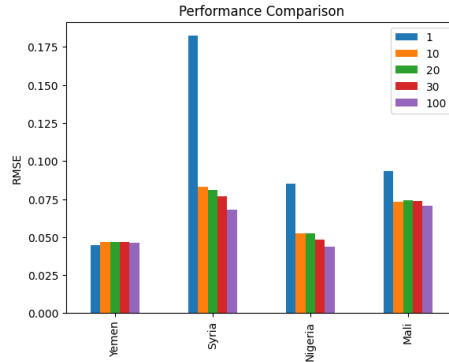


Figure 4: Aggregate RMSE of Ensemble RC with 1, 10, 20, 30 and 100 models on the four analyzed countries.

We can see that a single RC model is often too unstable in this context which might be attributed to the noisiness of the data. This is especially evident in Syria where prediction mostly fails. However, even an ensemble of 10 models increases the performance drastically. A further increase up to 100 still leads to an overall improvement, but the performance seems to be slowly plateauing and while even larger ensembles might further increase the predictive power by a small amount, the added resource requirements are unlikely to be worth the improvement.

A noteworthy outlier in this figure is the performance of a single RC model on Yemen, where it performs slightly better than the ensemble model and an increase of the ensemble size seems to add no value. This could be related to

high amount and quality of the Yemen data. It is also important to note that the hyperparameters used in this comparison were originally tuned for an ensemble of 100, thus there is a bias present in this analysis. For example particularly unstable configurations might not have been chosen for a single model, but are still suitable for the ensemble.