Forecasting Commodity Prices in Sub-Saharan Markets

twelve weeks thesis as part of the
Development Economics (M.Sc.) degree
at the University of Göttingen

Submitted on: September 28, 2021

By: Yonas Menghis Berhe
from Asmara
Matriculation Number: 21802486

1. Reviewer: Prof. Bernhard Brümmer
2. Reviewer: Dr. Bernhard Dalheimer

Georg-August-Universität Göttingen
Chair of Agricultural Economics and Rural Development

Table of Contents

Abstract

In developing nations, prices of agricultural commodities significantly affect people's livelihood and domestic economics. Fluctuation in food prices directly impact real incomes and agricultural production. Thus, accurate forecasts of commodity prices are vital in enhancing decision making and policy planning. This thesis aims to identify the most suitable methods in terms of forecasting monthly wholesale prices of Wheat, Maize and Sorghum in Addis Ababa, Khartoum and Nairobi commodity markets respectively.

The study compared the forecast performance of statistical and machine learning approaches namely: "ARIMA, SARIMA, K-nearest neighbor (KNN), Support Vector Machines(SVM), and Artificial Neural Networks (ANN)." Forecast accuracy was assessed with "mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean squared error (RMSE)" over different forecast horizons.

The experimental results suggest that ARIMA and SARIMA were most appropriate methods for short-term forecasting. While, ANN provided higher accuracy over long and short-term SVM and KNN maintained average forecast performance in both horizons. In addition machine learning models are better at capturing price trends and oscillations than statistical models.

List of Figures

List of Tables

List of Abbreviations

ACF	Autocorrelation function
ADF	Augmented Dickey and Fuller
AIC	Akaike information criterion
ANN	Artificial neural network
AR	Autoregressive
ARIMA	Autoregressive integrated moving average
ARMA	Autoregressive moving average
CV	Cross validation
FFNN	Feed Forward Neural Network
FFNN-CV	Cross validation optimized Feed Forward Neural Network
JB	Jarque and Bera
KNN	K-nearest neighbor
KNN-CV	Cross validation optimized K-nearest neighbor
KNN-MIMO	K-nearest neighbor with Multiple input multiple output forecasting
KNN-Recursive	K-nearest neighbor with recursive forecasting
KPSS	Kwiatkowski–Phillips–Schmidt–Shin
MA	Moving average
MAE	Mean absolute error
MAPE	Mean Absolute Percentage Error
MIMO	Multiple input multiple output
ML	Machine learning
MSE	Mean squared error
PACF	Partial autocorrelation function
RMSE	Root Mean Square Error
SARIMA	Seasonal autoregressive integrated moving average
SVM	Support vector machine
SVR	Support vector regression
SVR-Linear	Support vector regression with linear kernel
SVR-RBF	Support vector regression with Radial basis function
TDNN	Time-Delay Neural Network

Chapter 1. Introduction

In recent years, commodity prices have fallen, therefore both countries, as well as producers who are dependent on them, recognize that their pace of income is not in conjunction with the import costs and production costs. In fact, the price fluctuations in response to “normal demand-supply changes” are larger than those in other prices, which increases the producers’ cost of working capital to stock holding, while some prices also face “unpredictable and uncontrollable” shocks from external factors. Thus, producers encounter the “dual problem” of “lower returns and higher risks.” In fact, all the countries which produce commodities both developed and developing, face these problems although they are more serious in developing and least developed countries (Page and Hewitt 2001).

In the least Developing Countries(LDC’s) the agricultural sector is the most dominant part of their economies. This sector contributes to a significant share of the gross domestic product GDP (roughly 30 to 60 percent), a large proportion of the labour force of the nation is employed by this sector (40 to 90 percent), it is a primary source of foreign exchange and most essentially, it supplies essential food items and offers subsistence income to the majority of the population of the LDCs (FAO 2002). Thus the agricultural sector is the main driver for substantial economic growth as it helps in alleviating poverty and enhancing the situation of food security (FAO 2002). Consequently, commodity prices in LDCs play a crucial role in making food accessible to consumers and their standard of living as they directly influence real incomes. The prices of food have a significant effect on the livelihood of the poor in particular, who spend the majority of their income on food and basic necessities. Such occurrence is more severe in the least developed regions such as Sub-Saharan Africa (SSA) who go through conditions of extreme poverty.

Commodity prices are extremely “volatile international asset prices.” According to studies, the failed attempts of commodity price forecasting are mainly attributed to their relatively high volatility. However, the price forecast of most commodities has not faced the challenge of volatility sufficiently. Therefore in order to address these failures in price forecast, confidence intervals have been used to generate price forecasts as they reduce the ex-post forecast errors. It has been observed through studies that in the context of Sub-Saharan Africa, nine out of eighteen commodities experienced major changes in terms of volatility over the years.

Therefore considering the fundamental role of commodities in Sub Saharan Africa, forecasting volatility is very useful in economic decision making. Commodity forecasting volatility is considered to be crucial in multiple aspects of the influence of "international asset prices as well as monetary policy" (Ocran and Biekpe 2007). Various studies have emphasized that monitoring and forecasting food prices play a huge role in alleviating and controlling impacts posed by unstable prices as the problem of commodity dependence of the countries of Sub Saharan Africa is not only an export side development problem but is also interlinked with commodity import dependence (Brown, Pinzon, and Prince 2008). Appropriate forecasting systems prevent farmers and consumers against price risk and ensure proper balancing between demand and supply of food. In addition, forecast information keeps stakeholders well informed, improves planning and developing early warning and intervention systems and limits the cost of recovery (Brown, Pinzon, and Prince 2008). Moreover, forecast data can be used as input alongside other information regarding food security in order to provide insights on the conditions of food dynamics and aids in making appropriate policy decisions and reforms. Therefore, developing countries, especially SSA nations, can leverage the insights gained from predicting commodity prices to make appropriate decisions and formulate policy reforms.

There is high instability in commodity prices as well as their production. They are affected by unpredictable natural calamities such as irregular rainfalls, drought, floods as well as pests and disease. Commodity prices are not only determined by domestic and international markets but also by the level of commodity imports and trade practices as well. These factors result in an increased price variability and complexity to food production, which makes forecasting commodity prices challenging.

Machine Learning (ML) field in the computational sciences is gaining interest in modeling food dynamics in recent years. Machine learning methods possess the ability to predict the risk of famine and identify patterns in rainfall, crop production as well as consumption by improving classification accuracy (Okori and Obua 2011). Moreover, several research studies have highlighted that statistical analysis of time series data can also play a vital role where past yields and commodity price data can be utilized to estimate and predict future trends (Michel and Makowski 2013). Furthermore, ML methods exhibit superior performance with a nonlinear, nonparametric approach to analyzing time series data (Fischer, Krauss, and Treichel 2018) (Cao and Tay 2003) (Ticlavilca, Feuz, and McKee 2010).

This thesis focuses on the last approach of forecasting food prices to get future insights. To that end this study applied Statistical and Machine Learning approaches in forecasting prices of the most important commodities from SSA markets. The goal is to identify the most appropriate method by comparing the forecast performance of each approach.

1.1 Research Gap

Forecasting commodity prices is essential for alleviating market risk, reducing uncertainty of production, supporting macroeconomic regulation and policy formulation. Recognizing this importance, a considerable number of studies have applied statistical and ML methods for accurate prediction of commodity prices from a variety of viewpoints and in different settings.

The most common statistical methods include “Autoregressive moving average (ARIMA), Seasonal Autoregressive integrated moving average (SARIMA), Exponential smoothing (ES), and Holt-Winters (HW).” These methods are popular in forecasting any time series data. However, as commodity prices became more volatile, ML models with a sophisticated and self-learning capability have become more advantageous for price forecasting. In recent years methods such as “K-nearest neighbor (KNN), Support Vector Machines (SVM) and Artificial neural networks (ANN)” received a lot of attention.

The literature on commodity price forecasting varies greatly not only on the methods employed by each study but also on the diverse settings and forecast scenarios under which the studies have been carried out. Various kinds of forecasting models have been employed for a wide variety of purposes and agricultural commodities. For instance in India (Cao and Tay 2003) employed ARIMA to predict the price of cotton in major producing states. Exponential smoothing model (ES) is applied to carrot price prediction in China (Li, Xu, and Li 2010). Multivariate regression model was used to predict the price of avocado in the US by (Evans and Nalampang 2009). Other studies examined the prediction performance of a specific method on prices of several commodities. Such as the study by (Ming-hua et al. 2012) predicted several agricultural commodity prices using back propagation neural networks. (D. Zhang et al., 2018) predicted the prices of soy beans in China using QR-RBF neural network and (D. Zhang et al., 2020) also employed a model selection framework in forecasting the prices of different agricultural products. (Ayankoya et al., 2016) used back propagation neural network approach to predict grain commodities price in south Africa. Some studies even used a hybrid approach, such as (Xiong et al., 2015) predicted cotton and corn price using a combination of VECM and SVR and some emphasized on the forecast horizon (Li et al., 2010).

The literature consistently reported superiority of ML compared to statistical models. However, the models yield different forecast accuracies in different settings, which confirms the "no free lunch theory", stating that no single model is appropriate for all commodities (Wolpert & Macready, 1997). Since, the performance of a forecasting method depends on several aspects such as the size, type, frequency of the input data as well as the forecast horizon, then, it is most likely that forecast accuracy changes according to specific forecast scenarios. One way of solving this problem is to compare the performance of several statistical and ML forecasting techniques and select the most favorable one.

Although, several studies exists on comparative study of times series forecasting methods, they tend to be widespread in their approach. These studies explore forecasting methods either by focusing on specific domains *(Athiyarath et al., 2020)* or applying only statistical or ML methods (Iqbal, 2001) (Ocran & Biekpe, 2007) (Varma & Padma, 2019). Some limit their research to a certain commodities *(Kohzadi et al., 1996) (Jeong et al., 2017)* and others to certain region *(Griffith & Vere, 2000)*.

There is a lack of similar research studies for determining the appropriate model to apply in forecasting commodity prices by considering all aspects of least developed nations and especially those of Sub-Saharan Africa countries (SSA), which suffer conditions of extreme poverty and food insecurity. This makes it challenging for decision makers in those nations to identify the most appropriate and optimal model that matches their forecasting needs, given a specific type of commodity and the prevailing economic conditions. Applying forecasting methods in a different setting and for a different purpose other than the setting for which they are initially developed and studied for, can result in low model performance and forecast accuracy. This is because the model fails to take into account the specific characteristics of a given situation, since it has never been trained on data that represents those particular dynamics (Das, 2019). Therefore it's crucial to conduct studies focusing on a comparison of accuracy between several forecasting methods by taking into account specific commodities of the Sub-Saharan Africa(SSA) region.

1.2 Research Objectives

As previously stated, accuracy of a prediction model mainly depends on the given dataset and forecasting horizon. Based on this premise, this work aims to predict prices of three commodities from SSA namely; Wheat, Maize and Sorghum from Ethiopia, Kenya and Sudan respectively using Statistical and ML approaches. Furthermore, this study attempts to investigate the methods and comprehensively examine the predicted data in order to answer the following research questions:

RQ1: How does forecasting with "Machine Learning methods" perform in comparison to "Statistical methods" in predicting the given commodity price dataset?
RQ2: Which approach is most appropriate for a given the forecast horizon?
RQ3: Which method has an overall best forecasting performance?

1.3 Thesis Outline

The remainder of this thesis is arranged in the following manner:: Chapter 2 contains relevant literature on commodity price forecasting. Chapter 3 provides a theoretical background of time series analysis and introduces the selected statistical and machine learning models. Chapter 4 presents the commodity dataset. Exploratory data analysis on the prices is carried out and explained briefly. Chapter 5 covers the proposed methodology. Here, data preprocessing and model designs and forecasting procedures are discussed. Later, the hyperparameter optimization and overview of the forecast accuracy evaluation metrics are also presented. Chapter 6 compares and contrasts the performance of the implemented forecasting methods according to the forecast accuracy measurements. Results and evaluations of the comparison are also discussed in this chapter. Chapter 7 draws conclusions on the basis of the objectives and motivation of this study. This chapter also discusses the limitations of this research and suggested research topics for the future.

Chapter 2. Literature Review

In the previous sections we explored the importance of predicting agricultural prices in avoiding adverse effects of market risk to producers and maximizing real incomes of consumers. In addition, insights about future price trends are increasingly being used as input in formulating food related economic policies and stakeholders decision making. Thus, the study of agricultural price prediction has currently been an area of numerous and diverse research undertakings. This chapter covers related works of empirical research on time series prediction of agricultural prices. The compilation of the literature has been done to facilitate a comprehensive understanding of the methods and procedures involved in the study.

The literature on predicting time-series commodity prices can be classified based on their forecasting approaches into "conventional statistical models, machine learning models, and hybrid models" that integrate all of them (Chou & Ngo, 2016). Statistical models are models in which the input and output relationship are linked through a mathematical equation and rely on historical data. The most common statistical models are "autoregressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA) and general exponential methods" (Ruekkasaem & Sasananan, 2018). These methods are part of the Box-Jenkins approach that comprise a commonly used class of linear models in order to forecast univariate time series data (Box & Jenkins, 1990).

However, at present Machine learning (ML) models have been deployed for a wide variety of applications including time series forecasting due to their superior capability to process intricate input and output relationships. Although ML borrows several concepts from mathematics and statistics, it offers a different approach than statistical methods. With ML the main emphasis is in developing an algorithm that has the potential of learning the structure of their parameters based on an observed data as well as adapting by making best use of an objective or cost function (Sajda, 2006). This makes it possible to obtain information through automatic computations (Okori & Obua, 2011). Approaches based on ML includes but are not limited to "K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Artificial Neural Network (ANN), Time-Delay Neural Network (TDNN), Random Forest (RF) and Ensemble machine learning methods."

Finally we have “Hybrid models, which combine the functions of statistical and ML models in various ways for the purpose of time series forecasting” (Wang et al., 2019). As we discussed earlier, model performance varies depending on the given forecast scales. Thus, the forecast horizon is also categorized into three types namely, the short-term forecasts which range between an hour to a week , mid-term forecasts which range between a month to a year , and long-term forecast which refers to forecast horizon that exceeds a year (D. Zhang et al., 2020).

Numerous empirical research studies on forecasting commodity prices have been undertaken in recent years employing various forecasting techniques. In their study (Jha & Sinha, 2013) used ARIMA and TDNN in order to model and forecast the oilseed soybean and rapeseed mustard wholesale price per month in the Indian markets of Indore and Delhi. The TDNN model offered better forecast accuracy of “Root mean squared error (RMSE) and Mean absolute error (MAE)” in comparison to the ARIMA model. In a similar study (Cenas, 2017) “compared the performance of ARIMA and Kalman algorithms in forecasting rice prices in the Philippines.” Wholesale prices from 1990 to 2014 were used for training and it was found that the “Kalman model” exhibited lower in “MAE and RMSE” compared to the ARIMA model.

(Kaur et al., 2014) investigated different techniques of data mining such as “K-means, KNN and SVM” on different data sets to solve the problem of crop price prediction. They found satisfactory results that data mining techniques have the ability to solve the yield prediction problem. (Luo et al., 2010) focused on “back propagation neural network,” a neural network model based on an algorithm that is genetic and “Radial Basis Function (RBF) neural network model” to forecast Lenten‘s price for the Beijing wholesale markets. The genetic algorithm neural network model outperformed all other models. Similar models have also been used to predict prices of tomatoes using a three year weekly price data and found the same results (Subhasree & Arun, 2015). However, in another study that made use of three year weekly price of tomatoes in India for prediction, the results of the experiment on “radial basis function (RBF)” showed better performance than “back propagation algorithms (BP)” (Nasira & Hemageetha, 2012).

SVM has been widely used in different applications including in commodity price prediction. such as the study by (Jeong et al., 2017) using data provided by the enterprise resource planning (ERP) system of the Agricultural Product Processing Center (APC). The researchers utilized SVR (Support Vector Regression) to predict domestic onion prices in order to stabilize prices in the South Korean local markets. The paper by (Ticlavilca et al., 2010) used a Multivariate Relevance Vector Machine (MVRVM) to predict prices of cattle, hogs and corn using monthly data. Evaluation results showed good model performance for one and two months but accuracy started decreasing for the third month. The MVRVM model outperformed the ANN except for the corn prices.

Hybrid models such as the integration of “ARIMA, artificial neural networks (ANN), and fuzzy logic” have been employed to predict exchange rates and gold prices (Khashei et al., 2009). Another study by (Anggraeni et al., 2019) suggested another combination to predict the price of rice such as the “ANN and autoregressive integrated moving average with exogenous variables (ARIMAX).” Similarly (Fang et al. 2020) worked with a “combined neural network , SVM, and ARIMA, with ensemble empirical mode decomposition (EEMD)” to forecast commodity prices of both “rice and wheat.” In general the research showed that hybrid models have higher prediction potential than single models (Fang et al. 2020)

From the extant literature it is evident that predicting commodity prices with a variety of approaches requires some narrowing. Forecasting models that are suitable for certain situations are not necessarily and directly appropriate for some other situations. Therefore, scoping becomes unavoidable. (Mitchell, 1997) pointed out that the researchers task is to discover the approach that is most suitable with the available training dataset. Therefore, this study will focus on the comparison of the performance of common statistical and machine learning methods in forecasting crop prices from the main sub-Saharan African commodity markets.

Chapter 3. Theoretical background

In the last section we explored the application of various methods in forecasting time series commodity prices. This chapter covers the theoretical background of time series analysis as well as important aspects of forecasting. Moreover, mathematical foundations of the selected “statistical and machine learning” approaches are discussed.

3.1 Statistical Time series analysis

A “time series analysis” refers to a discrete or continuous observation of data that occurs sequentially in time (Brockwell and Davis 2002). These observations are a list of numbers consisting of time units collected at regular time intervals (Hyndman and Athanasopoulos 2018). Although there is no binding data size, the standard interval period between two observations could be daily, weekly and monthly. The higher frequency of the time interval the more information could be gained (Brockwell and Davis 2002). Stock prices, commodity prices, product sales, energy consumption data, weather data, etc are some of the examples of time series data. Therefore, the primary goal of “time series analysis” is to find and extract information about the dependency between close observation from historical data by formulating a mathematical equation and forecasting future values. In time series analysis, historical observation of the same variable can be analyzed either solely (univariate analysis) or in relations to another time series variables (multivariate analysis). Time series data can also be decomposed into several components in order to understand the underlying pattern (Brockwell and Davis 2002). These components are:

Trend: the increase and decrease in values of the “time series” in the long term. It also states the change in the direction of the times series (Hyndman and Athanasopoulos 2018).

Seasonal: refers to the repeating short-term cycle or patterns that occurs in the time series during the seasons through the year. These are caused due to some seasonal factors such as weather conditions and traditions harvest habits (Hyndman and Athanasopoulos 2018).

Cyclical: are also repeating patterns but with no precise frequency, usually longer than two years. The cycle usually is combined with the trend (Hyndman and Athanasopoulos 2018).

Irregular components: refers to the residuals, noise or random component of the time series. The presence of these components makes it impossible to forecast perfectly (Hyndman and Athanasopoulos 2018).

Time series decomposition can be done in two ways. If the magnitude of the seasonal fluctuation is no different from the “expected value” of the “time series” then “additive decomposition” is appropriate. Otherwise if there is a sign of higher variations then multiplicative decomposition is suitable (Hyndman and Athanasopoulos 2018).

3.2 Stationarity

Stationarity is a fundamental concept in time series. It refers “to the presence or absence of a trend in the dataset.” “A stochastic process is considered to be ”stationary” if the mean, variance and covariance among the observations do not vary as time passes. . The “time series” is considered “non-stationary” if there exists variability in the dataset.

Since most real world data is non-stationary it needs to be processed and transformed before modelling. Differencing is the manner in which “non-stationary time series” data is transformed into a “stationary” state by taking the differences between successive observations. This difference eliminates the variability and creates a new “time series.” First-order differencing is usually sufficient to obtain a stationary state for non-seasonal data.

Diagnosis of non-stationary data is carried out with functions of autocorrelation. “Autocorrelation function (ACF) and Partial autocorrelation function (PACF)” are statistical correlation measures of linear relationship between observations in time series dataset. The correlograms obtained from this function can be visualized to identify order of auto an autoregressive (AR) model by using the PACF plot and order of moving average (MA) model using ACF plot (Hyndman and Athanasopoulos 2018).

In addition, there are several standard tests for detecting non-stationarity in time series. Among the popular ones include “Augmented Dickey-Fuller (ADF) (Dickey & Fuller, 1981) and Kwiatkowski, Phillips, Schmidt and Shin (KPSS) (Kwiatkowski et al. 1992).” These tests estimated the parameters of the “autoregressive process using the ordinary least squares (OLS) method and testing the hypothesis of the presence of a unit root.” In ADF the null hypothesis of “unit root” is tested against the alternative of level or trend stationarity, however in case of KPSS the hypothesis is vice versa.

3.3 Time series methods

For a long time now, forecasting agriculture prices has mostly been carried out using statistical and econometric methods. Most research studies devoted much of the effort in enhancing forecasting capabilities of these approaches. The most widely used methods are the “autoregressive (AR) model, moving average (MA).” These models are the foundation for more elaborate models like the “ARMA (autoregressive moving average), ARIMA (autoregressive integrated moving average) and SARIMA (seasonal autoregressive integrated moving average)” (Hyndman and Athanasopoulos 2018).

These are a class of linear models that utilizes “Box-Jenkins methodology” in the process of building models. Another set of statistical methods known as ETS (exponential smoothing) models are also common models that explicitly analyze error, trend and seasonality in time series data. However this thesis will only focus on autoregressive models as they are more appropriate for the subject of study.

3.3.1 Autoregression (AR) model

As the name already implies, autoregressive models state that the variable y_t depends linearly on the previous values of its own. This method is appropriate for “univariate time series without trend and seasonality” (Hamilton, 1994). The AR(p) can be expressed as follows:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + e_t \qquad (1)$$

In the process of AR the predicted value of y_t depends on its previous values, a constant bias parameter C as well as an error term $e_t$or residuals. The basic assumption is that “the residuals are a set of identically distributed, uncorrelated random variables with mean zero and constant variance $e_t \sim N(0, \sigma^2)$” (Brockwell & Davis, 2002). A stationary AR process shows a smooth decay in an ACF plot, however PACF can be utilized to figure out the nature of the process of AR since it shows the cutoff point after the lag p (Shumway and Stoffer 2017).

3.3.2 Moving average (MA) model

In the MA process the linear combination of weighted past forecast errors of the series is used instead of previous observations of the variable. In other words, the predicted values of y_t is a linear weighted average combination of past errors (Hamilton, 1994). The MA(q) can be expressed as follows:

$$y_t = C + e_t + \theta_1 e_t + \theta_2 e_t + \cdots + \theta_p e_{t-q} \qquad (2)$$

Here the e_t represents the errors or the white noise of the series and C is a constant parameter of the model. The order of an MA process can be spotted in the ACF plot, since it shows the cut-off after lag q denoting autocorrelation beyond this point is close to zero (Shumway and Stoffer 2017).

3.3.4 Autoregressive moving average (ARMA) model

"The ARMA model is a mixed model of both autoregressive AR(p) and moving average MA(q) terms." The model was first formulated by Peter Whittle (in 1951) and later adopted by (Box & Jenkins, 1990). The ARMA(p, q) model can be expressed as:

$$y_t = C + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + e_t + \theta_1 e_t + \cdots + \theta_p e_{t-q} \qquad (3)$$

Here, the original "time series" y_t is a combination of past observations that is linear in nature, past forecast errors as well as a series of random errors which follow the normal probability distribution. Although determining model order can be difficult from the ACF and PACF plots, we can rely on several criteria of information to look for the best-fit parameters for a "time-series data" that is given. Akaike information criterion (AIC) is one common way of measuring the relative quality of a statistical model. AIC penalizes the increased number of parameters within the modes in order to prevent overfitting (Bisgaard & Kulahci, 2011). Therefore the lower the AIC the better the model. It is written as:

$$AIC = 2(k) - 2\, log\, log\, (L) \qquad (4)$$

The AIC assumes the parameters k are estimated with the maximized likelihood L. Therefore, it balances between model complexity and the goodness of fit, since $-$ *log(L)* decreases with increasing likelihood L.

3.3.5 Autoregressive integrated moving average (ARIMA) model

The previous models assume stationarity of time series in their modelling. However, most real world data consists of variations in its innovation which makes it non stationary. As we have seen earlier one solution introduced by Box and Jenkins is to differentiate the original time series by subtracting a present observation from previous time step's observation so as to make it stationary. In ARIMA modelling, differencing is the first step and then the time series is integrated to provide modelling for the original non stationary data. In ARIMA(p, d, q) the d stands for the number of differencing needed to make the time series stationary. Therefore , it can be expressed as follows:

$$y_t' = C + \phi_1 y_{t-1}' + \cdots + \phi_p y_{p-1}' + \theta_1 e_{t-1} + \cdots + \theta_q e_{t-q} \qquad (5)$$

Where $y_t' = y_t - y_{t-1}$ indicating the differenced time series with both lagged errors and lagged values. When the number of differencing is determined then estimating the model parameters of (p, q) are similar to that of the AIC process described in the ARMA model (Hyndman and Athanasopoulos 2018).

3.3.6 Seasonal autoregressive integrated moving average (SARIMA) model

"SARIMA is an extension of ARIMA that supports direct modeling of the seasonal component of a time series." The seasonal model consist of AR and MA terms with $m{\cdot}1$, $m{\cdot}2$, . . . lags, where m denotes the frequency of seasonal patterns occurring at a certain period of time. "A SARIMA model is represented in the form of (p, d, q)×(P, D, Q)[m]." The second tuple indicates the order of the seasonal elements "where P = seasonal AR order, D = seasonal differencing, Q = seasonal MA order and the m in the bracket shows the corresponding seasonal lag" (Hyndman and Athanasopoulos 2018). "In the seasonal model S is the length of the seasonal pattern, B is the lag operator and e_t is white noise" (Cryer & McCarthy, 1986) (Lai, 1991) (Box & Jenkins, 1990). It can be presented as;

$$\Phi(B^s)\phi(B)(x_t - \mu) = \Theta(B^s)\theta(B)e_t \qquad (6)$$

$$\text{Seasonal AR: } \Phi(B^s) = 1 - \Phi_1 B^S - \cdots - \Phi_P B^{PS} \qquad (6.1)$$

$$\text{Seasonal MA: } \Theta(B^s) = 1 - \Theta_1 B^S - \cdots - \Theta_Q B^{QS} \qquad (6.2)$$

3.3.7 Box-Jenkins approach

The Box-Jenkins procedure is followed in determining the optimal parameters of the ARIMA and SARIMA models. In this approach, the model order is identified from ACF and PACF plots. The exponential decay of the ACF and a significant spike at lag *p*, but none beyond in the PACF suggest a model with p AR terms. Similarly, exponential decay of the PACF and a considerable increase at lag q in the ACF, but none after imply a MA model with q MA terms (Hyndman and Athanasopoulos 2018).

	AR(p)	MA (q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

Table 3.1: ACF and PACF for ARMA process

Afterwards the model parameters of the "ARIMA*(p,d, q)* and SARIMA (*p, d, q*)(*P, D, Q*)[*m*]" are estimated by fitting the model over identified order range and using AIC criteria of the maximum likelihood information. Since the determination of the best model order from the plots is difficult, repetition of the process with different values of *p* and *q* is done.. "The model with a minimum AIC is considered to be the best fitted model." Finally, residuals of the models are tested for white noise to ensure that there is no more information left to be incorporated into the model. Finally the model with the optimal parameters is used for prediction.

3.4 Machine learning approaches

At present various “machine learning (ML)” approaches have become popular and are applied in a wide variety of fields. Unlike statistical models ML models are capable of learning directly from a given dataset without explicitly being programmed. Although, This level of adaptability has proven successful, the ML model can be computationally challenging and require huge amounts of data to produce a significant result (Makridakis et al., 2018).

The classification of ML algorithms is done in two categories based on the manner in which they learn from data. Models that are supervised learn the functions that map the input data to the output data. This type of method requires samples of labeled data to work and it’s the focus of this thesis. The “time series” forecasting can be recategorized into a “supervised learning problem” which facilitates access to the several popular ML algorithms.” The second group of algorithms don’t need labeled data to predict output, instead it learns structures and patterns in the data set. This is called unsupervised learning (Owen P. Hall 2010).

In addition, supervised ML models are further classified based on the type problems they solve and output they generate. Classification models forecast or classify the distinct values between two or more classes and assign values to predefined groups. The output element is a discrete attribute. Regression models on the other hand map the function for the assignment of values to continuous output which is a constant type of real value (Owen P. Hall 2010). This section will describe the selected ML models for time series forecasting.

3.4.1 The K-nearest neighbor model (KNN)

"The K nearest neighbor (KNN) is a nonparametric method that predicts target output from a group of K samples based on a distance function" (Hastie et al., 2009). "Given a data point, KNN computes the Euclidean or Manhattan distance between that point and all data points in the training set. Then it searches for the closest K point from the training data and sets it as the average of the target output" (Thanh Noi and Kappas 2018). KNN model in the simplest way is written as:

$$\hat{y}(x) = \frac{1}{K}\sum_{x_i \in N(x)} y_i \qquad (7)$$

Here, $N(x)$ is a set of "K nearest neighbors" of training sample point x. Then $\hat{y}$ is computed from the closest K instances in the "training set" and the final predicted output is the average of the corresponding targets. "Euclidean distance" is the one of the most popular distance measures between x and y points:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (7.1)$$

The parameter K is very important and should be a value that balances between bias and variance of the model (Thanh Noi and Kappas 2018). The application of KNN can be made to solve both "classification and regression tasks," therefore the output is a mean value for the former and the most common class for the latter. Moreover, KNN learns complex concepts for local approximations using simple procedures at almost zero cost of learning. However some studies have pointed out that it may be limited by the storage requirement needed to create a database of historical data for searching k-nearest neighbors (Shee et al., 2014).

3.4.2 Support Vector Machines (SVM)

"Support vector machine (SVM) is a supervised learning model originally developed for classification problems." SVM makes use of a mapping which is non-linear to transform the original variables into "high-dimensional feature space and linearly separate the observation set." Thus, through non-linear classification the effective handling of complex problems is facilitated. In this case the line represents a nonlinear decision boundary in the new spaces known as hyper planes. “The margin between nearest data point and hyper-plane is maximized to decide on the right hyper-plane” (Witten et al., 2011) . “A partitioning of a linear hyper-plane corresponds to a nonlinear partition in the output space” (Kim, 2003). Mapping to a new space is done by a mathematical function known as Kernel. A kernel function is an inner product that replaces a scalar product with some choice of a kernel. Common kernel functions include linear, quadratic, polynomial as well as Gaussian radial basis function. Function K is a “kernel function,” that can be represented as:

Linear Kernel	$K(x_i, y_i) = x_i^T . x_j$ (8)
Polynomial Kernel	$K(x_i, y_i) = (1 + x_i^T . x_j)^d$ (9)
Radial Basis Function (RBF) Kernel	$K(x_i, y_i) = e^{-\lambda \lVert x_i - y_i \rVert}, \lambda = \frac{1}{2\sigma^2}$, $\sigma = K\ bandwth$ (10)

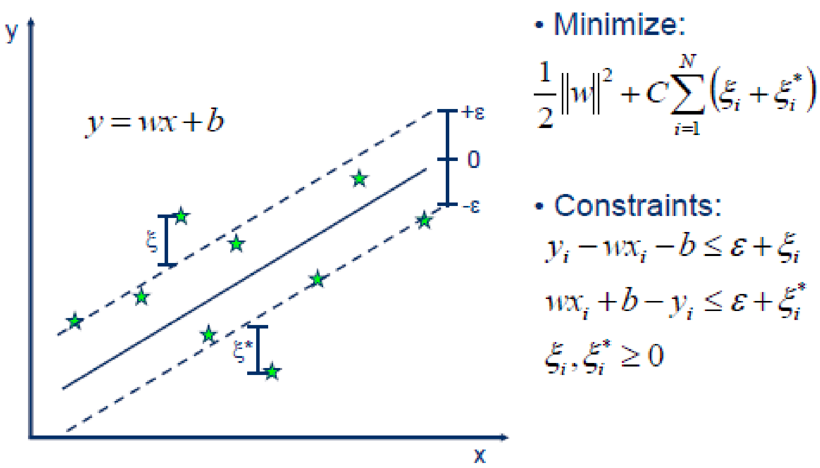
Table 3.2: SVR kernels

SVM is also deployed to solve regression tasks while maintaining all the main characteristics of the algorithm known as Support Vector Regression (SVR). Since it’s a regression problem the output is a real number and this makes it challenging to assume the value at hand, which has numerous possibilities. “Instead of finding a maximum-margin hyperplane to separate the data points, the SVR model aims to find a hyperplane that minimizes the distances to the data points.” Although the algorithm is complicated the main objective is error minimization and margin maximization within an error range that is tolerable. This can be expressed as a “convex optimization problem” (Bishop 2006).

$$\frac{1}{2} \lVert w \rVert^2 \quad \text{in bound of} \quad y_i - w_1 . x_i - b \leq \varepsilon \qquad (11)$$

$$w_1 . x_i + b - y_i \leq \varepsilon \qquad (11.1)$$

Where X is a multivariate set of m observations with observed outcome values Y. The errors cost is tolerated if they are within the band and the minimization can be mathematically defined as in the figure below. Where w is flatness, C is a constant and ξ_i and ξ_i^* are the slack variables (Bishop 2006).



Source: www.saedsayad.com

Figure 3.1: Support Vector Regression

One advantage of SVR over other models is that it looks for ways to work towards the minimization of an "upper bound" of generalized errors since it established on "the structural risk minimization principle." This makes it resistant to the problem of overfitting. In addition, it's a "linearly constrained quadratic program which means the solution of SVR is always globally optimal" (Awad & Khanna, 2015).

3.4.3 Artificial Neural Networks (ANN)

"Artificial neural networks (ANN) are adaptive computational models inspired by the inner workings of the human brain." ANNs are made up of interconnected web of neurons, also known as nodes or Perceptron. Neurons are computational units which pass and process information sequentially between layers (Ripley 1996). ANN is constructed with three layers: the input layer includes all the input variables known as input neurons and similarly an output layer that contains output variable or output neurons. The layer between them is called a hidden layer and ANNs can have multiple numbers of hidden layers in relation to the complexity of the task (Ripley 1996).

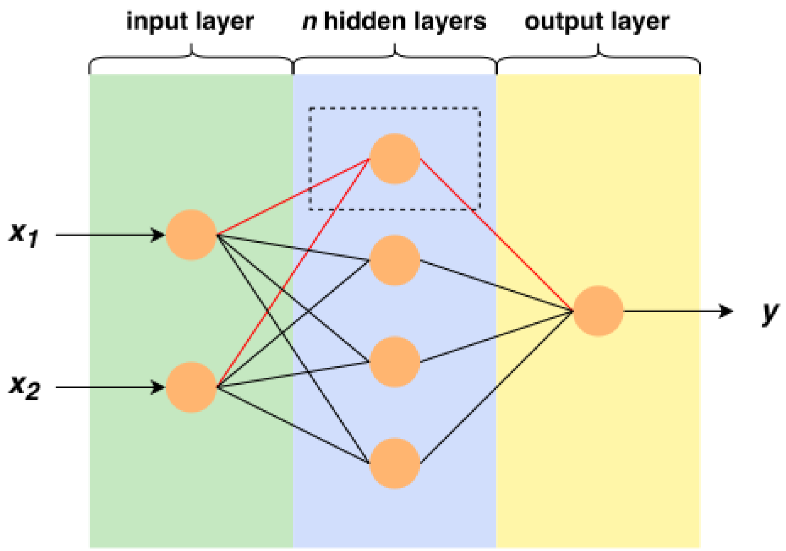
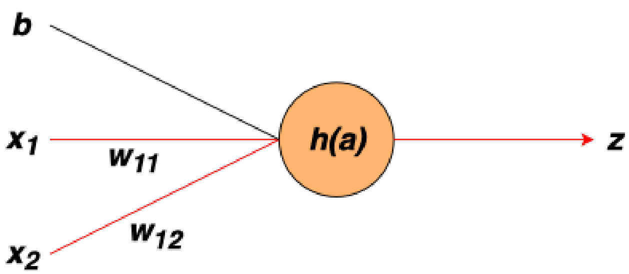


Figure 3.2: Single-hidden-layer network

In ANN, each of the neurons contains an activation function and has a connection to all "neurons" in the following layer. The neuron's output is a non-linear combination of inputs, certain weights of those inputs and an application function on the result. Weights are assigned from previous experience obtained through learning along with current states. Each neuron (a) in the "hidden layer" multiplies the inputs (x_1, x_2...) with their respective weights (w_1, w_2...) which represents the importance of the incoming signal (Gupta et al., 2005). It then sums up and passes the results to an activation function (h) and processes the output (z). "Neurons also include a bias (b) whose value is constant and equal to 1" (Dreyfus, 2005).

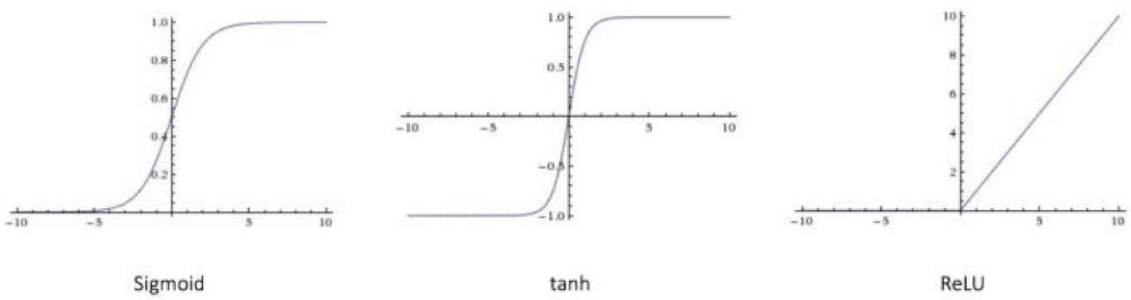


$$a = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b \qquad (12)$$

$$z = h(a) \qquad (12.1)$$

Figure 3.3: Single ANN neuron

"An activation function is mathematical equations that determine the output of a neural network model." They affect the convergence ability of the model and help to normalize the output of any input in the range between 1 to -1 or 0 to 1. The function can be linear or non-linear, the most common are normal linear, sigmoid, hyperbolic tahn and ReLu (Bishop 2006).



$f(x) = \frac{1}{1+ e^{-x}}$ $\qquad$ $f(x) = \frac{2}{1+ e^{-2x}} - 1$ $\qquad$ $f(x) = max\ (0, x)$

Figure 3.4: Activation functions

The topology or architecture of neural networks varies depending on the given problem. This architecture refers to the synaptic connections between neurons and layers. The Feed-forward neural networks (FFNN) or perceptron is a design architecture in which the signal flows unidirectionally among neurons from input to output layer. The simplest type of FFNN contains one layer associated with a group of input terminals, but if a network contains more than one layer of artificial neurons it's called multi-Layer Perceptron (MLP).

MLPs contain multiple hidden layers and make the networks computationally stronger. However there is no direct contact with an external layer and no feedback connections to the output layers. MLP can be mathematically expressed as follows:

$$y_t = \alpha_0 + \sum_{j=1}^{q} \alpha_j g\left(\beta_{0j} + \sum_{i=1}^{p} \beta_{ij} y_{t-i}\right) + e_t \qquad (13)$$

Here, " y_t is the output, y_{t-i} (i = 1,2, ⋯ , p) are the p inputs to the network, p and q are the number of input nodes and hidden nodes respectively, α_j (j = 0,1,2, ⋯ , q) and β_{ij} (i = 0,1,2, ⋯ , p; j = 0,1,2, ⋯ , q) are weights to the connection and $e_t$is the random shock. α_0 and β_{0j} are the biased terms."

The main learning process is called Error Backpropagation or Backward propagation (Rumelhart et al., 1986). It's an optimization method which adjusts the randomly assigned weights at the initialization of the network using a gradient descent. "This algorithm works in two phases: first, the input signal propagates through the network in a forward direction with the weights being fixed; then the error is propagated backwards from the output layer to the input layer. The weights are then adjusted based on the error-correction rule (e.g. sum squared error)" (Tang et al., 2007).

"Time-Delay Neural Network (TDNN) is another form of neural network architecture with a single hidden layer structured for time series prediction." The model follows the same learning process as FFNN but uses predicted values recursively as inputs to forecast future values. In their paper (Jha & Sinha, 2013) expressed the model as follows:

$$y_{t+1} = g\left[\sum_{j=0}^{q} \alpha_j f\left(\sum_{i=0}^{p} B_{ij} y_{t-i}\right)\right] \quad (14)$$

Here, "p is the number of input nodes and q is the number of hidden nodes. B is the connection weight attached to the input node(i) and hidden layer(j). similarly, α is the weight attached to the hidden layer and output node. g is an activation function at the output layer and f at the hidden layer." The i^{th} input of the lagged values is represented by y_{t-i}.This type of network is suitable for a univariate time series forecasting using historical data as inputs and maps the below function to predict of future values as output.

$$y_{t+1} = f\left(y + y_{t-1} + y_{t-2} \dots y_{t-p+1}, w\right) + e_{t+1} \quad (15)$$

Where " y_t is the forecasted output and f is a function formed by the structure of network and connection weights, $y_{t-1} + y_{t-2} \dots y_{t-p}$ are past values, w is a vector of weights and e_{t+1} are the residuals." TDNN functions as a non-linear autoregressive model to reduce the forecasting error by making adjustments in the weights at each time step.

ANNs are capable of solving linear and nonlinear problems, they can also handle complex tasks such as “function approximation, classification, and pattern recognition as well as optimization problems.” Moreover, ANNs do not need to be re-programmed and they tend to be fault tolerant due to the highly parallel nature of the network. Therefore, if a component of an ANN fails, the net continues to operate (Ghosh-Dastidar & Adeli, 2009).

3.4.4 Cross Validation

“Cross validation (CV)” is a resampling method used to optimize hyperparameters of machine learning models in order to prevent them from overfitting. ML models consist of different parameters that cannot be optimally estimated using a direct procedure, CV is commonly used to tune and estimate model parameters (Berrar 2019). For example the optimal number of *K* in KNN as well as the epsilon(sigma) and cost parameters in SVM. In this case CV is applied to different parameter values and the parameter that minimizes the CV error is then selected for building the final model. The most commonly used data resampling method is the “k-fold cross validation.” Here the training dataset available is randomly divided into equal sized *k* subsets. This study used the 10-fold cross-validation recommended by most studies (Berrar 2019), in which the model is trained on the “9 subsets and tests” on the remaining subset. This process is repeated until each *k* subsets serve as a validation set. Finally the average performance for each *k* validation subset is measured to get the overall CV performance.

Chapter 4: Data

Commodity prices vary by county, region and volume in which the transaction takes place. Since this study focuses on determining the appropriate forecasting model for selected countries (Ethiopia, Kenya and Sudan) in SSA regions, the scope is limited to analyzing the prices of the most essential staple crops (Wheat, Maize and Sorghum) from each country.

The crop prices were obtained by selecting the major commodity markets in those countries which happened to be located in their capital cities (Addis Ababa, Nairobi, Khartoum). Since, the majority of transactions and price determination occurs in these markets, it makes it suitable to study dynamic behavior of crop prices. Finally to enable data consistency, price data were sourced from "GIEWS FPMA Tool for monitoring and analysis of food prices setup by Food and Agriculture Organization of the United Nations (FAO)." This study used the latest and most available information from this database.

Figure 4.1: Monthly whole sale prices of Wheat(left), Sorghum(right) and Maize(bottom)

Monthly wholesale prices of the most significant crops of Wheat, Sorghum and Maize from SSA countries of Ethiopia, Sudan and Kenya were selected. Therefore, the price data was obtained from the major commodity markets of Addis Ababa, Khartoum and Nairobi respectively. In addition, crop prices are denoted in terms of per USD kilograms to ensure data consistency. The price series on Wheat covered a total of 253 months (January 2000 to January 2021), for Sorghum its 229 months (January 2001 to January 2020) and for Maize 178 months (January 2006 to January 2021). *Fig(4.1)* demonstrates the variations among the agricultural price data spanning for around 20 years.

4.1 Commodity Description

Wheat is the fourth largest produced crop in Ethiopia and accounts for second largest consumption. The country commercially imports wheat and most of the humanitarian food assistance is in the form of wheat. Timing and distribution of imports affect prices in addition to different macroeconomics factors such as population growth, high money supply and preference to wheat based products (FAO, 2002). Wheat prices from the Addis Ababa market exhibits an upward trend with several noticeable fluctuations. Price spiked around 2009 following the global economics crisis and recently it marginally dipped in March of 2019 but recovered the following year.

Sorghum dominates food production and it is a stable food with 96% of household consumption (Elzaki, Yunus Sisman, and Al-Mahish 2021). Price levels are influenced mostly by supply shortfall due to hostile weather, increasing fuel prices, civil war and conflicts (Elzaki, Yunus Sisman, and Al-Mahish 2021). Sorghum prices from the Khartoum market are highly volatile with multiple price shocks occurring within short interval periods. In addition, the price shows no period of substantial stability and high sensitivity.

Maize is the most important commodity in Kenya and makes-up about 65% household consumption. The majority maize is produced by small rural farmers, thus favorable weather conditions and improved farming methods determines crop supply. In addition, government stockholding and trade policies also affect wholesale maize market prices (Jayne et al., 2008) Maize prices from the Nairobi market experienced a major shock around 2010 which caused the price to spiral downwards to its lowest point. The price displayed patterned fluctuations to both extremes. A sharp increase occurred around 2017 and eventually declined to lower levels before recovering around the mean by 2020.

4.2 Descriptive statistics

Summary statistics from *Table(4.1)* shows a comparable mean value for the commodities, however we see large standard deviation for wheat and sorghum prices. Wheat prices show a symmetrical distribution while sorghum and maize prices report a positive skewness. Negative excess kurtosis indicates that the distribution of the commodity prices has a lower wider peak and thinner tails. Moreover the (Jarque & Bera, 1980) (JB) test rejects the normality of wheat and Sorghum prices at 1% and 5% level of significance, while it shows the maize prices as normally distributed.

	Mean	**SD**	**Var**	**Min**	**Max**	**Median**	**Skewness**	**Kurtosis**	**JB**
Wheat	0.372	0.137	0.018	0.12	0.71	0.38	0.000467	0.879364	7.84 (0.01982)
Sorghum	0.381	0.158	0.025	0.13	0.79	0.41	0.204	-0.799	7.43 (0.02425)
Maize	0.323	0.076	0.005	0.17	0.54	0.33	0.026	-0.417	1.1612 (0.5595)

Table 4.1: Summary statistics

Chapter 5. Methodology

This section covers the forecasting framework employed to run and evaluate the methods described in Chapter 3. It presents a thorough analysis of hyperparameter selection and the optimization process of each forecasting method. Later, the prediction output of the optimized models along with various model specific forecasting strategies is examined.

5.1 Forecasting framework

The aim of this forecasting framework is to enable a similar testing structure for each forecasting method for a given dataset. The process begins by reading the data and preprocessing it for completeness. Scaling the data is also carried out at this stage. Afterwards, the data is split into "training and testing sets." The training data set along with other adjusted parameters helps in training the models. Based on the estimates of the model, the "test set" then predicts values for a given forecast method. Finally, prediction results are saved and the evaluation of the performance of each method is done by comparing the forecasted values with the original values.

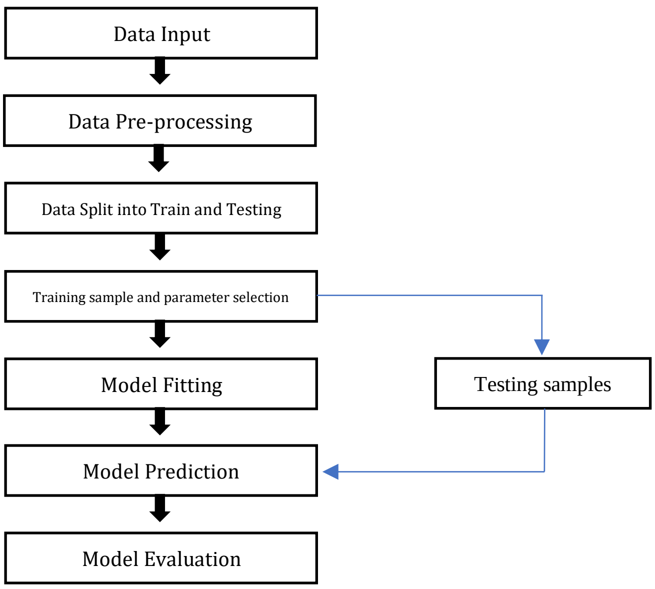


Figure 5.1: Forecasting framework

5.2 Data

As mentioned in the previous chapter, the dataset is a monthly wholesale price of the most significant crops of Wheat, Sorghum and Maize from SSA countries of Ethiopia, Sudan and Kenya respectively. The price data was obtained from the major commodity markets of Addis Ababa, Khartoum and Nairobi. In addition, crop prices are denoted in terms of per USD kilograms to ensure data consistency. The price series on Wheat covered a total of 253 months (January 2000 to January 2021), for Sorghum its 229 months (January 2001 to January 2020) and for Maize 178 months (January 2006 to January 2021).

5.3 Data Preprocessing

Preprocessing of data is applied to detect such outliers and missing values within the time series that might generate noise and inconsistency. Handling such discrepancies improves the quality of data and prediction results. Therefore missing values were removed and our data shows no significant outliers.

$$z = \frac{x-\mu}{\sigma} \qquad (16)$$

Furthermore, preprocessing transforms the time series data into flat distribution through scaling. The above formula also known as the "z-score, re-scales the data by first subtracting the mean (μ) from each data point (x) and then dividing it by its standard deviation(σ)." This converts the data so as to have a zero mean and a unit variance. Normalizing data before modelling greatly enhanced ML models learning process in pattern recognition (G. Zhang et al., 1998).

5.4 Data Partitioning

The input dataset is categorized into "training and testing sets" before any modeling is carried out. The training set also known as "in-sample data" is used during the model building process and allows learning patterns from the data. After the model completed training, the testing set is used to generate forecasting using the estimation generated by the model with unseen data sets also known as "out-of-sample data" by the mode during the training phase. Therefore performance evaluation of a model is done by comparing the forecasted outcome with the original values of this test data. "Typically, around 70% of the sample data is used as the training set while the remaining around 30% is used as the testing set" (Hyndman & Athanasopoulos, 2018). *Table 5.1* lists the training and testing split of the observations for each commodity

	Total samples	**Training samples**	**Testing samples**
Wheat	253	192	61
Sorghum	229	180	49
Maize	181	120	61

Table 5.1: Training and Testing data partition

5.5 Training and Prediction

At this stage, each model sets optimal parameters values using some form of optimization procedures and generates estimates from its unique fit function on the training dataset. Given our univariate time series as our training data, the original price values are the outcome variable and their lagged versions are the explanatory variables also known as features. Features are one of the key factors that have an impact on the quality of prediction. Since, we are working with a relatively small monthly dataset this study made use of only the 12th lag as an optimal feature. Moreover, the use of multiple lag values as features in the training stage causes the models to overfit and lack generalization.

After the training phase, the forecasting function of each method applies the optimal parameters values in combination with various forecasting strategies and uses the testing dataset for prediction. *Fig 5.2* summarizes the modelling flowchart of each approach and each section is discussed in the following subsection. However, the performance and evaluation results will be discussed in the next chapter.

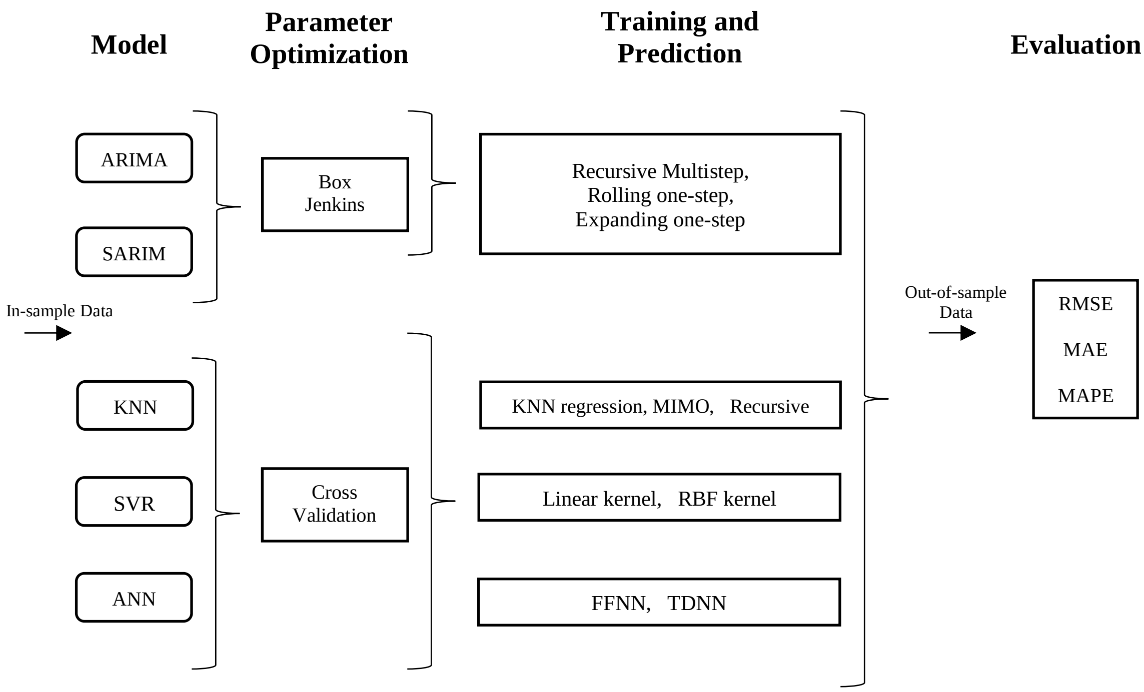


Figure 5.2: Modelling Flowchart

5.5.1 Time series Decomposition and Stationarity

Before building a model, first the training data of the commodity prices were decomposed into "trend, seasonal and cyclical components" using the Classical Seasonal Decomposition by moving averages. Since the price data of each commodity exhibits a non-linear series, we applied the multiplicative decomposition model in order to best capture each of these components in a given model. It is clear that the trend component for Wheat and Sorghum prices is increasing over time, indicating non-stationarity of the data but for maize prices that trend seems to be stable after a sharp decline in 2011 *(See appendix, Figure A1.1).*

Next, we plot the "ACF and PACF" for the regular time series *Figure (5.3).* Wheat and Sorghum prices show a persistent correlation between the data points indicating the price data is non-stationary. Conversely, Maize prices seem to exponentially decay beyond lag 15 indicating weaker correlation between the lagged values of the series. But, after taking the first difference the plots for all commodities indicated stationarity of the series *(See appendix, Figure A1.2).*

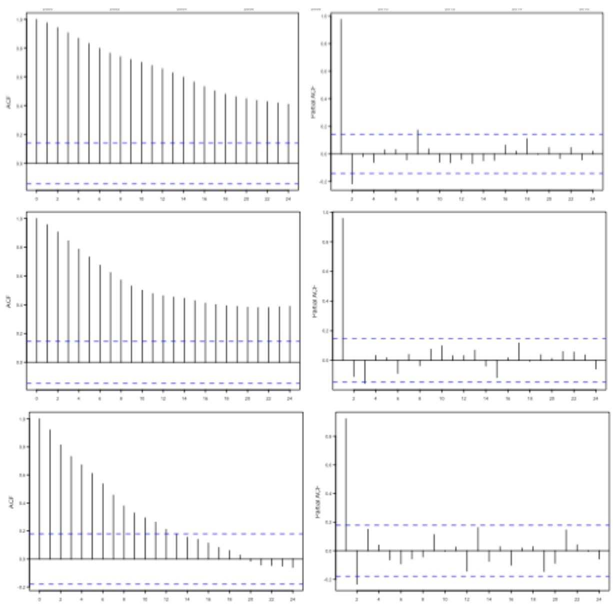


Figure 5.3: ACF and PACF level plots of Wheat (top), Sorghum (middle) and Maize (bottom)

In addition, we applied the ADF and KPSS to formally test for stationarity on the level time series *Table(5.2)*. Wheat and Maize prices show that their "time series is non-stationary." Subsequently, we take the first differences and the series becomes stationary, indicating that the price data is an integrated order of one.

		ADF		KPSS	
		t-static	p-value	t-static	p-value
Wheat	Level	-2.9269	0.1883	2.695	0.01
	Diff	-5.3695	0.01	0.039417	0.1
Sorghum	Level	-4.2904	0.01	2.4473	0.01
	Diff	-5.2769	0.01	0.031405	0.1
Maize	Level	-2.5144	0.3629	1.0537	0.01
	Diff	-4.7341	0.01	0.068163	0.1

Table 5.2: ADF and KPSS stationarity test

However, In case of the Sorghum prices, both tests gave contradictory results on the level and the differenced time series. Since, "ADF and KPSS" are designed to detect the presence of a "unit root" to test for non-stationarity, the presence of other forms of non-stationarity (seasonal) and trend stationarity might be the cause of the inconsistent results. Therefore, we decide to build two (S)ARIMA models, with and without differencing the time series. Then, the model with lowest training error will be selected for forecasting.

5.5.2 Fitting the optimal (S)ARIMA

Now that the data is ready and satisfies all the assumptions of modeling, we can determine the order of the models to be fitted into the data. To achieve this, the "*auto.arima()* function is used to find the appropriate values of the AR-component p and the MA-component q (S)ARIMA models." The function uses the stepwise algorithm to automate the process of searching and selecting possible "AR, MA and seasonal orders" within the restrictions provided and choose the parameters that minimize the AIC.

The function fit the models on the training data set by running twice, for ARIMA (without considering seasonality) and SARIMA (taking seasonality into account) models for each commodity. The function also included the non-stationary character of the univariate series by allowing for a first difference *(d = 1)* in the modelling process. As mentioned earlier, since the stationarity status of the Sorghum price series is not clear from the ADF and KPSS tests, the models are fit one more time using the level time series. The results listed on *Table(5.3)* had been suggested as the most appropriate and optimal models for the price data of each commodity.

	ARIMA	**SARIMA**
Wheat	ARIMA(0,1,1) AIC = -868.45 RMSE = 0.0245	SARIMA(0,1,1)(0,0,1)[12] AIC = -868.43 RMSE = 0.0244
Maize	ARIMA(1,1,2) AIC = -500.99 RMSE = 0.0283	SARIMA(1,1,2)(0,0,1)[12] AIC = -502.13 RMSE = 0.0279
Sorghum ***d = 1***	ARIMA(0,1,0) AIC = -603.94 RMSE = 0.0444	SARIMA(0,1,0)(0,0,0)[12] AIC = -603.94 RMSE = 0.0444
Sorghum ***d = 0***	ARIMA(1,0,2) AIC = -608.31 RMSE = 0.0431	SARIMA(1,0,2)(0,0,0)[12] AIC = –608.31 RMSE = 0.0431

Table 5.3: Parameter optimization of ARIMA and SARIMA models

The optimal orders for each commodity time series are listed along with the AIC as well as the "level of error (RMSE (root mean squared error)) on the training data set." Wheat prices show no AR terms and only have one MA term in the ARIMA model; this is similar to the SARIMA model but with one additional MA term for the monthly seasonal order. Maize prices on the other hand hold one AR and two MA components. It also contains one more MA term in the seasonal model. The optimal model order for Wheat and Maize prices is an integrated order of one, and the series shows stronger dependency on the linear combination of previous errors than its lagged values.

In case of Sorghum prices, the optimal ARIMA model on the differenced series contains no AR or MA terms. The ARIMA(0,1,0) represents a random walk model. This implies that prices are independent of each other and that past innovation cannot predict future movements. Moreover, it indicates an infinite slow mean reversion process of the series. However modeling the prices without differencing the data results in an ARIMA model with one AR component and two MA components. The findings also indicate that the seasonal model is not applicable for this particular price data. For this study the latter model is preferred for the forecasting task, since it shows less errors on the training data set with lower model complexity indicated by the

AIC and RMSE(root mean squared error) respectively. Furthermore, the lack of AR and MA components in the first model tells us little about the price dynamics of the given price data.

Moreover, the diagnostic plots of the ARIMA and SARIMA models show that the residuals are less correlated with its lagged version, indicating the existence of a white noise. Thus, the selected models have the most fitted model parameter values. This is confirmed as well by the Ljung-Box statistics p-values, showing that the residuals are independent of each other *(See Appendix, Figure A1.3 - A1.5)*. Especially, the residual results of Sorghum prices show a strong favor to the model without differencing as the p-values are more robust than the random walk model.

5.5.3 Forecasting with the (S)ARIMA models

After determining the optimal parameters of (S)ARIMA models, the next stage is forecasting the univariate commodity prices for a given forecast horizon. Since, the temporal structure of time series problems makes multi-step forecasting more complicated, we used the following three forecasting strategies for both the ARIMA and SARIMA models: 1. Recursive multi-step ahead forecast 2. Expanding window one-step ahead forecast 3. Rolling window one-step ahead forecast

The “recursive multi-step” strategy iteratively adds the predicted values of one-step forecast in the model in order to forecast the next time step. This method forecasts multi-periods ahead until the given forecast horizon, which in this study is the length of our “testing data set.” The expanding window one-step ahead forecast uses all the available “time series data” in order to predict the next observation. This method grows the training window as more data becomes accessible. The rolling window one-step ahead forecast is much like the expanding window, the difference is that it uses the most recent observations by having a fixed training window (10 year monthly data). The window then slides by one time period, in our case, observations are shifted by one month after every forecast. This method uses recent lag values of the time series for prediction. *Figure (5.4)* Shows the output of the different forecasting strategies for each price commodity. Performance of each strategy will be evaluated and discussed in detail in the next chapter.

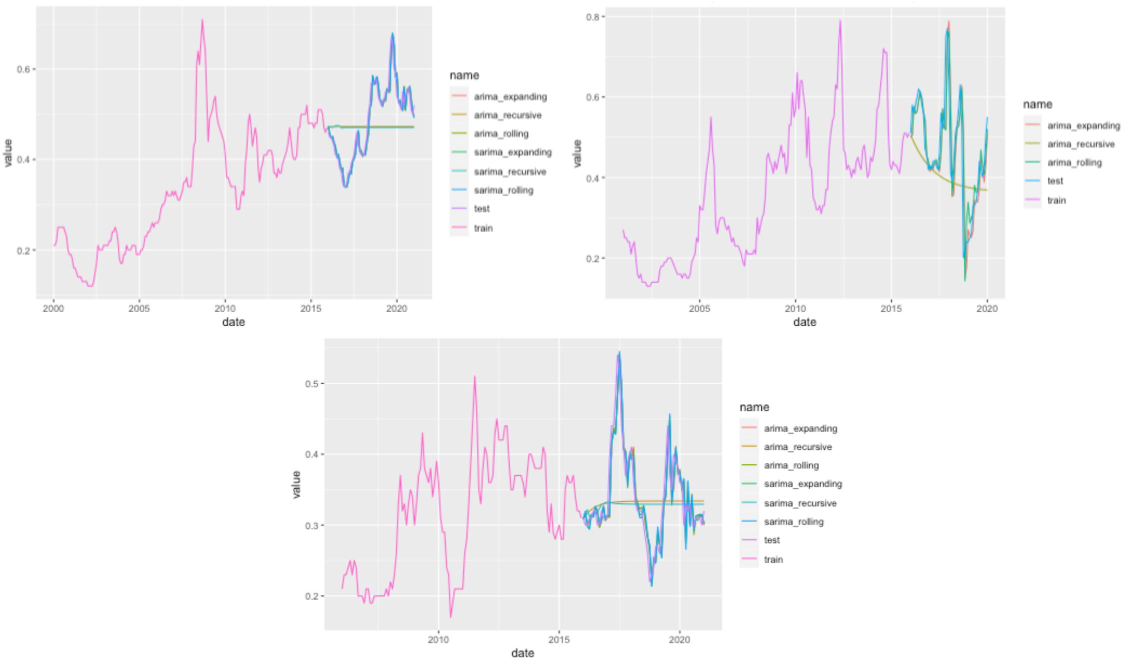


Figure 5.4: Forecasting with different strategies Wheat (right), Sorghum (left) and Maize (bottom)

5.5.4 K-nearest neighbors (KNN)

"The KNN algorithm" searches for the k-closest samples on the training dataset using a Euclidean distance calculation, assuming that the prediction of new values are similar to the prediction of the other nearby instances. This makes the size of *k* as a vital parameter for tuning the model performance. In setting the initial value of k, the rule of thumb is to take the "square root of the total size of the training dataset." However, in reality the value of k is heavily influenced by the quality of a given dataset. As a result it's difficult to determine the size of k beforehand. "The optimal value of k for the model has been found, by using the 10-fold cross validation resampling method." The working procedure of this validation method has briefly been presented in section 3 of this study.

To train the KNN model we used the *"knn"* regression function in the *caret* library and regressed the training data set on its 12th lag, since we are using monthly price data. During the procedure, the KNN regressor iterates over a rational number of k in a range (1:24) and use the "10-fold cross- validation" to estimate the optimal values of k and measure the output accuracy based on RMSE. This accuracy is measured between true values and the predicted values of each subset of the fold.

This process is implemented for each commodity time series, and the CV model is optimized according to size of k. Therefore, the value of "k with the lowest RMSE score is selected as the optimal model parameter." Within the range from 1 to 24 (over the range of 24 months), we have found the optimal k value for Wheat at k = 20, Sorghum k = 10 and Maize k = 24. The results are presented visually in *(See Appendix, Figure A1.6).* Afterwards we used the CV optimized *"knn"* model on the testing data set for prediction. The model performance for each commodity is displayed on *(See Appendix, Figure A1.7).*

In addition we implemented two more forecasting strategies with the KNN model; "Multiple Input Multiple Output (MIMO) and Recursive strategy using the *tsfknn* package." MIMO strategy presents "a multiple-output approach to multi-step-ahead forecasts." Improving the forecast accuracy by maintaining the stochastic dependencies between predicted values is its main aim (Ben Taieb et al., 2012). This KNN regression process consists of lags, instance, features, and targets parameters. In our case, because the time series data is monthly, the lagged values are set from 1:12. The first twelve lagged values are the features of the time series that build or describe the instance vector, which is shown by purple points on *Figure (5.5)*. The closest or nearest neighbor features to the instance vector are then found based on the Euclidean distance. The time series that comes after the nearest neighbors are the target values. "The length of the target vector is equal to the size of the forecast horizon." The optimal size of *k* from the previous stage is again used to average the targets of the nearest neighbors to forecast based on a given forecast horizon.

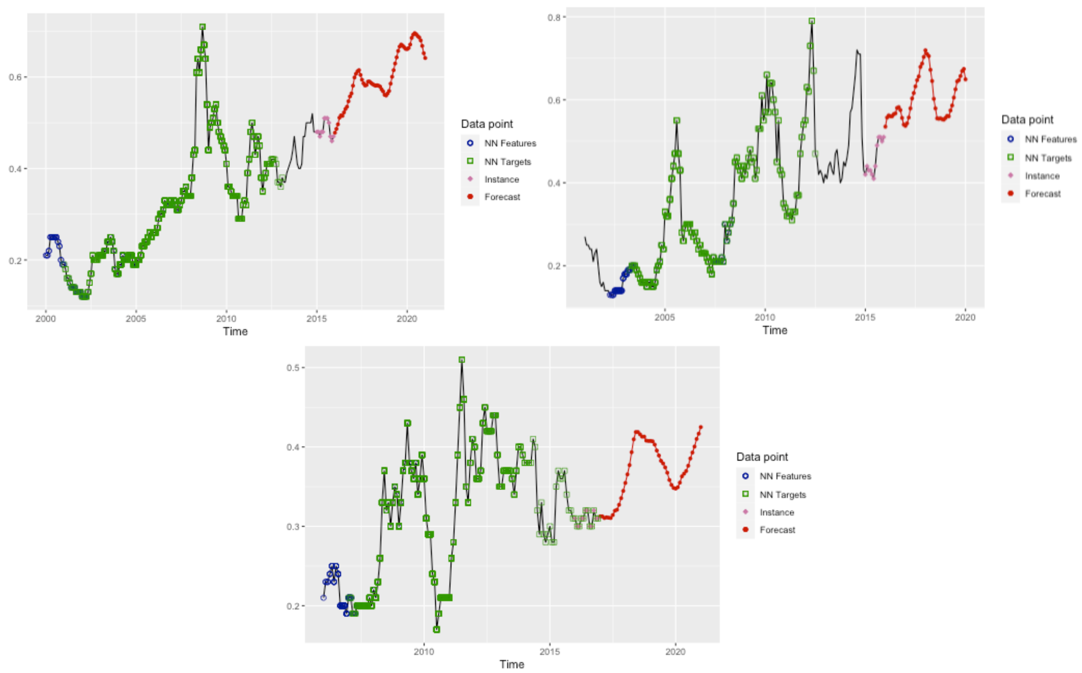


Figure 5.5: MIMO strategy to forecast prices of Wheat(left), Sorghum(right) and Maize(bottom)

The recursive strategy is similar to the iterative approach used by the statistical models for one-step a head prediction. In case of recursive KNN, the target vector of the training instance only contains one value. The model is then applied to forecast future periods based on the 12 lagged observations and using optimal k values. The predicted values are then used as features of the new instance to predict the next values. Forecast output of MIMO and recursive strategies for each commodity is shown in *(See Appendix, Figure A1.8 and A1.9).*

5.5.5 Support vector regression (SVR)

As presented earlier, SVR is a form of SVM build to solve regression tasks and it consists of three hyper-parameters that determine its performance. The penalty parameter C (punishes errors beyond a given value), the insensitive parameter ε (controls the width of the insensitive zone) and the kernel function parameter σ^2 (affects model performance). These parameters greatly influence the accuracy and the complexity of the SVR model.

The parameter C is defined by the user at the initialization of the model. Therefore in order to find the "optimal value" of C, we have used the SVR model to regress the training dataset on its 12^{th} lag with different values of the C parameter in range (0.25 to 128 on a 0.25 interval) and conduct a "10-fold cross-validation" for each value of C. This study applied "Radial Basis Function (RBF) ("*svmRadial*") and Linear ("*svmLinear*") kernel" from the *caret* library for this task, since both kernels are cable of approximating any distribution in the feature space and offers an improved prediction performance (Bishop 2006). However, for linear kernel the value of C is held constant at the value of 1. The polynomial kernel function is omitted due to its computational complexity and high computational requirement than our current machine.

Table (5.4) lists the optimal parameters values obtained from the cross validation regression on the training set. In general the wider the tube or the bigger the value of C the more support vectors are used in the estimation process, which in turn increases model complexity. As a result values of C tends to be selected from smaller denominations (Bishop 2006). These optimal values are used for prediction on the test set and the output the SVR model for each commodity is shown graphically on *(See Appendix, Figure A1.10 and A1.11.)*

	RBF kernel	
	C	sigma
Wheat	0.5	9.50
Sorghum	0.5	32.42
Maize	0.25	3.63

Table 5.4: Hyperparameter optimization of SVR for each commodity

5.5.6 Artificial Neural Network (ANN)

In a feed-forward artificial neural network (FFNN) the information or signals flow unidirectionally "from the input nodes through the hidden nodes and to the output nodes." The input data is first scaled within a given range, this pre-processing speeds up learning and leads to faster convergence of the neural network. The optimal weights assigned with each input are learned through the process of repetitions and are enhanced along with current states. The ANN model is trained using a "gradient descent optimizer, and sigmoid function" is used to activate the neurons in the ANN. (Goodfellow et al., 2016) pointed out that sigmoid activation vacation is suitable since the logarithm of the model's output needs to be suitable for gradient-based optimization of the log-likelihood of the training data." The prediction performance of the ANN model depends on several important "hyperparameters including the number of hidden layers, number of times the ANN is trained (number of iterations), the maximal number of neurons in the hidden layer (nodes within each layer) and the weight decay (weight regularization value to avoid overfitting)."

According to several studies (Lolli et al., 2017) (Gad & Jarmouni, 2020) the use of a single hidden layer in forecasting time series results in improved performance and is more appropriate while working with small datasets. Moreover, it requires less computational capacity and reduces model complexity. Therefore, in order to find the optimal values of these parameters, we set up a neural network with a single hidden layer using the scaled training data as input and its 12th lag series as the target values. Next, a grid search is set-up and prearranged with different size of neurons ranging from (1:20) and decay level ranging (0.5 : 1e-7) while holding the maximum number of training to a 1000 rounds.

Afterwards, we applied the *"nnet"* library together with the *"train"* function from the *caret* package and trained the ANN model using "10-fold cross validation" for each size of the neuron and level of weight decay. The performance of each pair is measured by RMSE within the testing part of the cross validation. Finally the pair with lowest RMSE score is selected as the optimal parameters listed on *Table (5.5)* and used for prediction with the out of sample data. Performance of the ANN on the testing dataset for each commodity is presented on *(See Appendix, Figure A1.12).*

	Hidden nodes	**Weight decay**
Wheat	3	0.001
Sorghum	2	1e-05
Maize	2	1e-05

Table 5.5: Hyperparameter optimization of FFNN for each commodity

The next ANN model applied in this study is the "TDNN (Time-Delay Neural Network)." TDNN is a type of "feed-forward neural network" with a single hidden layer and where each observation of the lagged time series serves as an individual input node for univariate forecasting. The *"nnetar"* function in the forecast package fits the TDNN model to our time series data, constructing a non-linear autoregressive model in which Multi-step forecasts are computed recursively.

To implement the model, first the appropriate number of hidden neurons for time series forecasting is determined using a recommended approach by most literatures (Hagan et al. 2014). Using the formula below provides a fitting number of nodes that are balanced between the input and out nodes as well as prevent the model from overfitting.

$$N_h = \frac{Ns}{a*(Ni+No)} \qquad (17)$$

Here, Ns the size of the training samples, Ni and No are the number of input and output nodes respectively. The a denotes an arbitrary scaling factor. It's the effective branching factor or number of nonzero weights for each neuron. In addition, the data is transformed using Box-Cox transformation (G. E. P. Box and Cox 1964) into a normally distributed shape and to ensure that residuals are homoscedastic.

Similarly, the model is trained using gradient descent optimizer, logistic sigmoid as an activation function, and with training simulations of 1000 rounds, each with random starting weights. Although the two neural networks share a similar training structure, in case of TDNN, the output values from each training round is added to the training samples and train the model for one-step forecasting. This is different from the previous model, in which the estimated weight parameters of lagged values are obtained through cross-validations.

TDNN model fits the transformed lagged inputs from the training set along with the optimal hidden nodes and forecast recursively for multiple periods until the given forecast horizon. In our case, the size of the test dataset is the same as the forecast horizon. “The output of the neural network autoregressive (NNAR) model is denoted as *NNAR(p,k)* (non-seasonal data) and *NNAR(p,P,k)[m]* (seasonal data) where, k is the number of hidden nodes.” These models are similar to the *AR(p)* and *SARIMA(p,0,0)(P,0,0)[m]* but with nonlinear functions. The results of the NNAR for a five year forecast horizon for each commodity is shown below in *Figure (5.6)*. Illustration of the predicted output and actual values side by side is shown in *(See Appendix, Figure A1.13)*.

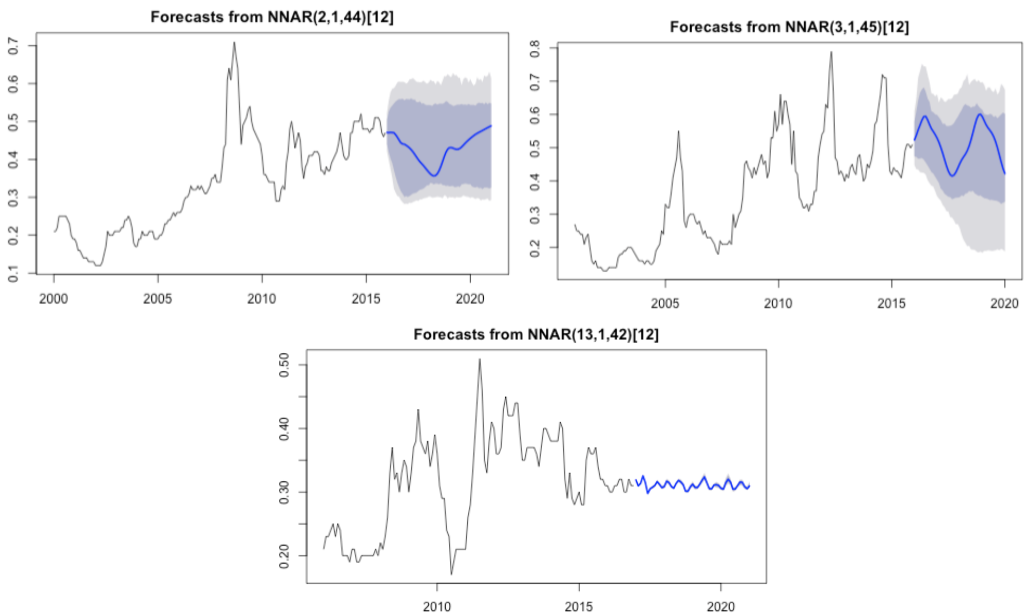


Figure 5.6: NNAR 5 year recursive forecast Wheat(left), Sorghum(right), Maize(bottom)

5.6 Forecast Accuracy

The performance of a forecasting method is evaluated based on the accuracy and efficiency of the predicted outcome. A good model minimizes the difference or the errors between the actual values and forecasted values. Although the evaluation results and performance assessment of each method is discussed in the next chapter, this subsection introduces the most common metrics for calculating errors according to several statistical literatures.

Root Mean Square Error (RMSE): is used as a primary evaluation metric in most studies. RMSE squares the sum of the difference between original and forecasted values for each time period and then divides this amount by the total number of the sample periods. The errors are squared to have an absolute value and by squaring them, RMSE punishes higher variance as with the error the value increases exponentially (Hastie, Tibshirani, and Friedman 2009). Here " y' is the predicted values, y is the actual values and n is the total number of data periods." RMSE tends to be sensitive to outliers.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y'-y)^2}{n}} \qquad (18)$$

Mean Absolute Error (MAE): "is also a scaled-dependent measure but it is computed by taking the mean of the absolute value of the error." In this method the error is measured in either direction, however it rather penalizes larger errors than smaller ones (Hastie, Tibshirani, and Friedman 2009). The interpretation of MAE is easier than RMSE since the unit of the dependent variable is directly transferable. MAE is also sensitive to outliers but to a lesser degree than RMSE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(y - y')| \qquad (19)$$

Mean Absolute Percentage Error (MAPE): "measures the absolute error relative to the actual values, which makes it easier to interpret the magnitude of the errors." MAPE unlike the previous metrices is a scaled independent measure. Therefore comparing different datasets is possible (Hyndman & Athanasopoulos, 2018). However MAPE tends to favor the values below the actual values. Some studies (Hyndman & Koehler, 2006) also noted that when values are close to zero MAPE becomes extremely large and sometimes undefines.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|(y-y')|}{y} \times 100 \qquad (20)$$

5.7 Tools

The forecasting models in this study are entirely constructed using the "R programming language Version 1.3.1093." Data manipulation and pre-processing were carries out using the '*tidyverse'* environment. Along with popular R libraries we have used *'ggplot*' and *'plot'* libraries for data visualization.

In order to implement the selected statistical forecasting methods, the study used '*auto.arima'* function from the *'forecast'* package, which is a generic function for predicting data from "time series or time series models." "The '*auto.arima'* function fits the best ARIMA model to a time series according to a given information criterion (such as AIC or BIC)." It operates as a "grid search" as it tries multiple combinations of p and q (also P and Q for seasonal models) parameters, then selects the model through which the given criterion is minimized.

For machine learning models the study used a specialized library for each model, designed for time series forecasting. However, the *caret'* (short for Classification And Regression Training) package is used in order to accommodate the parameter set-up procedure as well as streamlining the modeling process.

Finally the whole code is implemented on a computational capacity of a MacBook Air laptop with "2.2 GHz Intel Core i7 processor and 8 GB 1600 MHz DDR3 memory."

Chapter 6: Results and Evaluation

This chapter encompasses the evaluation results of each forecasting method based on the accuracy metrics mentioned in the previous chapter. Performance of each method is compared for each commodity and forecast horizon. In addition, key insights are discussed in detail.

6.1 Wheat

The comparative forecast evaluation results for Wheat prices are given on *Table (6.1)*. The long-run performance shows that (S)ARIMA expanding windows are the models with the lowest errors across all the metrics, closely followed by (S)ARIMA rolling windows. SVR with linear kernel consistently outperformed all other ML models and (S)ARIMA recursive forecasts over the long-run. The model fairly captures the upward trending as well as recurrent movement of the price and also converges with the actual values over the last 12 periods *(See appendix Figure A1.11)*. FFNN CV shows comparable performance to that of the SVR linear, but falls short in tracking the price dynamics of the intermittent fluctuations *(See appendix Figure A1.12)*. While KNN CV and KNN recursive shows lower RMSE than TDNN, TDNN tracks the movements and trend of the Wheat price better and converges at the end of the forecast horizon *(See appendix Figure A1.13)*. Finally KNN MIMO and SVR with RBF kernel displays worse performance for the long term forecast with the former overestimating while the latter underestimated the predicted values *(See appendix Figure A1.8* and *Figure A1.10)*.

5 year forecast horizon			
	RMSE	**MAE**	**MAPE**
SARIMA Expanding	0.02221	0.01645	0.03312
ARIMA Expanding	0.02228	0.01619	0.03250
ARIMA Rolling	0.02230	0.01620	0.03247
SARIMA Rolling	0.02235	0.01641	0.03302
SVR linear	0.08186	0.06875	0.14148
ARIMA Recursive	0.08256	0.07304	0.15398
SARIMA Recursive	0.08305	0.07348	0.15435
FFNN CV	0.08896	0.07543	0.15196
KNN CV	0.09715	0.08232	0.16051
KNN recursive	0.09996	0.08422	0.19723
TDNN	0.10058	0.08520	16.69646
SVR RBF	0.11195	0.08867	0.16972
KNN MIMO	0.13633	0.11778	0.26885

1 year forecast horizon			
	RMSE	**MAE**	**MAPE**
ARIMA Expanding	0.01335	0.01061	0.02680
ARIMA Rolling	0.01336	0.01060	0.02675
SARIMA Rolling	0.01372	0.01117	0.02818
SARIMA Expanding	0.01381	0.01133	0.02859
SVR RBF	0.04059	0.03250	0.08562
KNN CV	0.04355	0.03842	0.09980
TDNN	0.05906	0.05238	13.84401
SVR linear	0.06092	0.05154	0.13413
FFNN CV	0.06230	0.04998	0.13316
SARIMA Recursive	0.07425	0.06404	0.16726
ARIMA Recursive	0.07495	0.06458	0.16873
KNN recursive	0.12138	0.10698	0.28306
KNN MIMO	0.13670	0.11908	0.31606

Table 6.1: Performance evaluation of models on Wheat prices

In the short-run, SVR-RBF shows the highest performance with the lowest prediction error, although this model showed poor performance results in the long-run forecast. KNN CV follows SVR-RBF with a comparable performance and it resulted in medium prediction error in the long-run forecast as well. Next we have, FFNN CV, SVR-linear exhibiting an average short run prediction. While their rank changes depending on the metric, they still remain within the average error range. TDNN on the other hand shows an average level of errors according to RMSE and MAE in the "long-run and short run forecast." The MAPE however resulted in larger error values due to the presence of small actual values and the metrics preferred values higher than the actual values. Finally, KNN MIMO and KNN recursive achieved the lowest performance as their output results in overprediction within the short-term period *(See appendix Figure A1.8 & A1.9)*.

6.2 Sorghum

In the long-run forecast, ARIMA one step expanding and rolling as well as the recursive multi-step forecasts shows the lowest errors, but SARIMA is omitted since the seasonal model is not applicable in Sorghum prices *Table (6.2)*. FFNN CV performed better than other ML modes by achieving the lowest error in RMSE and MAE metrics. The model potentially traces the volatile nature of the prices to some extent until the sharp decline value that occurred around 2019 *(See appendix Figure A1.12)*. SVR-RBF and KNN CV shows a comparative performance after FFNN CV and both models capture the majority of the price trend and fluctuations *(See appendix Figure A1.10 and Figure A1.7)*. With close error comparison, TDNN shows an average prediction over the price oscillations and SVR-linear kernel with the predicted data points tracing the increasing price towards the forecast horizon *(See appendix Figure A1.13 and Figure A1.11)*. Lastly, KNN MIMO and KNN recursive are the worst performing models for sorghum prices, since their predicted values for the most part are higher than the actual values *(See appendix Figure A1.8* and *Figure A1.9)*.

5 year forecast horizon			
	RMSE	**MAE**	**MAPE**
ARIMA Rolling	0.07425	0.04801	0.12635
ARIMA Expanding	0.07837	0.04972	0.12631
ARIMA Recursive	0.13236	0.10145	0.22487
FFNN CV	0.14167	0.11342	0.28505
SVR RBF	0.14376	0.11699	0.27813
KNN CV	0.14409	0.14409	0.28196
TDNN	0.16838	0.12302	0.35636
SVR linear	0.17106	0.13770	0.32137
KNN MIMO	0.18489	0.15252	0.43148
KNN recursive	0.24363	0.19768	0.56829

1 year forecast horizon			
	RMSE	**MAE**	**MAPE**
ARIMA Expanding	0.03477	0.02559	0.04776
ARIMA Rolling	0.03495	0.02545	0.04752
TDNN	0.05769	0.04163	8.81372
KNN MIMO	0.06487	0.05065	0.10460
KNN recursive	0.08052	0.06454	0.13115
ARIMA Recursive	0.09891	0.08400	0.14531
FFNN CV	0.11252	0.08574	0.14597
KNN CV	0.12054	0.12054	0.16830
SVR RBF	0.12611	0.10062	0.17540
SVR linear	0.17110	0.13044	0.21928

Table 6.2: Performance evaluation of models on Sorghum prices

In the short-run forecast the ARIMA expanding and rolling models, still remains with lowest errors. In case of ML models, TDNN achieved the best performance, in which the forecasted values overlap with the actual ones during the first 12 periods *(See appendix Figure A1.13).* Although they showed poor performance in the long-run, KNN MIMO and KNN recursive resulted in lower levels of errors in the short run compared to other models. Next we have ARIMA recursive and FFNN CV resulting in average errors and track prices to some extent. With the highest errors, KNN CV and SVR models show the worst performance in the short-run forecast.

6.3 Maize

Similar to the previous commodities, for Maize prices (S)ARIMA models remain with the lowest errors in both forecast horizons. Among ML models, KNN CV resulted in the lowest forecast errors across all the metrics behind the (S)ARIMA models in the long-run *Table (6.3)*. The model moderately captures the price trend especially towards the end of the forecast horizon *(See appendix Figure A1.7)*. Following closely behind are SVR RBF, SVR linear and FFNN CV showing a comparable performance with similar prediction pattern *(See appendix Figure A1.10, A1.11 and A1.12)*. KNN MIMO, KNN recursive and TDNN are the models with the highest level of RMSE for Sorghum prices. However, KNN MIMO and TDNN appear to reasonably emulate the price innovations over the long run *(See appendix Figure A1.8* and *A1.13)*.

5 year forecast horizon			
	RMSE	**MAE**	**MAPE**
ARIMA Expanding	0.02881	0.02121	0.06138
SARIMA Expanding	0.02925	0.02175	0.06375
ARIMA Rolling	0.03047	0.02269	0.06601
SARIMA Rolling	0.03084	0.02312	0.06811
ARIMA Recursive	0.06444	0.04637	0.13127
SARIMA Recursive	0.06469	0.04717	0.13515
KNN CV	0.07406	0.05946	0.16608
SVR RBF	0.07813	0.06649	0.19052
TDNN	0.07886	0.05762	0.15306
SVR linear	0.08122	0.06825	0.19765
FFNN CV	0.08259	0.06798	0.19194
KNN recursive	0.08318	0.06885	0.20531
KNN MIMO	0.10320	0.08630	0.26582

1 year forecast horizon			
	RMSE	MAE	MAPE
ARIMA Expanding	0.01034	0.00813	0.02633
ARIMA Rolling	0.01037	0.00817	0.02644
SARIMA Expanding	0.01107	0.00879	0.02848
SARIMA Rolling	0.01122	0.00895	0.02899
SARIMA Recursive	0.01503	0.01202	0.03936
ARIMA Recursive	0.01622	0.01386	0.04532
KNN CV	0.08804	0.07255	0.19259
SVR RBF	0.09316	0.08263	0.22631
SVR linear	0.09845	0.08581	0.23737
FFNN CV	0.10092	0.08656	0.23353
KNN MIMO	0.11298	0.09483	0.20763
KNN recursive	0.11477	0.09910	0.21850
TDNN	0.13105	0.11625	0.25836

Table 6.3: Performance evaluation of models on Maize prices

Even in the short-run, the KNN CV persisted to achieve top performance with relatively low error margins. Closely behind are both the SVRs and FFNN model with comparable level errors among them and sometime trading places depending on the metric. Finally, KNN MIMO and KNN recursive reported the lowest performance with TDNN showing the high errors in the short-run forecast. It’s important to point out that, in case of maize prices the models suffered lower short-run prediction performance due to a shock causing the prices to sharply increase and then immediately decrease to lower levels at the beginning of the testing period. These changes within short period of time caused most of the models to responded by generating flatted predicted values to average out extremes.

6.7 Performance Summary

Evaluating prediction quality with RMSE assures us that we obtain unbiased forecasts from each method by examining the standard deviation of the residuals. Therefore, we compared the overall forecast performance by aggregating the RMSE score of each method and for each forecast horizon *(Figure 6.4).*

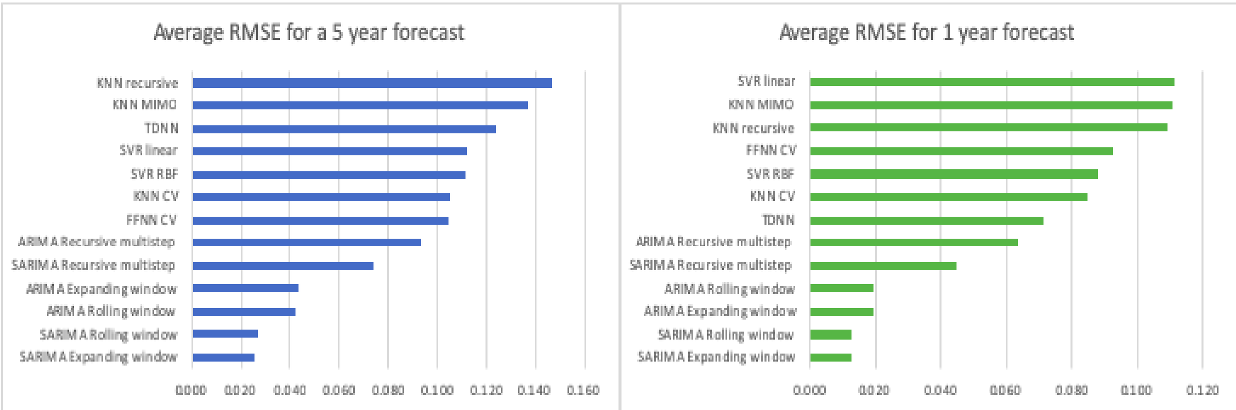


Figure 6.1: Performance comparison in RMSE per horizon

Base on the average RMSE scores, (S)ARIMA expanding windows have the lowest average forecast errors. Although they are slightly better, their results show insignificant difference from (S)ARIMA rolling windows. This means that it is beneficial to use all the available data for model training, which is the case for the expanding window forecast. Conversely, (S)ARIMA recursive forecasting performed worse out of the statistical methods. The prediction quality of (S)ARIMA recursive forecast deteriorates over the long-run as it accumulates errors due to the inclusion of previous forecast values. In addition, this method fails to capture the distinctive movements and innovations in all the commodity prices. It is apparent that the superior forecast performance of the (S)ARIMA models in both the "short and long-run" forecast horizons is derived from their ability to forecast one to two time periods ahead accurately. (S)ARIMA models rely on the correlation of past values, therefore as the time horizon increases and training observation becomes no longer available forecast performance significantly degrades.

Unlike statistical methods, machine learning approaches are able to discover non-linear relationships among the data points and capture the unique patterns of the univariate time series in each of the commodities. However, their prediction quality depends on the selected hyperparameters and as a result we see more performance variability among the models for each forecast horizon.

In the long run FFNN CV and KNN CV are the top performing models with FFNN CV showing slightly lower RMSE. Next we have the SVR models (Linear and RBF kernels) with comparable long-run performance by overcoming extreme price variations to a certain extent. TDNN and KNN MIMO and KNN recursive are at the bottom with the highest average errors. This indicates that optimal parameters obtained from the cross-validation procedure significantly affects prediction quality of a given model over the long-run. In our case FFNN obtained the optimal number of nodes and KNN CV obtained the optimal number of k or nearest neighbors with cross validation. Similarly, SVR models attained the optimal value of the 'Cost' parameter with cross validation procedure.

In the short-term forecast, TDNN displayed superior forecast performance by showing better tracking of the price trends of each commodity, while other models face challenges to keep low error rate *(See appendix Figure A1.13)*. It's observed that KNN CV consistently placed among the top performers even in the short-term prediction by showing a stable error level. SVR-RBF and FFNN CV exhibiting slightly higher average RMSE for short-term. Finally, KNN MIMO, KNN recursive and SVR linear are the poorest performers in the short-run forecast horizon.

Chapter 7. Conclusion

Prices of agricultural commodities have a big impact on the economies of Sub-Saharan African nations. The majority of disposable income is spent on food and changes on price levels drastically affect people's lives. Commodity prices have been falling drastically over the years there is a need for the establishment of a "long-term strategy" for SSA economies which should include the "reduced dependence on commodities and shifting into the production of manufactures or services" (Page and Hewitt 2001). However, it is difficult to address this situation within a short span of time. Thus, forecasting prices of commodities remains an integral part of agricultural policy planning and decision-making.

This thesis evaluated the forecasting performance of various statistical and machine learning approaches under the same settings and datasets, in order to determine the most accurate method for forecasting commodity prices. Statistical models of (ARIMA ,SARIMA) and machine learning models (KNN, SVR, ANN) with different forecasting strategies and configurations were extensively tested. The study used 20-year univariate monthly price data of Wheat(Ethiopia), Sorghum(Khartoum) and Maize(Nairobi) prices to compare the different categories of forecasting methods for "long-term(5 year) and short-term(1 year)" forecast horizons. Moreover, hyperparameters were optimized according to each model and dataset. Each model's performance was evaluated based on RMSE, MAE and MAPE metrics.

The evaluation outcomes reveal that, expanding and rolling (S)ARIMA forecasting strategies resulted in the lowest prediction error across all commodities. However the methods only forecast the next time period ahead. The recursive (S)ARIMA forecast showed lower errors at the beginning of the testing period but the prediction quality declined overtime. Since the model assumes a stationarity process of time series data, it is unable to handle the non-linear property of the commodity prices and results in large errors over the long-run.

On the other hand, "machine learning methods" capture the highly volatile price trends of the commodity prices by mapping the non-linear relationship and generating approximate predictions. "The models are able to learn any functional form from the training data and generalize estimates with the testing data." Thus, performance of each model depends on the given commodity dataset. SVR-linear and SVR-RBF performed best in predicting Wheat prices for long and short-run horizons respectively. For Sorghum prices FFNN and TDNN offered the most accurate predictions for long and short term respectively. In forecasting Maize prices KNN CV outperformed all other models in both forecast horizons.

Our findings remain consistent with the literature of (Mitchell, 1997) and (Wolpert & Macready, 1997) and confirm that the performance of particular ML models depends on the given dataset. Given our SSA commodity dataset, the single-layered FFNN CV and TDNN achieved the highest overall prediction performance in long and short-run forecast horizons respectively. KNN CV and SVR-RBF reported reasonable performance by consistently maintaining a steady level of prediction error over both horizons.

The empirical results show that statistical approaches are more suitable for short-term forecasting but fail to provide significant long-run information regarding price dynamics. Machine learning approaches on the other hand make better predictions and deliver pertinent information over the long-run horizon. In the context of agriculture price forecasting, machine learning models are substantially preferred due their superior ability in mapping the price variations and predicting the direction of change. These turning points capture the prevailing business cycles and provide insights into the future price trends of the commodities. However, it is also important to use appropriate ML methods that are specifically tested on the given dataset. As it takes into consideration the relevant information in the modelling process and as a result provides more information on the future price behavior of the given commodity. In addition, multiple ML models could be applied simultaneously or one at a time according to the forecasting needs of stakeholders. As a result, data driven decisions and policies could be made regarding food prices based on the forecast information.

This study omitted several methods such as the SVR with polynomial kernel and ANN with multiple hidden layers due to their high computational requirement. Hence, further research should consider evaluating these absent models and more categories of statistical, machine learning as well as hybrid models. The experiment can be extended to more commodities with larger training dataset and over multiple forecasting horizons, in order to conclude on the prediction power of the models.

Bibliography

Anggraeni, Wiwik, Faizal Mahananto, Ayusha Qamara Sari, Zulkifli Zaini, Kuntoro Boga Andri, and Sumaryanto. 2019. “Forecasting the Price of Indonesia’s Rice Using Hybrid Artificial Neural Network and Autoregressive Integrated Moving Average (Hybrid NNs-ARIMAX) with Exogenous Variables.” *Procedia Computer Science*, The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia, 161 (January): 677–86. https://doi.org/10.1016/j.procs.2019.11.171.

Athiyarath, Srihari, Mousumi Paul, and Srivatsa Krishnaswamy. 2020. “A Comparative Study and Analysis of Time Series Forecasting Techniques.” *SN Computer Science* 1 (3): 175. https://doi.org/10.1007/s42979-020-00180-5.

Awad, Mariette, and Rahul Khanna. 2015. “Support Vector Regression.” In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, edited by Mariette Awad and Rahul Khanna, 67–80. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4302-5990-9_4.

Ayankoya, Kayode, Andre P. Calitz, and Jean H. Greyling. 2016. “Real-Time Grain Commodities Price Predictions in South Africa: A Big Data and Neural Networks Approach.” *Agrekon* 55 (4): 483–508. https://doi.org/10.1080/03031853.2016.1243060.

Berrar, Daniel. 2019. “Cross-Validation.” In *Encyclopedia of Bioinformatics and Computational Biology*, edited by Shoba Ranganathan, Michael Gribskov, Kenta Nakai, and Christian Schönbach, 542–45. Oxford: Academic Press. https://doi.org/10.1016/B978-0-12-809633-8.20349-X.

Bisgaard, Søren, and Murat Kulahci. 2011. *Time Series Analysis and Forecasting by Example*. Wiley Series in Probability and Statistics. Wiley. https://doi.org/10.1002/9781118056943.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer-Verlag. https://www.springer.com/gp/book/9780387310732.

Box, G. E. P., and D. R. Cox. 1964. “An Analysis of Transformations.” *Journal of the Royal Statistical Society. Series B (Methodological)* 26 (2): 211–52.

Box, George Edward Pelham, and Gwilym Jenkins. 1990. *Time Series Analysis, Forecasting and Control*. USA: Holden-Day, Inc.

Brockwell, Peter J., and Richard A. Davis. 2002. *Introduction to Time Series and Forecasting*. 2nd ed. Springer Texts in Statistics. New York: Springer-Verlag. https://doi.org/10.1007/b97391.

Brown, Molly E., Jorge E. Pinzon, and Stephen D. Prince. 2008. “Using Satellite Remote Sensing Data in a Spatially Explicit Price Model: Vegetation Dynamics and Millet Prices.” *Land Economics* 84 (2): 340–57. https://doi.org/10.3368/le.84.2.340.

Cao, L.J., and F.E.H. Tay. 2003. “Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting.” *IEEE Transactions on Neural Networks* 14 (6): 1506–18. https://doi.org/10.1109/TNN.2003.820556.

Cenas, Paulo V. 2017. “Forecast of Agricultural Crop Price Using Time Series and Kalman Filter Method.” *Asia Pacific Journal of Multidisciplinary Research* 5 (4): 7.

Chou, Jui-Sheng, and Ngoc-Tri Ngo. 2016. “Time Series Analytics Using Sliding Window Metaheuristic Optimization-Based Machine Learning System for Identifying Building Energy Consumption Patterns.” *Applied Energy* 177 (C): 751–70.

Cryer, Jonathan D., and Justin McCarthy. 1986. *Time Series Analysis*. Duxbury Press.

Das, Panchanan. 2019. *Econometrics in Theory and Practice: Analysis of Cross Section, Time Series and Panel Data with Stata 15.1*. Springer Nature.

Dreyfus, Gérard. 2005. *Neural Networks: Methodology and Applications*. Berlin Heidelberg: Springer-Verlag. https://doi.org/10.1007/3-540-28847-3.

Elzaki, Raga, Muhammet Yunus Sisman, and Mohammed Al-Mahish. 2021. “Rural Sudanese Household Food Consumption Patterns.” *Journal of the Saudi Society of Agricultural Sciences* 20 (1): 58–65. https://doi.org/10.1016/j.jssas.2020.11.004.

Evans, Edward A., and Sikavas Nalampang. 2009. “Forecasting Price Trends in the U.S. Avocado (Persea Americana Mill.) Market.” *Journal of Food Distribution Research* 40 (2): 1–10.

Fang, Yongmei, Bo Guan, Shangjuan Wu, and Saeed Heravi. 2020. “Optimal Forecast Combination Based on Ensemble Empirical Mode Decomposition for Agricultural Commodity Futures Prices.” *Journal of Forecasting* 39 (6): 877–86.

FAO. 2002. “The State of Food Insecurity in the World 2002: When People Must Live with Hunger and Fear Starvation.” Terme di Caracalla, 00100 Rome, Italy: Food & Agriculture Organization. http://www.fao.org/3/y7352e/y7352e00.htm.

Fischer, Thomas, Christopher Krauss, and Alex Treichel. 2018. “Machine Learning for Time Series Forecasting - a Simulation Study.” FAU Discussion Papers in Economics 02/2018. Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics. https://econpapers.repec.org/paper/zbwiwqwdp/022018.htm.

Gad, Ahmed Fawzy, and Fatima Ezzahra Jarmouni. 2020. *Introduction to Deep Learning and Neural Networks with Python^{TM}: A Practical Guide*. San Diego: Academic Press.

Ghosh-Dastidar, Samanwoy, and Hojjat Adeli. 2009. “Spiking Neural Networks.” *International Journal of Neural Systems* 19 (04): 295–308. https://doi.org/10.1142/S0129065709002002.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Adaptive Computation and Machine Learning Series. Cambridge, MA, USA: MIT Press.

Griffith, Garry, and D. Vere. 2000. “Forecasting Agricultural Commodity Prices,” January.

Gupta, Madan M., Liang Jin, Noriyasu Homma, and Lotfi A. Zadeh. 2005. *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*. Wiley-IEEE Press. https://doi.org/10.1002/0471427950.

Hagan, Martin T., Howard B. Demuth, Mark H. Beale, and Orlando De Jesús. 2014. *Neural Network Design*. 2nd edition. s.L: Martin Hagan.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd ed. Springer Series in Statistics. New York: Springer-Verlag. https://doi.org/10.1007/978-0-387-84858-7.

Hyndman, Rob J., and George Athanasopoulos. 2018. *Forecasting: Principles and Practice*. 2nd ed. OTexts.

Hyndman, Rob J., and Anne B. Koehler. 2006. “Another Look at Measures of Forecast Accuracy.” *International Journal of Forecasting* 22 (4): 679–88.

Iqbal, Javed. 2001. “Forecasting Methods: A Comparative Analysis.” 23856. *MPRA Paper*. MPRA Paper. University Library of Munich, Germany. https://ideas.repec.org/p/pra/mprapa/23856.html.

Jarque, Carlos M., and Anil K. Bera. 1980. “Efficient Tests for Normality, Homoscedasticity and Serial Independence of Regression Residuals.” *Economics Letters* 6 (3): 255–59. https://doi.org/10.1016/0165-1765(80)90024-5.

Jayne, T. S., Robert J. Myers, and James Nyoro. 2008. “The Effects of NCPB Marketing Policies on Maize Market Prices in Kenya.” *Agricultural Economics* 38 (3): 313–25.

Jeong, Minje, Young Jin Lee, and Youngchan Choe. 2017. “Forecasting Agricultural Commodity Price: The Case of Onion.” *Journal of Research in Humanities and Social Science* 5 (6): 78–81.

Jha, Girish K., and Kanchan Sinha, eds. 2013. “Agricultural Price Forecasting Using Neural Network Model: An Innovative Information Delivery System.” *Agricultural Economics Research Review Agricultural Economics Research Review*. https://doi.org/10.22004/ag.econ.162150.

Kaur, Manpreet, Heena Gulati, and Harish Kundra. 2014. “Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications.” *International Journal of Computer Applications* 99 (12): 1–3.

Khashei, Mehdi, Mehdi Bijari, and Gholam Ali Raissi Ardali. 2009. “Improvement of Auto-Regressive Integrated Moving Average Models Using Fuzzy Logic and Artificial Neural Networks (ANNs).” *Neurocomputing*, Brain Inspired Cognitive Systems (BICS 2006) / Interplay Between Natural and Artificial Computation (IWINAC 2007), 72 (4): 956–67. https://doi.org/10.1016/j.neucom.2008.04.017.

Kim, Kyoung-jae. 2003. "Financial Time Series Forecasting Using Support Vector Machines." *Neurocomputing*, Support Vector Machines, 55 (1): 307–19. https://doi.org/10.1016/S0925-2312(03)00372-2.

Kohzadi, Nowrouz, Milton S. Boyd, B. Kermanshahi, and Iebeling Kaastra. 1996. "A Comparison of Artificial Neural Network and Time Series Models for Forecasting Commodity Prices." *Neurocomputing*. https://doi.org/10.1016/0925-2312(95)00020-8.

Lai, T. H. 1991. "Time Series Analysis Univariate and Multivariate Methods: William W.S. Wei, (Addison-Wesley, Reading, MA, 1990)." *International Journal of Forecasting* 7 (3): 389–90.

Li, Gan-qiong, Shi-wei Xu, and Zhe-min Li. 2010. "Short-Term Price Forecasting For Agro-Products Using Artificial Neural Networks." *Agriculture and Agricultural Science Procedia*, International Conference on Agricultural Risk and Food Security 2010, 1 (January): 278–87. https://doi.org/10.1016/j.aaspro.2010.09.035.

Lolli, F., R. Gamberini, A. Regattieri, E. Balugani, T. Gatos, and S. Gucci. 2017. "Single-Hidden Layer Neural Networks for Forecasting Intermittent Demand." *International Journal of Production Economics* 183 (January): 116–28. https://doi.org/10.1016/j.ijpe.2016.10.021.

Luo, Changshou, Qingfeng Wei, Liying Zhou, Junfeng Zhang, and Sufen Sun. 2010. "Prediction of Vegetable Price Based on Neural Network and Genetic Algorithm." In , AICT-346:672. Springer. https://doi.org/10.1007/978-3-642-18354-6_79.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward." *PLOS ONE* 13 (3): e0194889. https://doi.org/10.1371/journal.pone.0194889.

Michel, Lucie, and David Makowski. 2013. "Comparison of Statistical Models for Analyzing Wheat Yield Time Series." https://doi.org/10.1371/journal.pone.0078615.

Ming-hua, Wei, Zhou Qiaolin, Yang Zhijian, and Zheng Jingui. 2012. "Prediction Model of Agricultural Product's Price Based on the Improved BP Neural Network." *2012 7th International Conference on Computer Science & Education (ICCSE)*. https://doi.org/10.1109/ICCSE.2012.6295150.

Mitchell, Tom M. 1997. *Machine Learning*.

Nasira, G M, and N Hemageetha. 2012. "Forecasting Model for Vegetable Price Using Back Propagation Neural Network." *International Journal of Computational Intelligence and Informatics* 2 (2): 6.

Ocran, M.K., and Nicholas Biekpe. 2007. "Forecasting Volatility in Sub-Saharan Africa's Commodity Markets." *Investment Management & Financial Innovations* 4 (January): 91–102.

Okori, Washington, and J. Obua. 2011. "Machine Learning Classification Technique for Famine Prediction." *Undefined*, no. 2: 991–96.

Owen P. Hall, Jr. 2010. "Artificial Intelligence Techniques Enhance Business Forecasts." *2002 Volume 5 Issue 2*, no. 2 (August). https://gbr.pepperdine.edu/2010/08/artificial-intelligence-techniques-enhance-business-forecasts/.

Page, Sheila, and Adrian Hewitt. 2001. *World Commodity Prices: Still a Problem for Developing Countries*. Revised edition. London: Overseas Development Institute.

Ripley, Brian D. 1996. *Pattern Recognition and Neural Networks*. Cambridge ; New York: Cambridge University Press.

Ruekkasaem, Lakkana, and Montalee Sasananan. 2018. "Forecasting Agricultural Products Prices Using Time Series Methods for Crop Planning." *International Journal of Mechanical Engineering and Technology* 9 (July): 957–71.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1986. "Learning Representations by Back-Propagating Errors." *Nature* 323 (6088): 533–36. https://doi.org/10.1038/323533a0.

Sajda, Paul. 2006. "Machine Learning for Detection and Diagnosis of Disease." *Annual Review of Biomedical Engineering* 8 (1): 537–65. https://doi.org/10.1146/annurev.bioeng.8.061505.095802.

Shee, Hassan, Wilson Cheruiyot, and Stephen Kimani. 2014. "Application of K-Nearest Neighbour Classification in Medical Data Mining" 4 (April).

Shumway, Robert H., and David S. Stoffer. 2017. *Time Series Analysis and Its Applications: With R Examples*. 4th ed. Springer Texts in Statistics. Springer International Publishing. https://doi.org/10.1007/978-3-319-52452-8.

Subhasree, Methirumangalath, and Priya Arun. 2015. "Vegetable Price Prediction Based On Time Series Analysis | PDF | Machine Learning | Mean Squared Error." *IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS))* 5 (6). https://www.scribd.com/document/355729174/VEGETABLE-PRICE-PREDICTION-BASED-ON-TIME-SERIES-ANALYSIS.

Tang, Huajin, Kay Chen Tan, and Zhang Yi. 2007. *Neural Networks: Computational Models and Applications*. Studies in Computational Intelligence. Berlin Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-540-69226-3.

Thanh Noi, Phan, and Martin Kappas. 2018. "Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery." *Sensors* 18 (1): 18. https://doi.org/10.3390/s18010018.

Ticlavilca, A. M., D. Feuz, and M. McKee. 2010. "Forecasting Agricultural Commodity Prices Using Multivariate Bayesian Machine Learning Regression." In *Proceedings of the NCCC- 15 134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management*. St. Louis, MO. https://www.semanticscholar.org/paper/Forecasting-Agricultural-Commodity-Prices-Using-Ticlavilca-Feuz/1f9773a0f4d1291b88a69805dea7b07be2ab0ed6.

Varma, G. Nikhila, and K. Padma. 2019. “Forecasting Agricultural Commodity Pricing Using Neural Network-Based Approach.” *International Journal of Business Information Systems* 31 (4): 517. https://doi.org/10.1504/IJBIS.2019.101584.

Wang, Jue, Zhen Wang, Xiang Li, and Hao Zhou. 2019. “Artificial Bee Colony-Based Combination Approach to Forecasting Agricultural Commodity Prices.” *International Journal of Forecasting*, December, S0169207019302304. https://doi.org/10.1016/j.ijforecast.2019.08.006.

Witten, I. H., Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed. Morgan Kaufmann Series in Data Management Systems. Burlington, MA: Morgan Kaufmann.

Wolpert, D.H., and W.G. Macready. 1997. “No Free Lunch Theorems for Optimization.” *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82. https://doi.org/10.1109/4235.585893.

Xiong, Tao, Chongguang Li, Yukun Bao, Zhongyi Hu, and Lu Zhang. 2015. “A Combination Method for Interval Forecasting of Agricultural Commodity Futures Prices.” *Knowledge-Based Systems* 77 (C): 92–102. https://doi.org/10.1016/j.knosys.2015.01.002.

Zhang, Dabin, Shanying Chen, Ling Liwen, and Qiang Xia. 2020. “Forecasting Agricultural Commodity Prices Using Model Selection Framework With Time Series Features and Forecast Horizons.” *IEEE Access* 8: 28197–209. https://doi.org/10.1109/ACCESS.2020.2971591.

Zhang, Dongqing, Guangming Zang, Jing Li, Kaiping Ma, and Huan Liu. 2018. “Prediction of Soybean Price in China Using QR-RBF Neural Network Model,” September. https://doi.org/10.1016/j.compag.2018.08.016.

Zhang, Guoqiang, B. Eddy Patuwo, and Michael Y. Hu. 1998. “Forecasting with Artificial Neural Networks:: The State of the Art.” *International Journal of Forecasting* 14 (1): 35–62. https://doi.org/10.1016/S0169-2070(97)00044-7.

Appendix

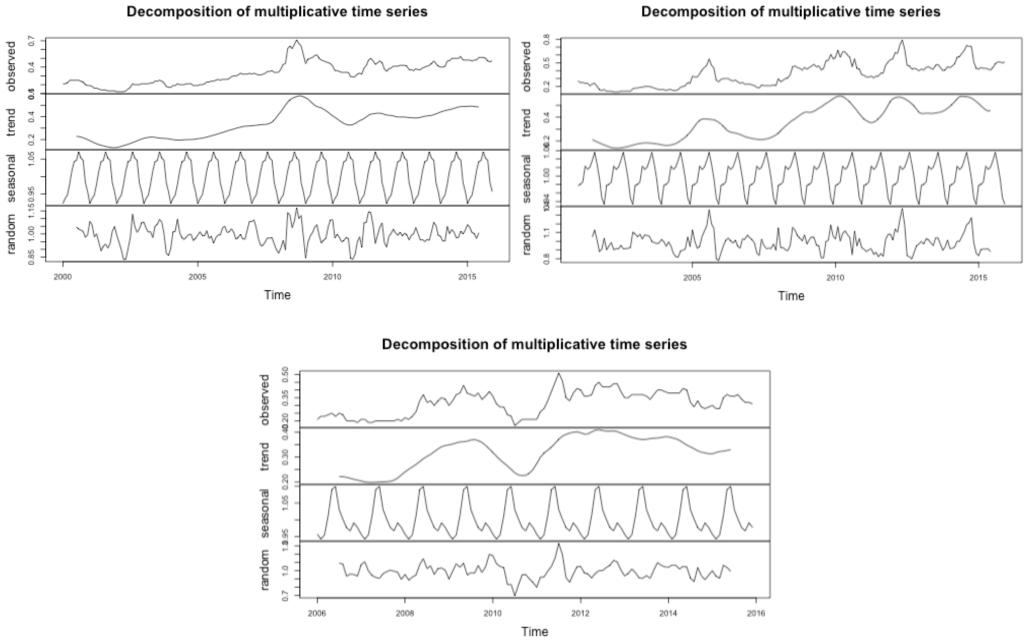


Figure A1.1: Timer series Decomposition Wheat (right), Sorghum (left) and Maize (bottom)

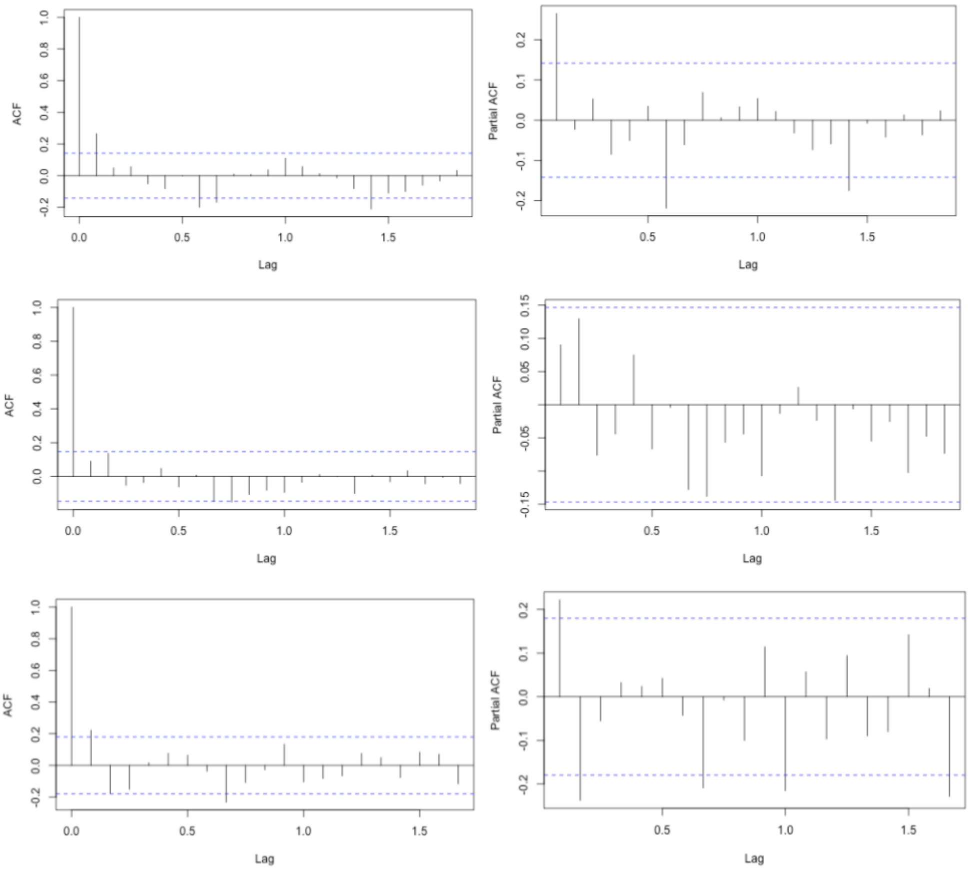


Figure A1.2: ACF and PACF (Diff) plots of Wheat (top), Sorghum (middle) and Maize (bottom)

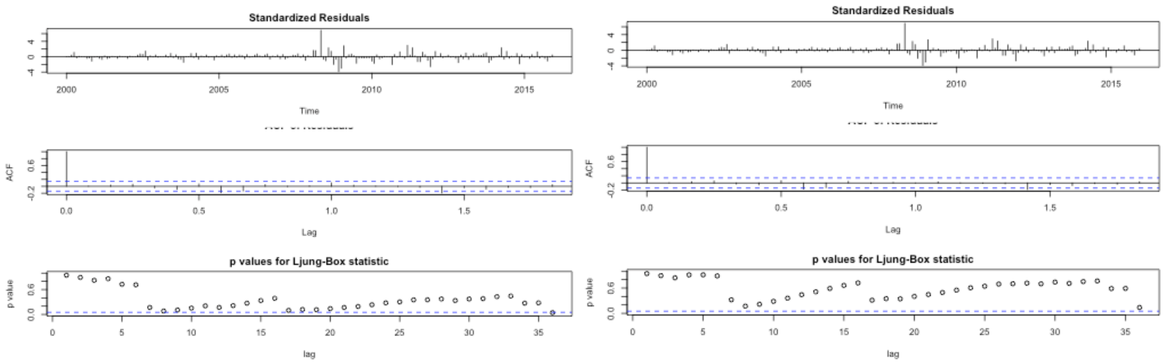


Figure A1.3: Residual diagnostics plots for ARIMA(left) and SARIMA(right) models of Wheat prices

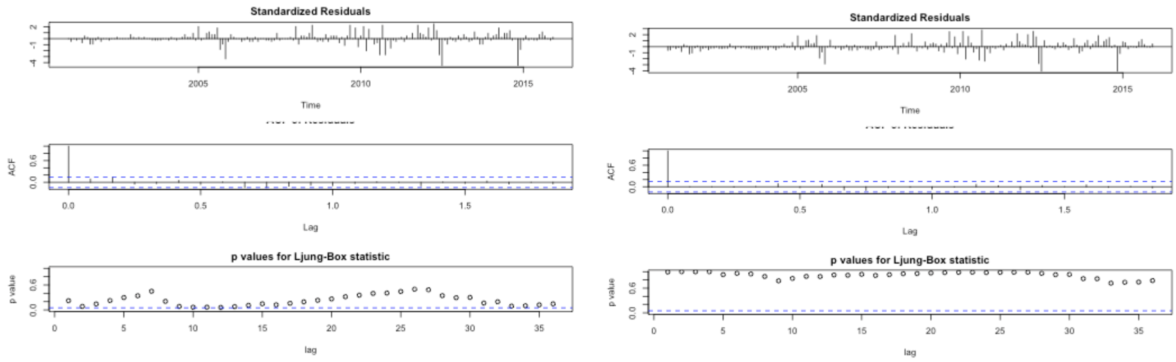


Figure A1.4: Residual diagnostics of ARIMA models for level(left) and differenced(right) series of Sorghum prices

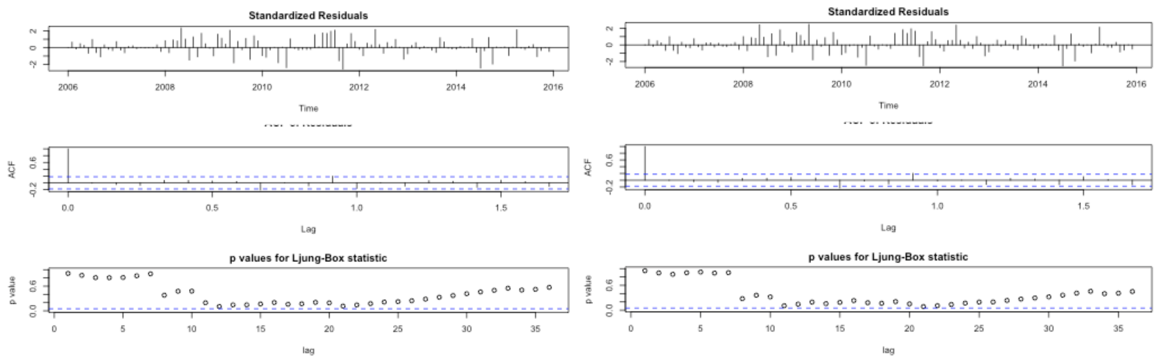


Figure A1.5: Residual diagnostics plots ARIMA(left) and SARIMA(right) models of Maize prices

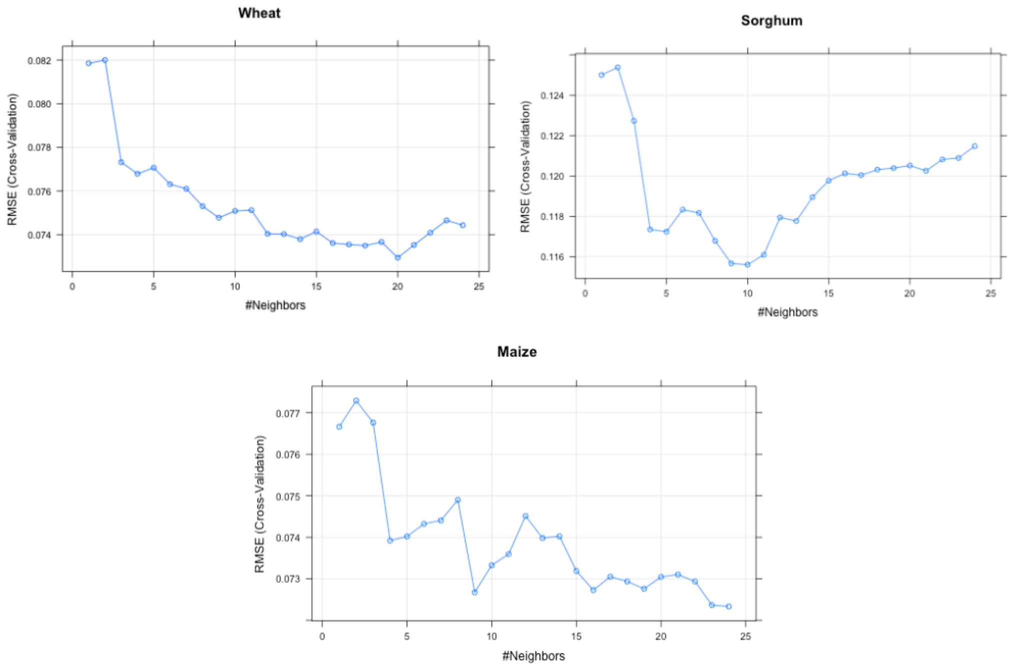


Figure A1.6: Optimal hyper parameter values of KNN for each commodity

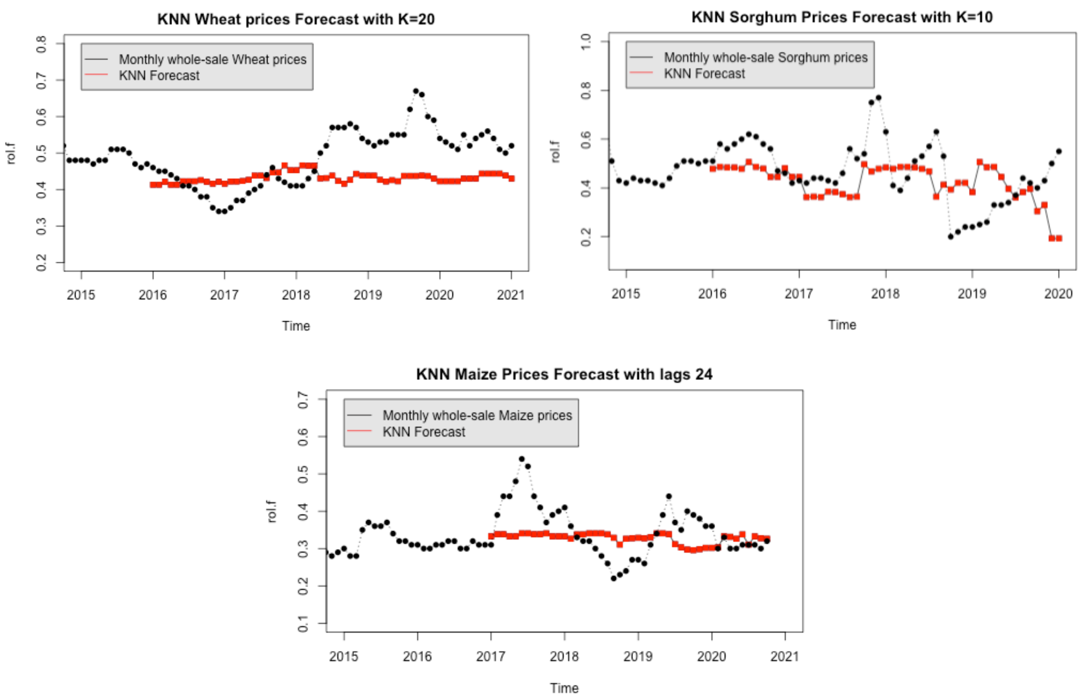


Figure A1.7: KNN CV regression prediction

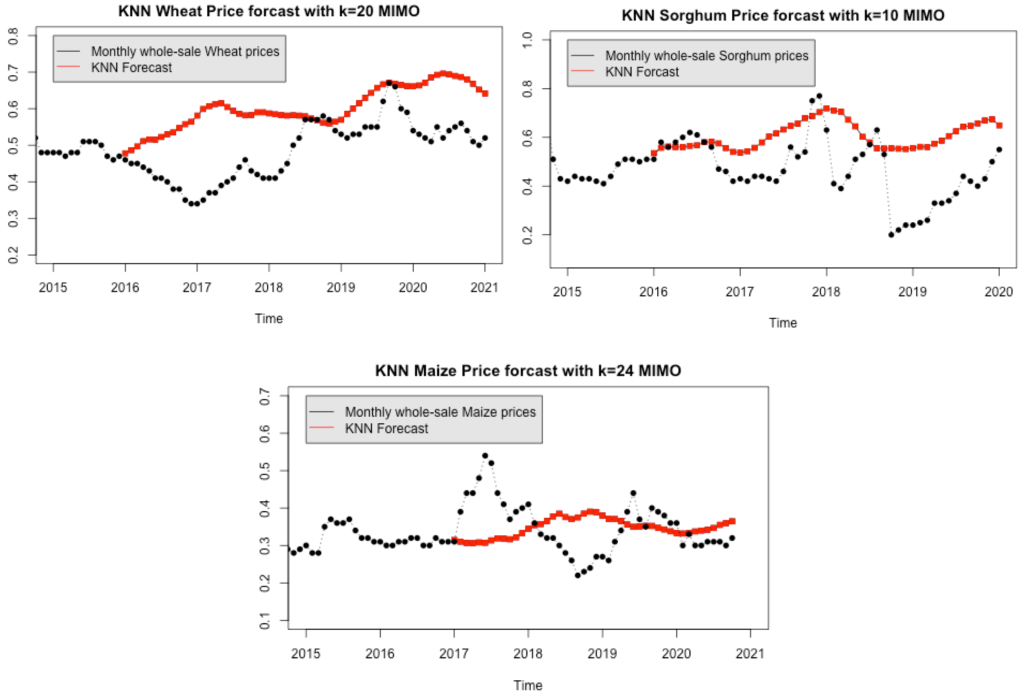


Figure A1.8: KNN MIMO forecasts

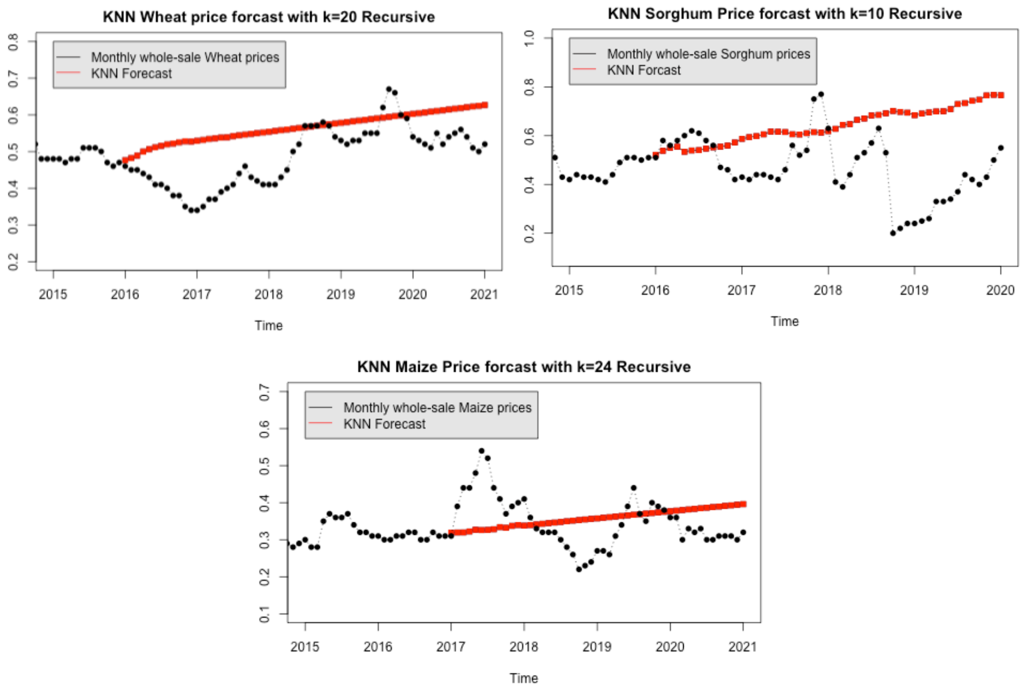


Figure A1.9: KNN Recursive forecasts

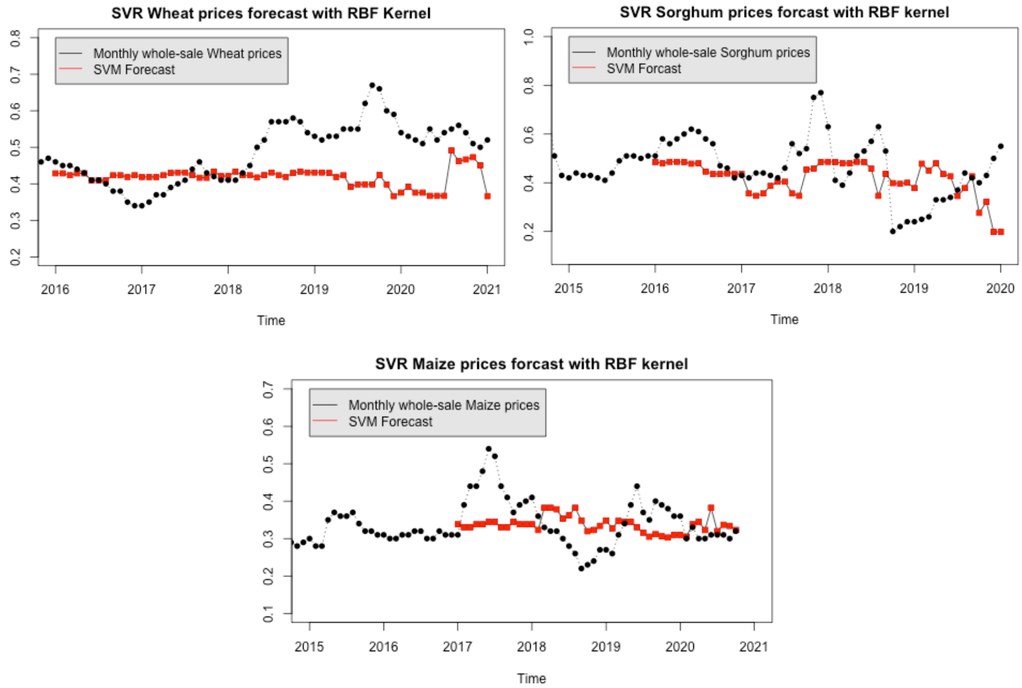


Figure A1.10: SVR RBF kernel prediction

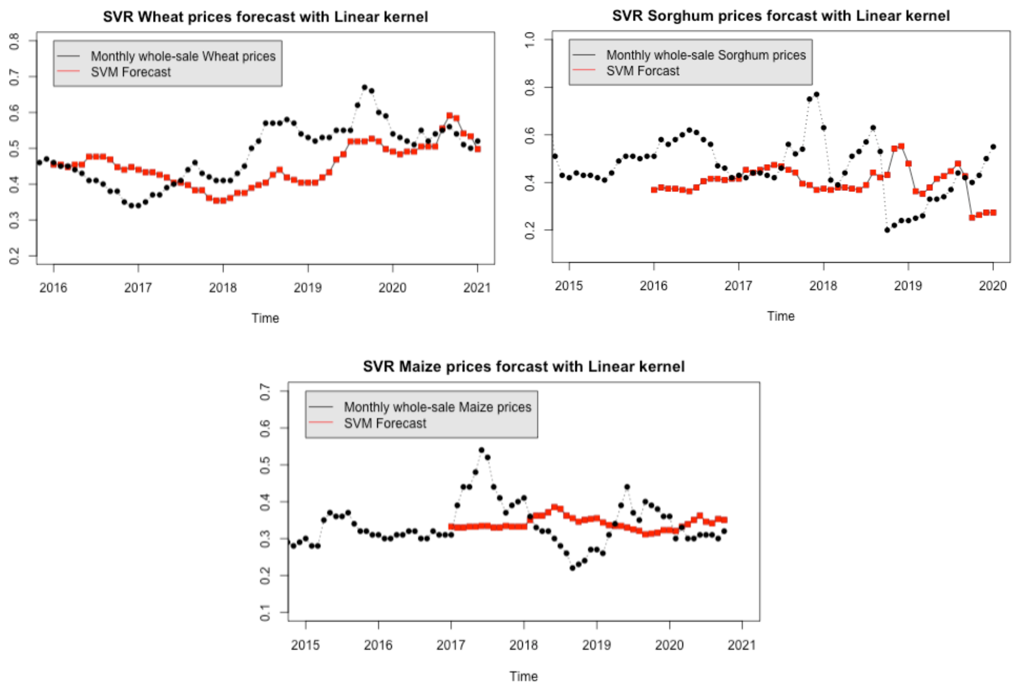


Figure A1.11: SVR Linear kernel prediction

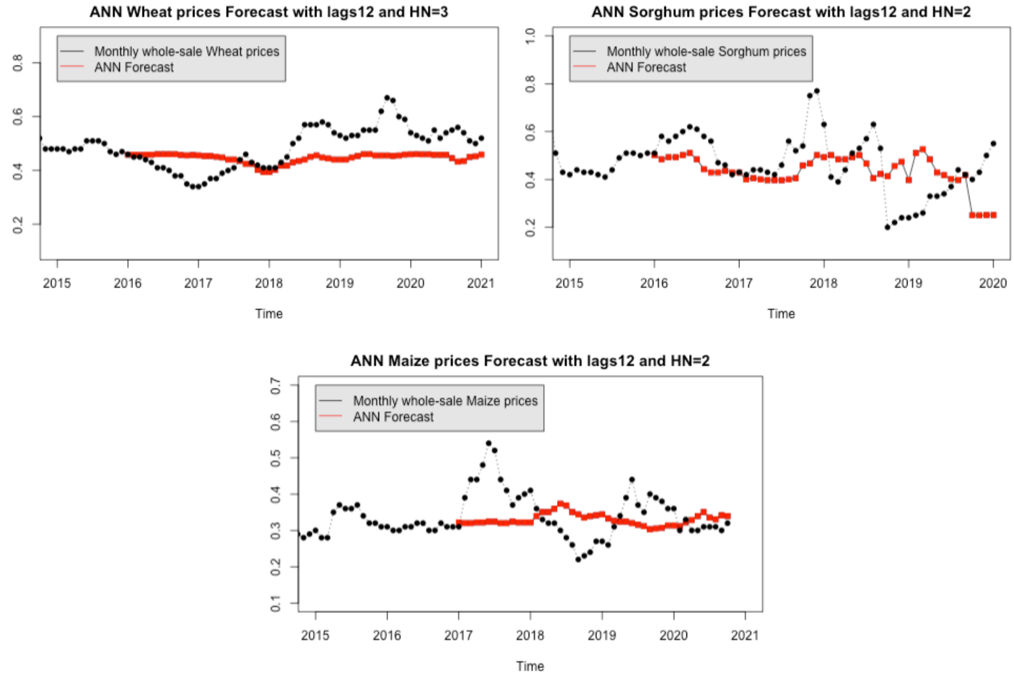


Figure A1.12: FFNN CV prediction

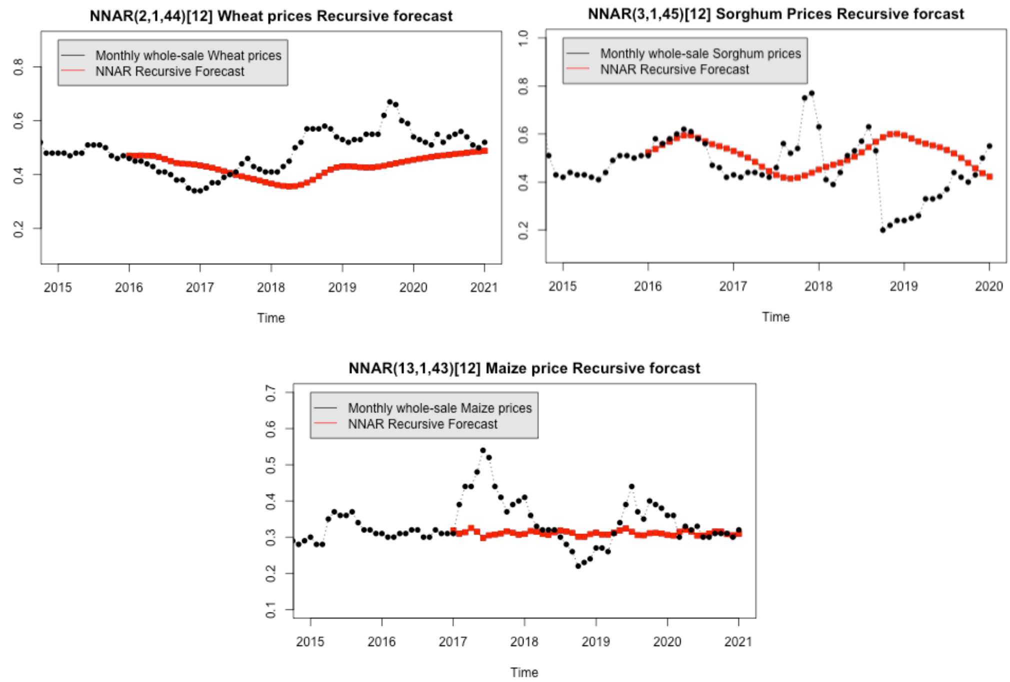


Figure A1.13: TDNN forecast

Declaration of Authorship

“I hereby declare that I wrote this thesis paper independently, without assistance from external parties, and without use of other resources than those indicated. All information taken from other publications or sources in text or in meaning are duly acknowledged in the text. The written and electronic forms of the thesis paper are the same. I give my consent to have this thesis checked by plagiarism software.“

Frankfurt am Main, September 28, 2021