International Journal of Mathematics, Statistics and Operations Research
Vol. 6; Number 1; 2026, pp. 21-43
https://doi.org/10.47509/IJMSOR.2026.v06i01.02

ARF INDIA
Academic Open Access Publishing
www.arfjournals.com

A Multi-Faceted ACG-Hybrid Model for Assessment of Effects of Climate Variability on Soybean Productivity

Oloo Erick Odhiambo[a], Erick Okuto[b], Benard Okelo[c] and Samuel Oyieke[d]

[a,b,c]*School of Biological, Physical, Mathematics and Actuarial Sciences,*
[d]*School of Business and Economics,*
Jaramogi Oginga Odinga University of Science and Technology, Box 210-40601, Bondo-Kenya

ABSTRACT
The statistical techniques such as multivariate regression analysis and Autoregressive moving average (ARIMA) have been commonly applied to study plant phenology over decades. However, there has been inadequacies in understanding complex spatiotemporal big data that exists. In the recent past, Recurrent Neural Network, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been applied to the study of crop productivity, however, these models when used individually have certain shortfalls for instance, LSTM and GRU belong to the Recurrent Neural Network family but differ in architecture and complexity. Therefore, there is need to leverage on the unique properties of each model, or duo models for instance the combined CNN-LSTM hybrid model. In this note, we develop and analyze a multi-faceted ARIMA-CNN-GRU (ACG)-hybrid model for soybean productivity and future projections.

KEYWORDS
ARIMA, ACG-hybrid model, Soybean, Projection.

1. Introduction

Mathematical modelling involving crop production [21] and particularly soybean production [23] has been done over decades with several considerations given to different factors including climatic variability conditions [25]. This has necessitated development of several and different models to describe these conditions [24] and to give future projection of soybean production worldwide [22]. The link between climate factors [5] and farming output [10] is evidently gaining fame due to global climate change (see [15], [17], [18] and [20]). Exposure parameters that describe climate variability include temperature, precipitation, solar radiation, and extreme events such as meteorological drought, heavy rain and floods [1]. Temperature is a major determinant of seed germination, photosynthesis, respiration, and transpiration while precipitation is characterized by the amount of water stored in the soil [6]. Changes in these parameters have notably affected agriculture [3], and comprehending their impact on crop

CONTACT Author[b]. Email: benard@aims.ac.za

Article History
Received: 06 January 2026; Revised: 11 February 2026; Accepted: 20 February 2026; Published: 30 June 2026

To cite this paper
Oloo Erick Odhiambo, Erick Okuto, Benard Okelo & Samuel Oyieke (2026). A Multi-Faceted ACG-Hybrid Model for Assessment of Effects of Climate Variability on Soybean Productivity. *International Journal of Mathematics, Statistics and Operations Research.* 6(1), 21-43.

yields is significantly important for food production [19], its security and sustainable agriculture globally [2]. In recent decades, climate change has been marked by rising global temperatures [4], altered rainfall patterns [7], and extreme weather events [9]. The Intergovernmental Panel on Climate Change (IPCC) report in [23] shows that human actions have caused global warming, with a rise in the global surface temperature by approximately 1.1 degrees Celsius as seen in [12]. This situation could worsen, as global circulation models suggest temperatures may rise by around 4 degrees and rainfall could vary by 20% globally by 2100 as illustrated in [11]. Additionally, the IPCC report highlights significant changes in the atmosphere and oceans, with more frequent heatwaves, rainfall events, and intense droughts [14]. The Fifth Assessment Report (AR5) from the IPCC states as follows [16], "The vulnerability to current climate variability and future climate change particularly threatens the development of poor and marginalized people." In Africa, the poor are mainly smallholder farmers who rely on inconsistent and small amount of rainfall for subsistence farming [13]. Common food crops in Sub-Saharan Africa include maize, sorghum, millet, and soybean, among others [33]. Soybean farming has increasingly gained fame as a dependable stable crop for its dietary and medicinal value to the human body [20]. It is important as a rich source of protein (containing about 36-40% of protein) that contributes to nutritional needs and food security [34], capable of dealing with stunted growth [35], underweight as well as wastage in children [30] according to United States Department of Agriculture findings (USDA) as shown in [1]. The crop when consumed improves iron levels in the body, dealing with anemia which is a common nutritional disease that affects under age children and pregnant women [9]. The crop is low in carbohydrates and calories [18], which is an ideal food for diabetic persons and properly regulates blood sugar levels [25].

Table 1.: World soybean production by country (1970-2020, million metric tons)

Country/Region	**1970**	**1980**	**1990**	**2000**	**2010**	**2020**
World Total	43.6	81.0	108.3	161.2	264.8	352.6
United States	30.7	48.9	52.4	75.0	90.6	112.5
Brazil	1.5	15.1	19.9	32.7	68.7	121.8
Argentina	0.3	3.5	10.7	27.8	52.7	48.8
China	8.8	8.0	11.1	15.4	15.1	19.6
India	4.3	0.5	2.2	5.3	12.6	10.2
Paraguay	0.1	0.5	1.2	2.8	7.4	9.9
Canada	0.7	0.7	1.4	2.7	4.3	6.2
Russia	0.9	1.1	2.2	1.3	1.2	4.3
South Africa	0.0	0.0	0.2	0.1	1.9	2.2
Kenya	0.0	0.1	0.2	1.1	1.8	2.1

Table 1.1 shows the production of soybean in some selected countries in the world for the last seven decades [31] It is noted that soybean is rich in isoflavones, a chemoprotective element considered as an alternative therapy for prostate cancer, breast cancer, and cardiovascular diseases [11]. It has been noted in [32] that the isoflavones component also helps improve skin health, mental health as well as the cognitive function

of an individual. It is also a vital cash crop ranked among the top traded agricultural commodities, with significant contributions to the economies of exporting countries like the U.S.A, Brazil and Argentina as shown in a report by Food and Agricultural Organization (FAO) [3]. In Kenya, soybean production is low at an average of 20000 metrics tons (MT) against industrial demand of approximately 120,000 MT and human consumption of 10,000-15,000 MT per annum [4].

Table 2.: World soybean production by region (1970-2020, million metric tons)

Region	**1970**	**1980**	**1990**	**2000**	**2010**	**2020**
North America	31.4	49.6	53.8	77.7	94.9	118.7
South America	2.0	19.2	32.0	64.5	131.5	184.8
Asia	13.1	8.5	13.3	20.7	27.7	29.8
Europe	1.1	1.6	4.1	4.3	3.8	9.1
Africa	0.0	0.5	0.7	1.0	1.5	2.5
Oceania	0.0	0.1	0.4	0.3	1.7	2.2

The statistics in Table 1.2 obtained from [17] clearly show that more than 50% of the consumed product is imported [10]. In Kenya, soybean is commonly grown in Western, Nyanza, Riftvalley, Central and Eastern regions [6]. Western region stands out as the leading producer of soybeans, accounting for approximately 50% of countries production [7]. To bridge the demand gap,the acreage under soybeans need be increased from 2,500 ha to 135,000 Ha [8]. The target will be achieved by increased awareness of the health benefits of soybeans and as a source of income by the International Agricultural Organizations in partnership with the government [29]. At this point there is need to understand the existing predictive models useful in this work regarding soybean production. The sensitivity of stable crops such as soybeans to variations in temperature [30], precipitation [27], and other climatic conditions has garnered significant attention in recent years [5]. Some classical statistical models like AutoRegressive Integrated Moving Average (ARIMA) have been intensively employed to study and forecast agricultural yield [59]. The ARIMA (p, d, q), first proposed by Box and Jenkins in 1976, analyzes time series data by handling temporal dependencies in yield data [2]. The model relies on the linear trend of the time series and can capture both nonseasonal and seasonal data of the time series [13]. It also performs dismally with complex nonlinear data [34]. The advancements in data science and machine learning have led to modern methods for examining agricultural data sets [12]. Machine Learning (ML) techniques that have been commonly utilized in the agricultural field are K-means [18], support vector machines (SVM) [36], random forest (RF) and artificial neural networks (ANN) [31]. Further studies on machine learning led to evolution of deep learning (DL) techniques. Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) have emerged among the best Deep Learning methods to analyze spatiotemporal data [28]. CNN model is best for accurately and efficiently analyzing spatial features from remote sensing data [12]. To train effectively, the model requires large input datasets as demonstrated in [35] and [14]. The LSTM and GRU models are variants of Recurrent Neural Network (RNN) [13]. The two models are different in architecture but play almost a similar role [16]; captures temporal dependencies which makes them well-suited for time-series analysis of climate impacts on crop yields [24]. The blend of traditional techniques with deep

learning strategies are set to effectively and efficiently improve the clarity and reliability of the hybrid Deep Learning Model interaction with multiple climate elements [25]. The work by [26], in their study to predict corn and soybean yield, used random forest (RF), deep fully connected neural network (DFNN), Least Absolute Shrinkage and Selection Operator (LASSO), and CNN-RNN models. The later significantly performed better than the rest due to its ability to handle the spatiotemporal dependencies of the yield, weather patterns, and soil data [27]. The CNN-LSTM Hybrid Deep Learning model has been applied on prediction of soybean production at county-level in China [28]. Recently developed model CNN-GRU model has been applied on estimation of county-level winter wheat yield also in China [29]. A novel ARIMA-CNN-LSTM model was developed and applied on forecasting future carbon pricing in China and East Asia [30]. Most of these models have been validated through several techniques and were found to be stable and robust [33]. However, they had weaknesses based on the time constraints and the big data that some could not handle [6]. It is also noted that a lot of numerical simulations have been done to determine whether these models mimic the true outcome of the real situation [17]. Both traditional models and modern models have been used in future yield projections worldwide. There is need to leverage on the intricate climatic conditions that affects soybean yields. This is a research gap and problem that has been posed by several authors including [14] and [28]. They suggested that there is need to develop a robust model that covers all the parameters of the climatic conditions that affect soybean production. Therefore, leveraging on the combined strengths of traditional statistical models (ARIMA model) and deep learning techniques (CCN and GRU), this study aimed to assess the effects of climate variability on soybean yield in various regions of the world. A novel ARIMA-CNN-GRU hybrid model has been developed, and its performance is evaluated based on its efficiency in predicting and forecasting soybean yield under varying climate factors. The advanced predictive model unveils patterns and correlations that were previously challenging to identify.

2. Preliminaries

In this section, we review the basic concepts that are key to our study employing a multi-faceted hybrid approach to the study of the effects of climate variability on soybean yield.

Definition 1. *([6]). ARIMA is a statistical model that has AutoRegressive component of order p, Moving Average of order q and an Integrating part (I) of order d. It is best applied on time series data. The autoregression and moving average components are used to describe the stochastic process.*

Definition 2. *([8]). Recurrent Neural Network (RNN) is a type of machine learning model designed to analyze temporal and time series data.*

Definition 3. *([15]). Convolutional Neural Network (CNN) is a type of deep learning model used primarily for extraction and recognition of image from the spatial data.*

Definition 4. *([22]). Long Short-Term Memory (LSTM) is a family of recurrent neural network designed to handle long term dependencies in sequential data. Consists of memory cell structure controlled by the three gates; the input, the forget and the output gate.*

Definition 5. *([14]). Gated Recurrent Unit (GRU) is a type of RNN designed to handle long term dependencies in sequential data. It differs with LSTM in terms of architecture, with fewer parameters and controlled with two gates; updated and reset gate.*

3. Research methodology

Techniques and methods which are useful in obtaining results are given in the present chapter. We obtained historical weather patterns (temperature, rainfall, and humidity) from National Meteorological and Hydrological Service (NMHS) which sources from weather stations databases worldwide. We also utilized NASA Earth Observatory to access satellite images of climate data dating from the year 2000 to 2025. Monthly maximum (T_{max}), minimum(T_{min}), average temperature, rainfall pattern and relative humidity (%RH) have been collected. Similarly, time-series data on soybean yields, and farming practices from local agricultural departments, universities, or through field surveys were collected. The mean $\bar{x}$, variance δ^2, and covariance $cov(x, y)$ of the soybean and the main climate factors; temperature, rainfall, and humidity are calculated as:

$$\bar{x} = \frac{1}{N} \sum x_i \tag{1}$$

$$\delta^2 = \sum (x_i - \bar{x})^2 \tag{2}$$

$$cov = \frac{1}{N-1} \sum (x_i - \bar{x}))(y_i - \bar{y}), \tag{3}$$

where $\bar{x}$ is the mean, x_i is the observed variable and N number of observations.
Similarly, the variability between climate factors and soybean yield is determined through correlation analysis given as;

$$Corr(x, y) = \frac{Cov(X, Y)}{\delta_x \delta_y}, \tag{4}$$

where δ_x and δ_y are standard deviations of climate factors and soybean yield respectively.
Correlation matrices and time-series analysis interpret the relationships between climate factors and soybean productivity. Spatiotemporal data together with time-series data will be cleaned and preprocessed by handling any missing values, diversifying the training dataset, and scaling the sample through dataset imputation, augmentation, and normalizing respectively. Data have been standardized before being fed into individual architecture in multi-step approach for better training performance.
Data preprocessing minimized bias, prevents overfitting, and ensures that different scales do not affect the model's performance.

3.1. *Model formulation techniques*

The ARIMA model also named Box and Jenkins, developed in 1976 is a linear statistical model used to handle time series data. The model is expressed as $ARIMA(p, d, q)$ with parameters p, d, and q.
The model comprises of Autoregressive (AR) model (p order);

$$X_t = a + \sum_{i=0}^{p} \theta_i X_{t-i} + \varepsilon_t, \tag{5}$$

where a is the intercept, θ is the autoregressive parameter and ε $iid(0, 1)$
Moving Average (MA) model of order q;

$$X_t = \mu + \varepsilon_t - \sum_{j=0}^{q} \phi_j \varepsilon_{t-j}, \tag{6}$$

where $\mu = E(X_t)$, ϕ is the parameter of the moving average and ε $iid(0, 1)$
Augmented Dickey-Fuller (ADF) is applied to check if the data is stationary. If the data is non-stationary, the differencing aspect is applied.
The integration part of d order determines the number of times the data will be differenced to attain stationarity as expressed by

$$\hat{X}_t = X_t - X_{t-1}. \tag{7}$$

The general basic equation for the ARIMA model is given as

$$X_t = \mu + \theta_1 X_{t-1} + ... + \theta_p X_{t-p} - \phi_1 \varepsilon_{t-1} - ... - \phi_q \varepsilon_{t-q} + \varepsilon_t. \tag{8}$$

To develop an optimal ARIMA model, the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots are first simulated to determine suitable values of its parameters p and q.
Autocorrelation is similar to correlation of the times with lagged values given by $Corr(X, Y) = \frac{Cov(X,Y)}{\delta_X \delta_Y}$.

$$Corr(X, Y) = \frac{E[XY] - E[X]E[Y]}{\sqrt{E[X^2] - E[X]^2}\sqrt{E[Y^2] - E[Y]^2}}. \tag{9}$$

Suppose $X = S_{t+1}$ and $Y = S_t$ for the signal S in time t and lag n, then the ACF plot for lag n is $corr(S_{t+k}, S_{t+k-1})$. ACF is used to determine the parameter q in moving average (MA) component of the ARIMA model.
For Partial Autocorrelation Function, with lag n through the range $0 < n < k$, the PACF plot for lag n is

$$\alpha(k) = corr(S_{t+k} - P_{t,k}(S_{t+k}), S_t - P_{t,k}(S_t)). \tag{10}$$

PACF is used to determine the parameter p in autoregressive (AR) component. With defined parameters in place, model diagnosis will be conducted using ACF and PACF plots of residuals to ascertain that the residuals are normally distributed and not

autocorrelated. The Akaike Information Criterion (AIC) is thus applied help select the best forecasting model.

$$AIC = -2\ln L + 2k, \tag{11}$$

where k is the number of parameters and L is the maximum function of the likelihood function of the model. A minimal value of AIC is indicates a good ARIMA model fit. The processed ARIMA data with its residuals normally distributed are then transmitted for CNN modeling.

3.2. *Convolution Neural Network (CNN)*

A CNN is a unique case feed-forward neural networks consisting of the convolution, pooling, and fully connected layers. It receives an order 3 tensor as its input image which then sequentially goes through a series of processing. The layers extract features, reduce dimensionality, and classify objects respectively making the network easier to optimize and independent of the data size. Similarly, the model contains a rectified linear unit (ReLU), a layer free of parameters and has no effect on the size of the input. ReLU performs truncation on every individual input element, given as

$$y_{i,j,d} = max\{0, x^l_{i,j,d}\}. \tag{12}$$

Differentiating equation above we have

$$\frac{dy_{i,j,d}}{dx^l_{i,j,d}} = [x^l_{i,j,d} > 0] \tag{13}$$

If the indicator function [.] is unit, the argument is true; zero otherwise. The main purpose of a ReLU in CNN is to help increase its nonlinearity to match with higher nonlinear mapping of pixel values in the input.

3.3. *Convolution layer*

In this layer, multitudinous convolution kernels are used to detect the input image matrix of the same order. For the D number of kernels in a convolution layer, then we shall have a tensor in $\mathbb{R}^{H\times W\times D}$ with variable index $0 \leq i < H$, $0 \leq j < W$, $0 \leq d < D$. The $k-^{th}$ layer of 3-order tensor is of size $\mathbb{R}^{H^k\times W^k\times D^k}$ with variable index $0 \leq i < H^k$, $0 \leq j < W^k$, $0 \leq d < D^k$ and convolution kernel is also of order 3 tensor of size $H * W * D^k$. If the kernel is larger than $1 * 1$, then the spatial extent of the output is smaller than the input. Intertwining the kernel with the input at every spatial location corresponds to the strides s. Mathematically, Convolution procedure is given as

$$y_{i^{k+1},j^{k+1},d} = \sum_{i=0}^{H}\sum_{j=0}^{W}\sum_{d^k=0}^{D} f_{ij,d^k,d} \times x^k_{i^{k+1}+i,j^{k+1}+j,d^k}, \tag{14}$$

where x^k is the input tensor into the $k^{th} - layer$ and y or (x^{k+1}) is the output tensor. The operation is then replicated for all $0 \leq d \leq D = D^{k+1}$ and for any spatial location (i^{k+1}, j^{k+1}).

3.4. *Pooling layer*

The pooling layer reduces the spatial dimensions of the feature map thus keeping at low the number of parameters and computations in the network hence minimizing overfitting. Let $x^k \in \mathbb{R}^{H^k * W^k * D^k}$ be the input to the pooling layer or $(k^{th} - layer)$, whose spatial scale for the pooling is (H*W) as defined in the CNN structure. Dividing H^k by H and W^k by W and the strides equal to the pooling spatial scale, the output of pooling y or $(x^{k+1)}$ will be of order 3 tensor of size $\mathrm{H}^{k+1} * W^{k+1} * D^{k+1}$, with $H^{k+1} = \frac{H^k}{H}$, $W^{k+1} = \frac{W^k}{W}$ and $D^{k+1} = D^k$. A pooling layer works upon x^k channel by channel uninfluenced. Within each channel, the matrix with $H^k * W^k$ elements are divided into $H^{k+1} * W^{k+1}$ disjoint subregions, each of size $H * W$. The pooling operator then maps a subregion into a single number. The max pooling layer records the maximum values in each rectangular neighborhood of each position (i, j) in $2D$ of each input feature. Commonly utilized form of max-pooling uses filter of size $(2, 2)$ together with stride 2, which corresponds to fractionation of the feature map spatially into a regular grid of square or cubic blocks, taking maximal value or average over such blocks for each input feature.

3.5. *Fully connected layer*

A fully connected layer comes at the tail end after convolutions and pooling performs high-level reasoning, and permits the network to learn complex correlation in the data. In the fully connected layer, CNN output flattened and viewed as a mono layer. Therefore, for computation of any element in y, all elements in the input x^k have to be utilized. The output layer is determined by the weight matrix $W_{m,n}$ with m rows and n columns together with bias vector $b \in \mathbb{R}^{\geqslant}$. For the input layer x^k of size $H^k * W^k * D^k$, we use convolution kernels whose size is $H^k * W^k * D^k$, then D such kernels form an order 4 tensor in $H^k * W^k * D^k * D$ which gives the output $y \in \mathbb{R}^{\mathbb{D}}$. The output of CNN is flattened and viewed as a single vector, then fed into the GRU.

3.6. *Gated Recurrent Unit (GRU) model*

Gated Recurrent Unit (GRU) collects the output of CNN layers as its input to work on temporal dependencies in the time-series data; in our case-the impact of changing weather patterns on soybean yields over time. GRU plays a similar role to LSTM, the difference comes in with gated mechanisms; LSTM consists of a memory cell regulated by three gates; input gate, output gate and forget gate, while GRU consists of two gates; reset gate and update gate. It consists of a sigmoid (δ) activation function and *tanh* activation function, which cures the problem of vanishing gradient in GRU. Sigmoid (δ) activation given by the function $f(x) = \frac{e^x}{e^x+1}$ converts any input signal into a number between 0 and 1. Similarly tanh activation function given by the function $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ converts any input signal into a number between -1 and 1.

3.7. *Reset gate*

Reset gate regulates amount of the information in the previous hidden state to be forgotten and replaced with new information. For the 0 value, the information in the previous hidden state is discarded, while for value 1, the information is kept. The

equation is given as

$$r_t = \delta(W_{r.}[h_{t-1}, x_t] + b_r), \tag{15}$$

where:
W_r is the weight matrix associated with the reset gate.
$[h_{t-1}, x_t]$ represents the concatenation of the current input and the previous hidden state.
b_r is the reset gate bias element .
δ is the sigmoid activation function.

3.8. *Update gate*

Update gate ascertain amount of the information in the previous state that is to be retained in the current state. Helps the model capture long-term dependencies and checks on what to retain. Update gate equation is the sum of product of update gate weight matrix and the previous hidden state and input value plus the update gate bias, all past through the sigmoid function as shown below

$$z_t = \delta(W_{z.}[h_{t-1}, x_t] + b_z). \tag{16}$$

3.9. *Candidate hidden state*

This is the memory component in the GRU just like memory cell in LSTM. It is used to determine the information stored from the past time steps. It is a function of tanh function, weight matrix associated with hidden state and reset gate. The equation for the output gate is

$$\tilde{h_t} = \tanh[(W_{xh}.x_t) + W_{hh}(r_t.h_{t-1}) + b_h]. \tag{17}$$

The candidate hidden state is then used to update the hidden state based on current input information and the previous hidden state. The update gate plays a pivotal component ate this stage as shown from the equation

$$h_t = (1 - z_t)h_{t-1} + z_t.\tilde{h_t}. \tag{18}$$

3.10. *ARIMA-CNN-GRU (ACG)-hybrid model*

In this section, the architecture of ARIMA model is integrated with both CNN and GRU models to develop a novel hybrid ACG-hybrid model. The hybrid model will leverage on the strength of each to help deal with linear and non-linear data. The ARIMA(p, d, q) will first transform raw dataset into a stationary dataset by first-order differencing to remove trends and seasonality, followed by capturing of linear data features. The ARIMA residual data is then reshaped and scaled then fed into CNN-GRU architecture. CNN extracts distinctive spatial relationship features of ARIMA residual and feeds its output into the GRU layers, finally capturing the long-term temporal dependencies of these features. We shall use the dataset for the period 2010 to 2024, a period of 15 years. Cross-validation of the dataset is done through dataset training and testing. Python software will then be used extensively.

4. Main results

4.1. *ACG-Hybrid model formulation*

We carry out the mathematical development of a hybrid ACG model for soybean productivity. The model involves the integration of linear time series analysis with deep learning techniques to capture both linear and nonlinear patterns in soybean production data. We begin with the problem formulation as follows:
Consider $\{Y_t\}_{t=1}^{T}$ as a real-valued time series (TS) having $Y_t \in \mathbb{R}$ and T as the total number of observations. Then the ARIMA (p, d, q) model for a TS $\{Y_t\}$ is given as:

$$\phi(B)(1-B)^d Y_t = \theta(B)\epsilon_t,$$

where:
B-backshift operator: $BY_t = Y_{t-1}$,
$\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$-AR polynomial,
$\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$-the MA polynomial,
d-differencing order,
$\{\epsilon_t\} \sim \mathcal{N}(0, \sigma^2)$-white noise.
Now to develop a fully three component hybrid model, we consider the CNN-GRU dual part which processes the ARIMA residuals and other additional characteristics. Consider $\mathbf{X}_t \in \mathbb{R}^m$ as the feature vector given at the time t, and $R_t = Y_t - \hat{Y}_t^{\text{ARIMA}}$ as the ARIMA residuals. We take the assumption that the differenced series $(1-B)^d Y_t$ considered to be weakly stationary, and the residuals $\{R_t\}$ also considered to be a stationary process having moments which are finite up to order 4. The first preliminary result is to decompose the ARIMA component as seen in the proposition below.

Proposition 1. *The ARIMA forecast can be decomposed into linear and residual components $\hat{Y}_t^{ARIMA} = L_t + \eta_t$, where L_t is the linear forecast component and η_t captures model inadequacies.*

Proof. We get from the ARIMA model component that $\phi(B)(1-B)^d Y_t = \theta(B)\epsilon_t$. By simple manipulation on the right hand side and carrying out polynomial operators expansion gives $(1-B)^d Y_t = \sum_{i=1}^{p} \phi_i B^i (1-B)^d Y_t + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}$. Therefore, expected squared error is minimized by $\hat{Y}_t^{\text{ARIMA}}$ as $\hat{Y}_t^{\text{ARIMA}} = \mathbb{E}[Y_t | \mathcal{F}_{t-1}]$, where $\mathcal{F}_{t-1}$ is the filtration up to time $t-1$. The difference $R_t = Y_t - \hat{Y}_t^{\text{ARIMA}}$ represents the residual process. Hence, from the residual process after filtration we obtain $\hat{Y}_t^{\text{ARIMA}} = L_t + \eta_t$, where L_t is the linear forecast component and η_t represents model inadequacies.
In the next lemma we consider the residual properties of the ARIMA component.

Lemma 1. *Let $(1-B)^d Y_t$ be considered to be weakly stationary, and the residuals $\{R_t\}$ also considered to be a stationary process having moments which are finite up to order* 4. *Then the ARIMA residuals $\{R_t\}$ satisfy the following conditions:*

(i). $\mathbb{E}[R_t] = 0$,
(ii). $Cov(R_t, R_{t-k}) = \gamma(k)$ *where $\gamma(\cdot)$ is the autocovariance function,*
(iii). R_t *may exhibit nonlinear dependencies not captured by the linear ARIMA model.*

Proof. Case 1: By definition and under construction, we have $\hat{Y}_t^{\text{ARIMA}}$ as the conditional expectation, therefore $\mathbb{E}[R_t] = \mathbb{E}[Y_t - \mathbb{E}[Y_t | \mathcal{F}_{t-1}]] = 0$.
Case 2: Next we consider covariance. We have that the covariance can be calculated

as $\text{Cov}(R_t, R_{t-k}) = \mathbb{E}[R_t R_{t-k}] - \mathbb{E}[R_t]\mathbb{E}[R_{t-k}]$. Now, since $\mathbb{E}[R_t] = 0$, we have that $\text{Cov}(R_t, R_{t-k}) = \mathbb{E}[R_t R_{t-k}] = \gamma(k)$.
Case 3: The ARIMA model captures only linear dependencies through the AR and MA components. Any nonlinear patterns in the data appears in the residuals.

Next we consider the CNN Architecture. The CNN component consists of L convolutional layers. For layer l, it involves the the operation $\mathbf{Z}_t^{(l)} = f\left(\mathbf{W}^{(l)} * \mathbf{Z}_t^{(l-1)} + \mathbf{b}^{(l)}\right)$, where, $\mathbf{Z}_t^{(0)} = [R_{t-w+1}, \ldots, R_t, \mathbf{X}_t]^\top$ is the input (residuals and features over window w), $*$ denotes convolution operation, $\mathbf{W}^{(l)}$ are convolutional filters, $\mathbf{b}^{(l)}$ are bias terms, $f(\cdot)$ is the activation function. At this point we consider feature extraction capability of the CNN component in the next proposition.

Proposition 2. *In the residual sequence, the CNN component with L layers and filters k can extract local patterns of length up to k^L.*

Proof. Let layer 1 have a single filter of size k. From [6], the layer processes k consecutive residuals. So, from layer 2, each filter processes outputs from k filters of layer 1, effectively covering $2k-1$ original residuals. Therefore, by mathematical induction, at layer l, the receptive field size becomes $\text{RF}(l) = 1 + \sum_{i=1}^{l}(k_i - 1)\prod_{j=1}^{i-1} s_j$, where s_j is the stride at layer j. For stride 1, this simplifies to $\text{RF}(l) = 1 + l(k-1)$. Thus, with L layers, patterns of length $1 + L(k-1)$ can be captured. Hence, in the residual sequence, the CNN component with L layers and filters k can extract local patterns of length up to k^L.
At this juncture we consider the GRU component. The GRU processes the CNN-extracted features $\mathbf{h}_t^{\text{CNN}}$ sequentially as follows:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z[\mathbf{h}_{t-1}, \mathbf{h}_t^{\text{CNN}}]) \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r[\mathbf{h}_{t-1}, \mathbf{h}_t^{\text{CNN}}]) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_h[\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{h}_t^{\text{CNN}}]) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t,
\end{aligned}$$

where σ is the sigmoid function and $\odot$ denotes element-wise multiplication. Therefore we are able to do temporal dependency capture as seen in the result below.

Proposition 3. *The GRU component with hidden state dimension d_h can capture temporal dependencies with a vanishing gradient mitigation factor of $O(\frac{1}{d_h})$ per time step.*

Proof. To do the proof, consider the gradient flow through time. For a GRU, the gradient of loss L with respect to parameters θ is given by $\frac{\partial L}{\partial \theta} = \sum_{t=1}^{T} \frac{\partial L}{\partial \mathbf{h}_t} \frac{\partial \mathbf{h}_t}{\partial \theta}$, so the Jacobian of the hidden state transition is $\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} = (1 - \mathbf{z}_t) + \text{diag}(\mathbf{z}_t)\frac{\partial \tilde{\mathbf{h}}_t}{\partial \mathbf{h}_{t-1}}$. Since $\mathbf{z}_t \in (0, 1)$, the term $(1 - \mathbf{z}_t)$ helps preserve gradient flow. The spectral norm of this Jacobian satisfies the condition $\left\| \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \right\|_2 \leq \gamma < 1$, for some γ, which prevents exponential gradient decay/vanishment. Specifically, the update gate

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{P}_t] + \mathbf{b}_z\right) \tag{19}$$

creates adaptive shortcuts that maintain gradient flow over long sequences. The reset

gate is also given as

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{P}_t] + \mathbf{b}_r\right). \tag{20}$$

For candidate activation we have the equation given as

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_h \cdot [\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{P}_t] + \mathbf{b}_h\right), \tag{21}$$

and finally, in general the final hidden state is given by

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t, \tag{22}$$

where, $\mathbf{z}_t$ is the Update gate vector, $\mathbf{r}_t$ is the Reset gate vector, $\tilde{\mathbf{h}}_t$ is the Candidate hidden state, $\mathbf{h}_t$ is the Current hidden state, $\odot$: Hadamard (element-wise) product, $\sigma(\cdot)$ is the Sigmoid function given by $\sigma(x) = \frac{1}{1+e^{-x}}$, $\tanh(\cdot)$ is the Hyperbolic tangent function, $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h$ is the Weight matrices, $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h$ is the Bias vectors.
Now we embark on the ACG-Hybrid model formulation. The complete ACG-Hybrid model is given as

$$\hat{Y}_t = \underbrace{\hat{Y}_t^{\text{ARIMA}}}_{\text{Linear component}} + \underbrace{g(\mathbf{h}_t^{\text{CNN-GRU}})}_{\text{Nonlinear correction}}, \tag{23}$$

where $g : \mathbb{R}^{d_h} \to \mathbb{R}$ is a fully connected output layer. We first consider the universal approximation capability. The ACG-hybrid model can approximate any continuous function $f : \mathcal{X} \to \mathbb{R}$ on compact subsets of $\mathcal{X}$, where $\mathcal{X}$ is the space of time series segments, to arbitrary precision. We do the formulation in three parts:
(i). ARIMA Component: The ARIMA model provides a consistent linear estimator for the linear component of the time series. This is achieved from the computation of ARIMA component in Proposition **??** and Lemma **??**.
(ii). CNN Component: By the universal approximation theorem for CNNs, the CNN can approximate any continuous feature extraction function from the residual space. This component is reached at through the deep learning techniques and computations of Proposition **??**.
(iii). GRU Component: GRUs are universal approximators of sequence-to-sequence functions as described in [27]. Let f^* be the true generating function. Finally, we carry out decomposition of the following equation stepwisely into linear and nonlinear terms, that is, $f^*(\mathbf{Y}_{t-w:t-1}, \mathbf{X}_t) = f_L(\mathbf{Y}_{t-w:t-1}) + f_N(\mathbf{Y}_{t-w:t-1}, \mathbf{X}_t)$, where f_L is linear and f_N is nonlinear. Integrating the three components gives the required ACG-Hybrid model in Equation **??**.

Remark 1. *The ARIMA component approximates f_L with error ϵ_L. The residuals R_t contain f_N plus noise. The CNN-GRU component, as a universal approximator, can learn f_N from $\{R_{t-w:t-1}, \mathbf{X}_t\}$ with error ϵ_N. Thus, the total error is bounded by $\epsilon_L + \epsilon_N$, which can be made arbitrarily small with sufficient model capacity. This give a consequence on forecast consistency as seen in the next corollary. Under the earlier assumption and with sufficient training data, the hybrid model produces consistent forecasts $\lim_{T\to\infty} \mathbb{E}[(\hat{Y}_t - Y_t)^2] = \sigma^2_{\min}$, where $\sigma^2_{\min}$ is the irreducible error due to the innovation process ϵ_t. We have that for any $\delta > 0$, there exists model parameters Θ such that $|f^*(\cdot) - f_\Theta(\cdot)| < \delta$ almost surely, where f_Θ is the hybrid model function.*

Then

$$\mathbb{E}[(\hat{Y}_t - Y_t)^2] = \mathbb{E}[(\hat{Y}_t - f^*(\cdot) + f^*(\cdot) - Y_t)^2]$$

$$= \mathbb{E}[(\hat{Y}_t - f^*(\cdot))^2] + \mathbb{E}[(f^*(\cdot) - Y_t)^2] + 2\mathbb{E}[(\hat{Y}_t - f^*(\cdot))(f^*(\cdot) - Y_t)].$$

The cross-term vanishes as $T \to \infty$ by orthogonality of estimation error and innovation. Thus, $\lim_{T\to\infty} \mathbb{E}[(\hat{Y}_t - Y_t)^2] = \lim_{T\to\infty} \mathbb{E}[(\hat{Y}_t - f^*(\cdot))^2] + \sigma^2_{\min}$. Hence, the first term goes to 0 as $\delta \to 0$ as required.

Remark 2. *In a general set up and considering the soybean scenario, let Y_t be a representation of soybean production at time t, where $t = 1, 2, \ldots, T$. The time series is decomposed into*

$$Y_t = L_t + N_t + \epsilon_t, \tag{24}$$

where:
L_t-Linear component (captured by ARIMA),
N_t-Nonlinear component (captured by CNN-GRU),
ϵ_t-Residual error term, $\epsilon_t \sim \mathcal{N}(0, \sigma^2_\epsilon)$.
The three components are crucial representation of the final ACG model. We first consider the ARIMA component of the model.
The ARIMA (p, d, q) model for the linear component L_t takes the form

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right)(1 - B)^d L_t = \left(1 + \sum_{j=1}^{q} \theta_j B^j\right) a_t, \tag{25}$$

where:
ϕ_i-Autoregressive parameters, $i = 1, \ldots, p$,
θ_j-Moving average parameters, $j = 1, \ldots, q$,
B-Backshift operator, $B^k L_t = L_{t-k}$,
d-Differencing order,
a_t-White noise, $a_t \sim \mathcal{N}(0, \sigma^2_a)$.
In the model development it is important to consider the ARIMA with exogenous variables and for soybean production forecasting, we incorporate climate variables:

$$\left(1 - \sum_{i=1}^{p} \phi_i B^i\right)(1 - B)^d L_t = \left(1 + \sum_{j=1}^{q} \theta_j B^j\right) a_t + \sum_{k=1}^{m} \beta_k X_{k,t}, \tag{26}$$

where $X_{k,t}$ represents exogenous variables such as precipitation, temperature, and soil moisture indices.
Next we consider the CNN component for feature extraction. We construct the the

input matrix as follows: Let $\mathbf{Z}_t$ *be the multivariate input matrix at time* t*:*

$$\mathbf{Z}_t = \begin{bmatrix} Y_{t-1} & Y_{t-2} & \cdots & Y_{t-\tau} \\ X_{1,t-1} & X_{1,t-2} & \cdots & X_{1,t-\tau} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m,t-1} & X_{m,t-2} & \cdots & X_{m,t-\tau} \end{bmatrix}^T \in \mathbb{R}^{\tau \times (m+1)}. \tag{27}$$

The convolution operation on Matrix **??** *is then carried out. In deed, the the convolutional operation for the* k*-th filter at layer* l *is given by*

$$C_{l,t}^{(k)} = \sigma\left(\sum_{i=1}^{f_l} \mathbf{W}_l^{(k)} \cdot \mathbf{Z}_{t-i+1} + b_l^{(k)}\right), \tag{28}$$

where:
$C_{l,t}^{(k)}$*-Convolutional output of the* k*-th filter at layer* l*, time* t*,*
$\mathbf{W}_l^{(k)}$*-Convolutional kernel weights for filter* k *at layer* l*,*
f_l*-Filter size at layer* l*,*
$\sigma(\cdot)$*-Activation function (ReLU):* $\sigma(x) = \max(0, x)$*,*
$b_l^{(k)}$*-Bias term for filter* k *at layer* l*.*
It is worthnoting that the Max-Pooling Layer should be given deep learning considerations. This is because max-pooling operation reduces spatial dimensions given by

$$P_{l,t}^{(k)} = \max_{i \in [1, s_l]} C_{l,t+i-1}^{(k)}, \tag{29}$$

where s_l *is the pooling window size at layer* l*.*

The developed hybrid model in Equation **??** requires a complete forward pass algorithm (CFPA) which is a procedural computational technique for propagating data via a multi-relational graph to produce nodal representation in context [17]. The algorithm is in form of multi-step, iterative formulated format having data derived from near nodes collected along an attention mechanism. The importance of CFPA lies in its role of unifying the strengths of CNNs) in intricate structured data [18]. This makes the ACG-Hybrid model particularly powerful for tasks in heterogeneous information networks [19].

Algorithm 1 ACG-Hybrid model complete forward pass algorithm

Input: Historical data $\{Y_1, \ldots, Y_{t-1}\}$, exogenous variables $\{\mathbf{X}_1, \ldots, \mathbf{X}_t\}$
Output: Prediction $\hat{Y}_t$
Step 1: ARIMA Prediction
$\hat{L}_t = \sum_{i=1}^{p} \hat{\phi}_i L_{t-i} + \sum_{j=1}^{q} \hat{\theta}_j a_{t-j} + \sum_{k=1}^{m} \hat{\beta}_k X_{k,t}$
Step 2: Residual Calculation
$R_{t-i} = Y_{t-i} - \hat{L}_{t-i}$, for $i = 1, \ldots, \tau$
Step 3: CNN Feature Extraction
Construct input matrix $\mathbf{Z}_t$ from residuals using Eq. **??**
Apply convolutional layers: $\mathbf{C}_t = \text{CNN}(\mathbf{Z}_t)$ using Eq. **??**
Apply pooling: $\mathbf{P}_t = \text{Pool}(\mathbf{C}_t)$ using Eq. **??**
Step 4: GRU Temporal Modeling
Compute update gate: $\mathbf{z}_t$ using Eq. **??**
Compute reset gate: $\mathbf{r}_t$ using Eq. **??**
Compute candidate activation: $\tilde{\mathbf{h}}_t$ using Eq. **??**
Compute final hidden state: $\mathbf{h}_t$ using Eq. **??**
Step 5: Final Prediction
Nonlinear adjustment: $N_t = g(\mathbf{h}_t)$ using Eq. **??**
$\hat{Y}_t = \underbrace{\hat{Y}_t^{\text{ARIMA}}}_{\text{Linear component}} + \underbrace{g(\mathbf{h}_t^{\text{CNN-GRU}})}_{\text{Nonlinear correction}}$

The Algorithm 1 forms the first crucial steps in the center-stage computations and verification of the accuracy of the model and it utilization in analysis of the soybean future projection in-terms of its production. For a new model, it is important to consider certain properties like the loss function and it's optimization In this regard the model is trained by minimizing the combined loss function given as

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} (Y_t - \hat{Y}_t)^2 + \lambda_1 \sum_{i=1}^{p} \phi_i^2 + \lambda_2 \sum_{j=1}^{q} \theta_j^2 + \lambda_3 \|\mathbf{W}\|_F^2, \tag{30}$$

where the first term is the mean squared error, λ_1, λ_2 are the regularization for ARIMA parameters, λ_3 is the regularization for neural network weights, and $\|\cdot\|_F$ is the Frobenius norm. After loss minimization, a two-stage optimization is carried out. In the first stage we do ARIMA parameter estimation using the equation

$$\min_{\phi,\theta,\beta} \sum_{t=1}^{T} (Y_t - \hat{L}_t)^2 + \lambda_1 \sum_{i=1}^{p} \phi_i^2 + \lambda_2 \sum_{j=1}^{q} \theta_j^2. \tag{31}$$

In stage two we also carry out CNN-GRU parameter estimation using

$$\min_{\mathbf{W},\mathbf{b}} \sum_{t=1}^{T} (R_t - g(\mathbf{h}_t))^2 + \lambda_3 \|\mathbf{W}\|_F^2. \tag{32}$$

Parameter estimation is very crucial in building a robust time series system for accurate future projection [21]. It ensures that the ARIMA component captures linear patterns, trends, and seasonality, CNN component carries out proper extraction of local patterns and features and also ensures dependencies from sequential data that is to be analyzed

[28]. The GRU component is useful for deep learning of the long-term dependencies while it also helps with the gating mechanisms dynamics [22]. Optimization is also useful in error reduction in parameterizations through techniques like via grid search, Bayesian optimization, or utilization of genetic algorithms [26]. This is significant in finding the best hyperparameters and minimizing forecast errors like MAE, RMSE, or MAPE. Again, optimization helps in achieving higher accuracy in a hybrid model than in single models, makes the hybrid model more adaptive to changing patterns of climate and provides reliable and accurate projection for future decision-making [29]. Since our interest is in soybean production and forecasting then its crucial to carry out a specific formulation in this regard. We begin with the seasonal component and we realize that since there is the annual cycle of soybean production, we incorporate a seasonal component given by

$$S_t = \sum_{i=1}^{4} \left[\alpha_i \sin\left(\frac{2\pi i t}{12}\right) + \beta_i \cos\left(\frac{2\pi i t}{12}\right) \right]. \tag{33}$$

Now, in a general set up, the complete hybrid model with seasonality becomes

$$Y_t = L_t + S_t + N_t + \epsilon_t. \tag{34}$$

With the seasonality particular components we consider soybean specific input variables. and so for soybean production forecasting, we consider

$$\mathbf{X}_t = \begin{bmatrix} P_t & \text{(Precipitation)} \\ T_t & \text{(Temperature)} \\ S_t & \text{(Soil moisture)} \\ F_t & \text{(Fertilizer application)} \\ I_t & \text{(Irrigation)} \end{bmatrix} \tag{35}$$

Model formulation requires rigorous mathematical formulation using mathematical skills and statistical techniques while incorporating deep learning techniques also as seen in the earlier sections. At this point we give the ACG-Hybrid model formulation summary in the table below.

Table 3.: ACG-Hybrid model integration components

Component	**Mathematical Form**	**Purpose**
ARIMA	Eq. **??**	Capture linear trends and climate effects
CNN	Eq. **??**	Extract spatial-temporal features
GRU	Eq. **??**	Model long-term dependencies
ACG-Hybrid	Eq. **??**	Combine linear and nonlinear components

Since development of a new model requires mathematical rigour, it is important to consider certain mathematical properties of the developed model. The first case is the bias-variance decomposition. The main idea is that the total error of a model can be mathematically decomposed into three interpretable components that is $Total\ Error = Bias^2 + Variance + Irreducible\ Error$. This Helps a lot in under-

standing the source of prediction error whereby the expected prediction error is hence decomposed as seen in the equation below

$$\mathbb{E}[(Y_t - \hat{Y}_t)^2] = \text{Bias}^2(\hat{Y}_t) + \text{Var}(\hat{Y}_t) + \sigma_\epsilon^2. \tag{36}$$

In deed, Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n \sim P(x, y)$ be independent and identically distributed training data. The data generating process is given by $y = f(x) + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. A learning algorithm produces estimator $\hat{f}(x; \mathcal{D})$. For squared error loss $L(y, \hat{f}(x)) = (y - \hat{f}(x))^2$:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D},x,y}[(y - \hat{f}(x))^2] &= \mathbb{E}_x\left[\mathbb{E}_{\mathcal{D},y|x}[(y - \hat{f}(x))^2]\right] \\
&= \mathbb{E}_x\Bigg[\underbrace{(f(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x)])^2}_{\text{Bias}^2(x)} \\
&\quad + \underbrace{\mathbb{E}_{\mathcal{D}}[(\hat{f}(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x)])^2]}_{\text{Variance}(x)} \\
&\quad + \underbrace{\sigma^2}_{\text{Irreducible Error}}\Bigg]
\end{aligned}$$

Parameterize model by complexity $\lambda \in \Lambda$:

$$\begin{aligned}
\text{Bias}^2(\lambda) &= \mathbb{E}_x[(f(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_\lambda(x)])^2] \\
\text{Variance}(\lambda) &= \mathbb{E}_x[\mathbb{E}_{\mathcal{D}}[(\hat{f}_\lambda(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}_\lambda(x)])^2]] \\
\text{Total Error}(\lambda) &= \text{Bias}^2(\lambda) + \text{Variance}(\lambda) + \sigma^2
\end{aligned}$$

Optimal complexity $\lambda^* = \arg\min_\lambda \text{Total Error}(\lambda)$. For the practical estimation we consider B as the bootstrap samples $\{\mathcal{D}^{(b)}\}_{b=1}^B$ given by

$$\begin{aligned}
\bar{f}_B(x) &= \frac{1}{B}\sum_{b=1}^B \hat{f}^{(b)}(x) \\
\widehat{\text{Bias}}^2(x) &= (\bar{f}_B(x) - f(x))^2 \\
\widehat{\text{Variance}}(x) &= \frac{1}{B-1}\sum_{b=1}^B (\hat{f}^{(b)}(x) - \bar{f}_B(x))^2 \\
\widehat{\text{Total Error}}(x) &= \frac{1}{B}\sum_{b=1}^B (y - \hat{f}^{(b)}(x))^2
\end{aligned}$$

Next we consider the ensemble methods. We start by checking bagging where for B

bootstrapped models we have

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{(b)}(x)$$
$$\text{Bias}_{\text{bag}} = \text{Bias}_{\text{single}}$$
$$\text{Var}_{\text{bag}} = \rho \cdot \text{Var}_{\text{single}} + \frac{1-\rho}{B}\text{Var}_{\text{single}},$$

where ρ is average correlation between models. For boosting we check for additive model $\hat{f}^{(m)}(x) = \hat{f}^{(m-1)}(x) + \nu \cdot h_m(x)$:

$$\text{Bias}^{(m)} < \text{Bias}^{(m-1)} \quad \text{(sequential bias reduction)}$$
$$\text{Variance}^{(m)} > \text{Variance}^{(m-1)} \quad \text{(variance increases)}$$

Next we put it in a time series context. For h-step ahead forecast $\hat{y}_{t+h|t}$ we have

$$\begin{aligned}\text{MSE}(h) &= \mathbb{E}[(y_{t+h} - \hat{y}_{t+h|t})^2] \\ &= \underbrace{(\mathbb{E}[y_{t+h}|\mathcal{F}_t] - \hat{y}_{t+h|t})^2}_{\text{Squared Bias}} \\ &\quad + \underbrace{\text{Var}(y_{t+h}|\mathcal{F}_t)}_{\text{Forecast Variance}} \\ &\quad + \underbrace{\text{Var}(\varepsilon_{t+h})}_{\text{Irreducible Error}}.\end{aligned}$$

For ARIMA(p,d,q) we have $\phi(B)(1-B)^d y_t = \theta(B)\varepsilon_t$, and for forecast variance we have $\text{Var}(\hat{y}_{t+h|t}) = \sigma^2 \sum_{j=0}^{h-1}\psi_j^2$. This is very important as it helps in diagnosis of underfitting and overfitting in the data analysis scenario.

4.2. *Statistical assumptions*

4.2.1. Assumptions for ARIMA Component

The time series properties are as follows:

(i). *Stationarity after Differencing*: For $d > 0$: $(1-B)^d L_t$ is weakly stationary with constant mean, variance, and autocovariance structure.
(ii). *Invertibility Condition*: All roots of $1 - \sum_{i=1}^{p}\phi_i z^i = 0$ lie outside the unit circle.
(iii). *Causality Condition*:All roots of $1 + \sum_{j=1}^{q}\theta_j z^j = 0$ lie outside the unit circle.
(iv). *White Noise Assumption*: $a_t \sim$ i.i.d. $\mathcal{N}(0,\sigma_a^2)$, $\mathbb{E}[a_t a_s] = \begin{cases}\sigma_a^2 & \text{if } t = s \\ 0 & \text{otherwise.}\end{cases}$
(v). *Linearity in Parameters*: $\mathbb{E}[L_t|\mathcal{F}_{t-1}] = \sum_{i=1}^{p}\phi_i L_{t-i} + \sum_{j=1}^{q}\theta_j a_{t-j}$ where $\mathcal{F}_{t-1}$ is the information set at time t-1.
(vi). *Homoscedasticity*: $\text{Var}(L_t|\mathcal{F}_{t-1}) = \sigma_a^2 \quad \forall t$.
(vii). *Exogenous Variables Independence*: $X_{k,t}$ is predetermined $X_{k,t} \in \mathcal{F}_{t-1} \quad \forall k,t$.
(viii). *No Perfect Multicollinearity*: $\text{rank}[\mathbf{X}^T\mathbf{X}] = m$ where $\mathbf{X} = [X_{1,t},\ldots,X_{m,t}]$.

4.2.2. Deep Learning component assumptions

CNN Assumptions

(i). *Spatial-Temporal Invariance*: Features are translation invariant in the time dimension.
(ii). *Local Connectivity*: Patterns in soybean production are locally correlated in time.
(iii). *Stationary Feature Distribution*: $P(C_{l,t}^{(k)}) \approx P(C_{l,t+\Delta}^{(k)})$ for small Δ.
(iv). *Parameter Sharing*: $\mathbf{W}_l^{(k)} is constant across stime windows.$

4.2.3. GRU Assumptions

(i). *Markov Property*: $P(\mathbf{h}_t|\mathbf{h}_{t-1}, \mathbf{h}_{t-2}, \ldots, \mathbf{h}_1) = P(\mathbf{h}_t|\mathbf{h}_{t-1})$.
(ii). *Gradient Flow*: $\frac{\partial \mathcal{L}}{\partial \mathbf{h}_t}$ can be effectively computed via backpropagation through time (BPTT).
(iii). *Gate Saturation*: $\mathbf{z}_t, \mathbf{r}_t \in (0, 1)$ (sigmoid ensures gating).

4.2.4. ACG-Hybrid model integration assumptions

Model Combination

(i). *Additive Decomposition*: $Y_t = L_t + N_t + \epsilon_t$ is valid and complete.
(iii). *Orthogonality*: $\text{Cov}(L_t, N_t) = 0 \quad \forall t$.
(iii). *Residual Independence*: $R_t = Y_t - \hat{L}_t$ contains only nonlinear patterns.
(iii). *Scale Compatibility*: Both components operate on the same scale and measurement units.

4.2.5. Data-specific assumptions for soybean production

Agricultural process assumptions

(i). *Annual Cycle*: $Y_t = Y_{t-12} + \text{trend} + \text{random innovation}$.
(ii). *Growing Season Dependency*: $Y_t = f(\text{Climate}_{t-6:t-1}, \text{Management}_{t-3:t-1})$.
(iii). *Climate-Response Relationship*: $\frac{\partial Y_t}{\partial X_{k,t}}$ exists and is smooth for climate variables.
(iv). *Technological Progress*: $\text{Trend}(Y_t) \geq 0$ (non-decreasing yield potential).
(iii). *Bounded Production*: $Y_t \in [Y_{\min}, Y_{\max}]$, where bounds are physiologically determined.

Exogenous variable assumptions

(i). *Measurement Quality*: $X_{k,t} = X_{k,t}^{\text{true}} + \eta_{k,t}, \quad \eta_{k,t} \sim \mathcal{N}(0, \sigma_{\eta,k}^2)$.
(ii). *Lead-Lag Relationships*: $\text{Cov}(Y_t, X_{k,t-\tau}) \neq 0$ for some $\tau \in \{1, 2, \ldots, 6\}$.
(iii). *Stationary Climate Statistics*: $\mathbb{E}[X_{k,t}] \approx$ constant over training and testing periods.

4.2.6. Training and optimization assumptions

Loss function assumptions

(i). *Quadratic Loss Appropriateness*: $\mathcal{L}(\hat{Y}_t, Y_t) = (Y_t - \hat{Y}_t)^2$ is appropriate for production forecasting.

(ii). *Convexity in Parameters*: $\mathcal{L}(\Theta)$ is convex in ARIMA parameters $\{\phi, \theta, \beta\}$.
(iii). *Differentiability*: $\frac{\partial \mathcal{L}}{\partial \Theta}$ exists almost everywhere.

Optimization assumptions

(i). *Two-Stage Separability*: $\arg\min_{\Theta} \mathcal{L}(\Theta) = [\arg\min_{\Theta_{\text{ARIMA}}} \mathcal{L}_1, \arg\min_{\Theta_{\text{NN}}} \mathcal{L}_2]$.
(ii). *Gradient Descent Convergence*: $\Theta^{(k+1)} = \Theta^{(k)} - \eta \nabla \mathcal{L}(\Theta^{(k)}) \to \Theta^*$.
(iii). *Learning Rate Schedule*: $\sum_{k=1}^{\infty} \eta_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \eta_k^2 < \infty$.
(iv). *Parameter initialization*: $\mathbf{W} \sim \mathcal{N}(0, \sigma_w^2), \quad \sigma_w^2 = \frac{2}{n_{\text{in}}+n_{\text{out}}}$.

Temporal assumptions

(i). *Structural Stability*: $P(Y_t|\mathcal{F}_{t-1})_{\text{train}} \approx P(Y_t|\mathcal{F}_{t-1})_{\text{test}}$.
(ii). *No Structural Breaks*: No abrupt changes in production technology or climate patterns.
(iii). *Horizon Limitations*: Forecast horizon $h \leq 12$ months (due to annual cycle)
(iv). *Information Set Completeness*: $\mathcal{F}_t$ contains all relevant information up to time t.

Validation assumptions

Model evaluation

(i). *Independent Test Set*: $\{(Y_t, \mathbf{X}_t)\}_{\text{test}} \perp \{(Y_t, \mathbf{X}_t)\}_{\text{train}}$.
(ii). *Sufficient Data*: $T_{\text{train}} > 10\times$ number of parameters.
(iii). *Normality of Errors*: $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$ for inference purposes.
(iv). *No Data Leakage*: Test data is not used in any part of model training or selection.

Practical implementation assumptions

Computational assumptions

(i). *Numerical Stability*: Condition number$(\mathbf{X}^T\mathbf{X}) < 10^3$.
(ii). *Floating Point Precision*: $\text{Error}_{\text{numerical}} \ll \text{Error}_{\text{model}}$.
(iii). *Memory Constraints*: Batch size$\times$Sequence length$\times$Features $<$ Available RAM.
(iv). *Training Time*: $T_{\text{convergence}} <$ Practical time constraints.

These assumptions provide the theoretical foundation for the ARIMA-CNN-GRU hybrid model. While perfect satisfaction of all assumptions is rarely achieved in practice, awareness of potential violations and appropriate diagnostic checks ensure robust application to soybean production forecasting. The hybrid nature of the model provides some robustness to assumption violations through complementary strengths of statistical and deep learning approaches.

4.2.7. Risk assessment and violation consequences

It is important to note that violation of any of the above assumptions can lead to serious mathematical and analysis errors and inaccurate results. However, there are mitigation strategies should there be any risk that emanates from the assumptions

Table 4.: Assumption violation consequences and mitigation strategies

Assumption	**Violation Consequences**	**Mitigation Strategies**
A1 (Stationarity)	Biased parameters, spurious relationships	Differencing, detrending, transformation
A4 (Normality)	Invalid inference, poor prediction intervals	Bootstrap methods, robust estimation
D2 (Orthogonality)	Overfitting, double counting of patterns	Regularization, component analysis
E1 (Annual Cycle)	Poor long-term forecasts	Seasonal ARIMA, Fourier terms
I1 (Structural Stability)	Forecast failure, model breakdown	Rolling window, adaptive learning
G1 (Quadratic Loss)	Poor performance for extreme events	Quantile loss, robust loss functions

5. Concluding remarks

Soybean production remains instrumental to farmers worldwide and its optimal production is important in ensuring food security in the world. We give a conclusion of our study in this chapter and finally give recommendations for future studies. We intentioned develop an ACG-hybrid model for assessing the effect of climate variability on soybean yield. We have have developed a new multi-faceted hybrid model for assessment of effects of climate variability on soybean productivity as seen in Equation.

References

[1] Abbas, A. C., Yuansheg, J., Asad, A., Waqar, A., Ilhan, O., Avik, S., Fayyaz, A., Modeling the impact of climatic and non-climatic factors on cereal production: Evidence from Indian agricultural sector,*Journal of Environmental Science and pollution Research*, 29(2021), 1-20.

[2] Agarwal S., Tarar S., A hybrid approach for crop yield prediction using machine learning and deep learning algorithms, *J. Phys.: Conf. Ser.,* **1714**(1) (2021), 90-102.

[3] Bali V., Kumar A., Gangwar S., A novel approach for windspeed forecasting using LSTM-ARIMA deep learning models. *International Journal of Agricultural and Environmental Information Systems*, 11(3), (2020), 13-30.

[4] Cheng-Zhi C., Cong-Jian L., Global Warming and World Soybean Yields: ARIMA-Based Projection Analysis, *Climatic Change Stud.,* 7(2021), 90-104.

[5] Cao J., Integrating multi-source data for soybean yield prediction across China using machine learning and deep learning approaches, *Agric. For. Meteorol.,* 297(2021), 78-99.

[6] Chi Y. N., Time Series Forecasting of Global Price of Soybeans Using a Hybrid SARIMA-NARNN Model, *Data Sci.: J. Comput. Appl. Inform.,* 5(2021), 85-101.

[7] Duman H., Forecasting Soybean Production in Turkiye: A Comparative Analysis of Automated and Traditional Methods, *Igdir Univ. Agro-Sci. J.,* 2(2024), 90-109.

[8] Elavarasan D., Durairaj Vincent P. M., Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications, *IEEE Access,* 8(2020), 86886-86901.

[9] Esteban O. T., Ismail P., Agustin G., Assessing Impact of climate change on soybean production in Argentina, *Journal on Climate Services*, 43(2024), 10-58.
[10] FAO, Hybrid linear time-series approach for long-term forecasting of crop yield, FAO Agric. Report, 2024.
[11] Gounmeein A. l, Ismail M. T., Forecasting the soybean yield using a Box-Jenkins seasonal ARIMA model, *International Journal of Mathematics and Computer Science*, 15(2020), 27-40.
[12] IPCC, Intergovernmental Panel on Climate Change report on Climate Change Impacts, Adaptation and Vulnerability, 2007.
[13] Karim M. E., Foysal M., Das S., Soybean yield prediction using bi-LSTM and GRU-based hybrid deep learning approach, *Proceedings of Third Doctoral Symposium on Computational Intelligence*, 6(2022), 701-711.
[14] Lei J., Yingchao Z., Kaijian H., Bangzhu Z., Soybean future price forecasting based with ARIMA-CNN-LSTM model, *Journal of Procedia Computer Science*, 162(2019), 30-38.
[15] Malladi R. K., Dheeriya P. L., Time series analysis of soybean yield returns, *Journal of Economics and Finance*, 45(2023), 75-94.
[16] NBSC, National Bureau of Statistics of China report on National Economic and Social Development, 2023.
[17] Nelson D. M., Pereira A. C., De Oliveira R. A., Soybean yield prediction with LSTM neural networks. *International Joint Conference on Neural Networks*, 23(2017), 1419-1426.
[18] Nevavuori J., End-to-End 3D CNN for plot-scale soybean yield prediction using multitemporal UAV-based RGB images, *Precis. Agric.*, 25(2023), 834-864.
[19] Poudel S., Shaw R., The Relationships between Climate Variability and Crop Yield in a Mountainous Environment in Lamjung District, Nepal, *Journal on Climate*, 4(2026), 56-78.
[20] Roondiwala M., Patel H., Varma S., Predicting soybean prices using LSTM. *International Journal of Science and Research*, 6(2017), 1754-1756.
[21] Saidulu B., Sharma M., Forecasting Soyabean Crop Production using ARIMA Model, *Int. J. Innovative Sci. Res. Technol.,* 10(2025), 3016-3023.
[22] Shammi S. A., Meng Q., Use time series NDVI and EVI to develop dynamic crop growth metrics for yield modeling, *Ecol. Indicat.,* 121(2021), 107-124.
[23] Shen C., Analysis of detrended time-lagged cross-correlation between two nonstationary time series, *Phys. Lett.,* 379(2015), 680-687.
[24] Shibo G., Erjing G., Zhentao Z., Meiqi D., Xi W., Zhenzhen F., Kaixin G., Wenmeng Z., Wenjing Z., Zhijuan L., Chuang Z., Xiaoguang Y., Impacts of mean climate and extreme climate indices on soybean yield and yield components in Northeast China, *Journal on Science Total Environment*, 838(2022), 156-284.
[25] Siami S., Tavakoli N., Namin A. S., A comparison of ARIMA and LSTM in forecasting time series. *IEEE International Conference on Machine Learning and Applications*, 17(2018), 1394-1401.
[26] Sun J., Liping D., Ziheng S., Yonglin S. and Zulong L., County-level soybean yield prediction using deep CNN-LSTM model, *Sensors*, 19(2019), 1-18.
[27] USDA, Report on Dietary Guidelines Support Consumption of Soy Products, 2021.
[28] VanKlompenburg T., Kassahun A., Catal C., Crop yield prediction using machine learning: A systematic literature review, *Comput. Electron. Agric.,* 177(2020), 105709.
[29] WHO, World Health Organization Report on Noncommunicable Diseases, 2023.
[30] Wooldridge J. M., *Introductory econometrics: A modern approach,* Springer-Verlag New York, 2015.
[31] Xiong T., Bao Y., Hu Z., Multiple-output support vector regression with a firefly algorithm for interval-valued for soybean yield forecasting. *Knowledge-Based Systems*, 55(2014), 87-99.
[32] Yadulla S., Yasa S. V. Pothu N., A Deep Learning Model for Soybean Yield Prediction, *EasyChair Preprint 10612*, 2023.

[33] Zhao A., Zhang A., Liu X., Cao S., Spatiotemporal changes of normalized difference vegetation index (NDVI) and response to climate extremes and ecological restoration in the Loess Plateau, China, *Theor. Appl. Climatol.,* 132(2018) 555-567.

[34] Zhou Y. K., Detecting Granger effect of vegetation response to climatic factors on the Tibetan Plateau, *Prog. Geogr.,* 8(5), (2019), 718-730.

[35] Zuur A. F., Leno E. N., Walker N. J., Saveliev A. A., Smith G. M., *Mixed Effects Models and Extensions in Ecology with R 574,* Springer, New York, 2009.