Intelligent forecasting of economic growth for African economies: Artificial neural networks versus time series and structural econometric models*

Chuku Chuku[†1,2], Jacob Oduor[‡1], and Anthony Simpasa[§1]

[1] Macroeconomics Policy, Forecasting and Research Department, African Development Bank
[2] Centre for Growth and Business Cycle Research, University of Manchester, U.K., and Department of Economics, University of Uyo, Nigeria

Preliminary Draft April 2017

Abstract

Forecasting economic time series for developing economies is a challenging task, especially because of the peculiar idiosyncrasies they face. Models based on computational intelligence systems offer an advantage through their functional flexibility and inherent learning ability. Nevertheless, they have hardly been applied to forecasting economic time series in this kind of environment. This study investigates the forecasting performance of artificial neural networks in relation to the more standard Box-Jenkins and structural econometric modelling approaches applied in forecasting economic time series in African economies. The results, using different forecast performance measures, show that artificial neural network models perform somewhat better than structural econometric and ARIMA models in forecasting GDP growth in selected frontier economies, especially when the relevant commodity prices, trade, inflation, and interest rates are used as input variables. There are, however, some country-specific exceptions. Because the improvements are only marginal, it is important that practitioners hedge against wide errors by using a combination of neural network and structural econometric models for practical applications.

Keywords: Forecasting, artificial neural networks, ARIMA, backpropagation, economic growth, Africa
JEL Classification:

---

*Paper prepared for presentation at the workshop on forecasting for developing economies to be held at the IMF organized by the IMF, International Institute of Forecasters, American University, and George Washington Research Program on Forecasting.

†chuku.chuku@manchester.ac.uk; c.chuku@afbd.org; Phone: +44 777 660 4518
‡j.oduor@afdb.org
§a.simpasa@afdb.org

1 Introduction

Forecasting economic time series is a challenging task, and more so for developing economies where a host of factors usually not accounted for in mainstream economic thinking play significant roles in shaping the overall macroeconomic outcomes in these environments. The popularity of computational intelligence systems—particularly artificial neural networks—for dealing with nonlinearities and forecasting of time series data has continued to receive substantial attention in the literature, especially in the last two decades (see recent examples and reviews in Giusto & Piger, 2017; Teräsvirta, Van Dijk, & Medeiros, 2005; Crone, Hibon, & Nikolopoulos, 2011; De Gooijer & Hyndman, 2006; Ghiassi, Saidane, & Zimbra, 2005). The major attraction of this class of models lies in their flexible nonlinear modeling capabilities. With an artificial neural network (ANN), for example, there is no need to specify a particular model form, rather, the model is adaptively formed based on the features presented from the data, making it appropriate for situations where apriori theoretical expectations do not hold or are violated.

But despite the popularity and advantages of the ANN approach, they have hardly ever been applied to forecasting economic time series in developing economies in spite of the numerous applications that show their superior performance (under certain conditions) over traditional forecasting models in developed economies. For examples, Tkacz (2001) finds that neural networks outperform linear models in the prediction of annual GDP growth for Canada, but not quarterly GDP growth; Heravi, Osborn, and Birchenhall (2004) who that neural network models dominate linear ones in predicting the direction of change of industrial production for European economies, but linear models generally outperform neural network models in out of sample forecasts at horizons of up to a year; Feng and Zhang (2014) using ANN versus GM(1,1) models to perform tendency forecasting of economic growth in cities of Zhejiang, China, find that forecast results from ANN were better and more effcient that those from the GM model.

Our objective is to determine whether forecasting by artificial neural networks, which have an inherent learning ability, provide superior forecasting performance when compared to traditional techniques such as time series and structural econometric models. We provide evidence from in- and out-of-sample forecast experiments on selected frontier economies in Africa. Specifically, we use neural networks to forecast GDP growth in South Africa, Nigeria, and Kenya and compare the results with traditional ARIMA and structural econometric models. We examine the forecast performance measures using absolute and relative forecast evaluation criteria—the mean squared prediction error and the mean absolute percentage error.

Overall, our results show that artificial neural network models perform somewhat better than structural econometric models and ARIMA models in forecasting GDP growth in developing economies, especially when the relevant primary commodity prices, trade, inflation, and interest rates are used as the input variables. The most probable explanation for the superior performance of ANN models is because they are better able to capture the non-linear and chaotic behaviour of the important input variables that help to explain growth in many developing economies in Africa. There are, however, some country-specific exceptions, and also because the improvements are only marginal, it is important that practitioners hedge against wide errors by using a combination of neural network and structural econometric models for practical applications.

Moreover, some practical implications emerge from the present study. First, to the extent that complexity is not overemphasized, parsimonious artificial neural network models can be used to provide benchmark forecasts for economic and financial variables in developing economies that are exposed to potential chaotic and external influences on growth determination Second, like many statistical forecasting models, neural network systems are capable of also misleading and producing outlier forecasts at certain data points (as we would see in the cases of Nigeria and Kenya), it is recommended that forecasts from neural network models should always be revalidated with forecasts from a structural econometric model. Finally, time series ARIMA models should only be considered as a last resort for forecasting in these kinds of environment, as they almost always perform worse than others, perhaps because of the sudden changes and chaotic pattern of macroeconomic variables in developing economies.

The rest of the paper is organized as follows. In Section 2, we present the forecasting models considered in the paper with some description of the algorithm for backpropagation in neural network models and the forecast performance measures used. In Section 3, we describe the data, sample, and features of the input variables. In Section 4, we present the results of the forecasting exercise and discuss some implications. Section 5 concludes.

2 Forecasting models

In this section, we present the different forecasting models used in our forecasting competition. Because our emphasis is on computational intelligence forecasting, we present a fairly elaborate description of the use of artificial neural networks in forecasting economic growth in a developing economy context; while the more familiar time series and structural econometric models are discussed briefly.

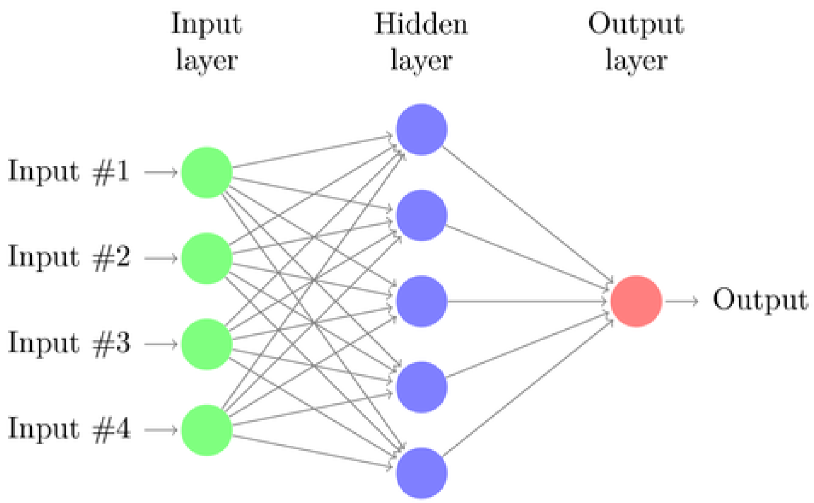
2.1 Artificial neural networks

Artificial neural networks are models designed to mimic the biological neural system—especially the brain and are composed of interconnected processing elements called neurons. Each neuron receives information or signals from external stimuli or other nodes and processes this information locally through an activation function; after which it produces a transformed output signal and sends it to other neurons or external output. It is this collective processing by the network that makes ANN a powerful computational device and able to learn from previous examples which are then generalized to future outcomes (see Zhang, Patuwo, & Hu, 1998; Hyndman & Athanasopoulos, 2014). A prototypical architecture of a multi-layer neural network system is depicted in Figure 1

ANN models have become popular in forecasting economic time series because of their ability to approximate a large class of functions with a high degree of accuracy (see Khashei & Bijari, 2010).[1] So that the usual problematic issues encountered in forecasting macroeconomic indicators; seasonality, nonstationarity, and nonlinearity are handled by

[1]Some recent examples of economic and financial applications of computational intelligence models and its performance competitions with other models include Giusto and Piger (2017); Qi (2001); Sokolov-Mladenović, Milovančević, Mladenović, and Alizamir (2016); Crone et al. (2011); Clements, Franses, and Swanson (2004); Heravi et al. (2004))

Figure 1: Topological structure of a feed-forward neural network



Note: A multi-layer perceptron with the first (input) layer reviving external information and being connected to the hidden layer through acyclic arcs which transmit signals to the last layer, outputting the solution (or forecast in this case).

this class of models (see Tseng, Yu, & Tzeng, 2002; Zhang & Qi, 2005). But more than that, the fact that the model is formed `intelligently` from the characteristics of the data, and hence, does not require any prior model specification, makes ANN suitable and appropriate for environments where theoretical guidance is not available, or is unreliable, to suggest the appropriate data generating process of an economic series.

We use the most common and basic structure of ANN models used in time series forecasting; see Zhang et al. (1998), Hippert, Pedreira, and Souza (2001), and De Gooijer and Hyndman (2006) for thorough surveys of this literature. In particular, we adopt a single hidden layer feed-forward network, characterized by a network of three layers of simple processing units (see Figure 1). The relationship between the outputs (y_t) and the inputs (x_t) of the model has the following form;

$$y_t = \omega_0 + \sum_{j=1}^{q} \omega_j \cdot \mathcal{S}\left(\omega_{0j} + \sum_{i=1}^{p} \Omega_{ij} \cdot x_{t-i}\right) + \epsilon_t, \tag{1}$$

where $\{\omega_j,\ j = 0, 1, \ldots, q\}$ and $\{\Omega_{ij},\ j = 0, 1, \ldots, q;\ i = 0, 1, \ldots p\}$ are the model parameters, which represent the connection weights. Specifically, ω_j represents the weights from the hidden to the output nodes, and Ω_{ij} denotes a matrix of parameters from the input nodes to the hidden-layer nodes. While p is the number of input nodes (or neurons) which are comparable to the number of predictor variables in a standard regression framework; q is the number of units in the hidden layer; $\mathcal{S}(\cdot)$ is the choice of the activation (transfer) function used; and ϵ is the error term.

The activation function determines the relationship between the inputs and outputs of a node and a network, and it is used to introduce nonlinearity in the model (Zhang et al., 1998). Although, according to Chen and Chen (1995), any differentiable function qualifies as an activation function. For application purposes, only a small number of "well behaved" functions are used.[2] Typically, the sigmoid (logistic) function, the hyperbolic tangent (tanh), the sine or cosine, and linear functions. For this study, we use the sigmoid function, depicted in Figure 2which is the most popular choice in forecasting environments (see Qi, 2001; Zhang & Qi, 2005). Thus,

$$\mathcal{S}(\chi) = \frac{1}{1 + \exp(-\chi)}. \tag{2}$$

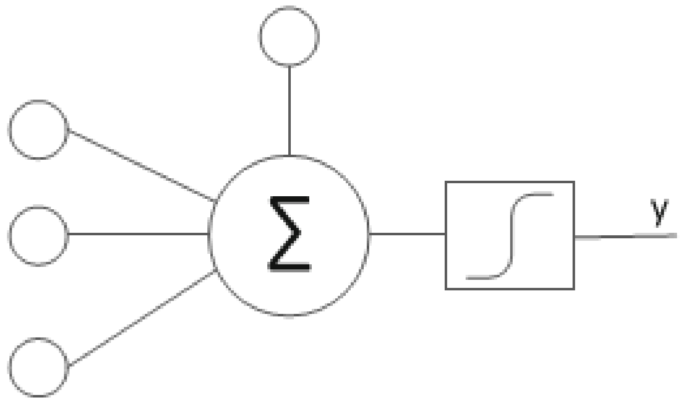
Because, in the literature, it is not clear whether different activation functions have major effects on the performance of the networks, we also experiment with the hyperbolic tangent function,

$$\mathcal{H}(\chi) = \frac{1 - \exp(-2\chi)}{1 + \exp(-2\chi)}. \tag{3}$$

The next item for the ANN modelling process is to choose the architecture of the model. This specifically involves choosing the five most important parameters of the model: the number of input nodes (predictor variables and their lags), the number of hidden layers, the number of hidden nodes, and the number of output nodes(variable to forecast). The choice of the architecture is the most important decision in a forecasting environment because it determines how successfully the model can detect the features, capture the pattern in the data, and perform complicated non-linear mappings from input to output variables (Zhang et al., 1998)

[2]By well behaved, it is typically supposed that the eligible class of functions should be continuous,

Figure 2: Structure of a Sigmoid (Logistic) neuron



We have followed the tradition in most forecasting applications (for examples, Tkacz (2001); Qi (2001); Zhang and Qi (2005); Kaytez, Taplamacioglu, Cam, and Hardalac (2015); Feng and Zhang (2014)), by using only one hidden layer with a small number of hidden nodes in our application. The reason is because although there are many approaches to selecting the optimal architecture of an ANN model, the process is often complex and difficult to implement. Moreover, none of the available methods can guarantee delivery of the optimal solution of parameters for all practical forecasting problems.[3] Furthermore, there is also theoretical evidence suggesting that single layer architectures can adequately approximate any complex linear function to any desired level of accuracy, and has better performance in terms of not overfitting models (see Zhang et al., 1998).

Once a decision has been taken on the architecture of the ANN (i.e., the number of predictors, the number of lags, and the hidden structure), the next step is to train the model using the data. Training involves a minimization process in which arc weights of a network are iteratively modified to minimize a criterion (often the mean square error) between the desired and actual output for all output nodes over all input patterns (Zhang

bounded, monotonically increasing and differentiable (Zhang, 2003).

[3]Typical approaches for selection of ANN model architecture are: (i) the empirical approach, which uses chooses parameters based on the performance of alternative models (Ma & Khorasani, 2003); (ii) Fuzzy inference methods, where the ANN is allowed to operate on fuzzy instead of real numbers (Leski & Czogała, 1999); (iii) pruning algorithms that respectively add or remove neurons from the initial architecture using a pre-defined criterion (Jain & Kumar, 2007; Jiang & Wah, 2003); and (iv) Evolutionary strategies that use genetic operators to search over the topological space by varying the number of hidden layers and hidden neurons (see a recent review in Khashei & Bijari, 2010).

et al., 1998). Again, although there exist many optimization methods to choose from, there is no algorithm that guarantees delivery of the global optimal solution in a reasonable amount of time; hence, we adopt the most popularly used optimization method which gives the "best" local optima (see Sałabun & Pietrzykowski, 2016; Zhang et al., 1998)

Specifically, we train the model using the Backpropagation (abbreviated from "backward propagation of errors") algorithm, which uses a gradient steepest descent method. Using this algorithm, the step size, which governs the learning rate of the model and the magnitude of weight changes, must be specified. To control for some of the known problems of the gradient descent algorithm, for examples, slow convergence and sensitivity to the choice of the learning rate, we follow Williams and Hinton (1986) by including an additional momentum parameter which allows for larger learning rates and ensures faster convergence in addition to its potential to dampen tendencies for oscillations.

The process of minimizing the errors using the BP algorithm involves comparing the result from the output layer with the desired result; if the errors exceed the threshold, then the value of the errors will be fed back to the inputs through the network, and the weights of nodes in each layer will be changed along the way until the error values are sufficiently small (see Lippmann, 1987; Feng & Zhang, 2014). To be more concrete, let m be the number of layers in the network, y_j^m represents the output from node j in layer m, $y_j^0 = x_j$ denotes the external input (stimulus) at node j, Ω_{ij}^m is the weight of the connection between node i in layer m and node j in layer $m+1$, and θ_j^m is the threshold at node j in layer m. Then, the iterative steps in the BP algorithm are as follows.

Step 1. **Initialize weights:** Initialize all weights and thresholds to a small random value. Typically, $\omega^0 \in (-1, 1); \quad \Omega_{ij}^0 \in (-1, 1)$

Step 2. **Present input and desired output:** Present the input vector $x_0, x_1, \ldots x_N$ and specify the desired outputs $d_0, d_1, \ldots d_N$. Note that the input vector could be new on each trial or typically, samples from a training set which are presented cyclically until weights stabilize.

Step 3. **Calculate actual outputs (forecasts):** Feed the signal forward and use the

Sigmoid function to calculate outputs $y_0, y_1, \ldots y_N$. That is, calculate

$$\hat{y}_j^m = S\left(\chi_j^m\right) = \mathcal{S}\left(\sum_i \Omega_{ij}^m \cdot y_i^{m-1} + \theta_j^m\right), \tag{4}$$

which involves processing the output at each node j from the first layer through the last layer until it completes the network.

Step 4. **Calculate errors in output:** Calculate the error for each node j in the output layer as follows;

$$\delta_j^m = \hat{y}_j^m\left(1 - \hat{y}_j^m\right)\left(d_j - \hat{y}_j^m\right) \tag{5}$$

where the error is the difference between the computed output and the desired target output.

Step 5. **Calculate errors in hidden nodes:** Calculate the error for each node j in the hidden layer as follows;

$$\delta_j^{m-1} = \mathcal{S}'\left(\chi_j^{m-1}\right)\sum_i \Omega_{ij} \cdot \delta_i^m, \tag{6}$$

which describes the process of feeding back errors layer by layer.

Step 6. **Update the weights and thresholds:** Using a recursive algorithm starting at the output nodes and working backwards, adjust the weights and thresholds as follows;

$$\Omega_{ij}^m(t+1) = \Omega_{ij}^m(t) + \eta\delta_j^m\hat{y}_i^{m-1} + \alpha\left[\Omega_{ij}^m(t) - \Omega_{ij}^m(t-1)\right] \tag{7}$$

$$\theta_j^m(t+1) = \theta_j^m(t) + \eta\delta_j^m + \alpha\left[\theta_j^m(t) - \theta_j^m(t-1)\right] \tag{8}$$

where $\alpha \in (0,1)$ is the momentum parameters, $\eta \in (0,1)$ is the learning rate, and t is the iteration counter.

Step 7. **Loop until convergence:** Go back to Step 2 and repeat the iteration up to Step 6 until the network error is sufficiently small. That is, to minimize

$$E = \frac{1}{N}\sum_{n=1}^{N}(\epsilon_i)^2\,, \tag{9}$$

or more precisely,

$$\frac{1}{N}\sum_{n=1}^{N}\left(y_t - \left[\omega_0 + \sum_{j=1}^{q}\omega_j \cdot \mathcal{S}\left(\omega_{0j} + \sum_{i=1}^{p}\Omega_{ij}\cdot x_{t-i}\right)^2\right]\right) \tag{10}$$

2.2 The ARIMA time series approach

The autoregressive integrated moving average (ARIMA) time series approach to forecasting has remained an attractive approach to economists because of its ability to use purely technical information (past values)— with no requirement for economic fundamentals and theory— to forecast economic time series (see the recent survey in De Gooijer & Hyndman, 2006). Moreover, its ability to parsimoniously handle stationary and non-stationeries series, typical features of economic variables, has helped to further entrench it in the discipline.

In an ARIMA model, the future values of a variable are modelled as a linear function of past observations and random errors. So that, the data generating process has the form;

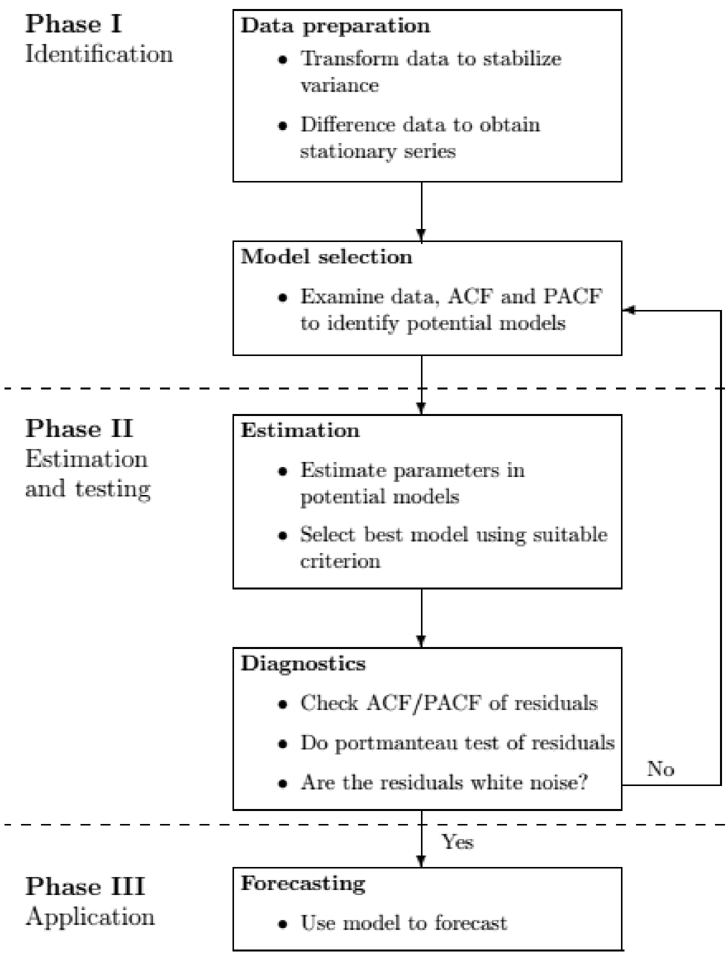
$$\begin{aligned} y_t =& \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \\ &+ \epsilon_t - \theta_1\epsilon_{t-1} - \theta_2\epsilon_{t-2} - \cdots - \theta_q\epsilon_{t-q}, \end{aligned} \tag{11}$$

where y_t are actual values of the variable, ϵ_t are the error terms which are assumed to be independently and identically distributed with mean zero and constant variance σ^2; $\{\phi_i, \quad i = 1, 2, \ldots, p\}$ and $\{\theta_j, \quad i = 0, 1, 2, \ldots, q\}$ are the model parameters, with p and q as integers indicating the order of the autoregressive (AR) and moving average (MA) terms, respectively.

Following the pioneer works of Yule (1926) and Wold (1938), Box and Jenkins (1976) developed a practical approach for time series analysis and forecasting now known as the Box-Jenkins ARIMA methodology (see Box, Jenkins, Reinsel, & Ljung, 2015; De Gooijer & Hyndman, 2006). The Box-Jenkins methodology involves three phases of iterative steps: (i)

Model identification phase (i.e., data preparation and model selection), (ii) Estimation and post estimation diagnosis phase, and (iii) Application phase. Figure 3, which is adapted from Makridakis, Wheelwright, and Hyndman (2008), depicts a schematic representation of the Box-Jenkins methodology for time series analysis.

Figure 3: Schematic representation of Box-Jenkins methodology



Source: Adapted from Makridakis et al. (2008)

In the identification phase, two main activities take place: (i) data preparation, and (ii) model selection. Data preparation is required because economic time series are often non-stationary and characterized by seasonality; therefore, data transformation is required to make the series stationary. Inducing stationarity ensures that the mean, variance, and autocorrelation structure of the series are not dependent on time. Whenever stationarity is lacking, it can be induced by applying differencing or power transformations techniques before an ARIMA model can be fitted. To select plausible models, Box and Jenkins (1976) show that if a time series is generated by an ARIMA, it should have some theoretical autocorrelation properties; and by matching the empirical autocorrelation properties with the theoretical autocorrelation properties using the autocorrelation function (ACF) and partial autocorrelation function (PACF) from sample data, one can identify the order of the ARIMA model (see Hyndman & Athanasopoulos, 2014; Zhang, 2003)

The second phase of the Box-Jenkins methodology involves the estimation of the model parameters. This involves a non-linear optimization procedure that seeks to minimize an overall measure of errors– typically by maximum likelihood estimation (MLE). After the model has been estimated, it remains to check for model adequacy; that is, whether the model assumptions about the errors, ϵ_t, are satisfied. This is typically done by plotting the ACF/PACF of the residuals, conducting the Portmanteau test among others statistics. If the model is not judged adequate, the researcher restarts the process from phase one, using information from the diagnostics tests to identify a more plausible model.

Once the researcher is satisfied that the most plausible model has been selected, the next step is to generate h-step forecasts of the variable y_t, where h is the forecast horizon. Three different, but related, types of forecasts can be considered: a point forecast, an interval forecast, and a density forecast. An appropriate loss function is used to determine the optimal h-step ahead point forecast. The interval forecasts consist of a lower and upper bound which contains the actual values with a certain probability, while the density forecast provides a complete characterization of the future observations y_{T+h}, which can be used to construct any sort of point or interval forecast (see Franses, Dijk, & Opschoor, 2014).

2.3 The structural econometric approach

In the structural econometric approach, economic theory is used to develop mathematical statements about how a set of observables (endogenous) variables, y, are related to another set of observables (explanatory) variables, x (Reiss & Wolak, 2007). Our structural econometric specification follows the style of the linear specification in Tkacz (2001). In particular, the structural econometric forecast equation for output growth is of the form

$$y_t = \alpha + \sum_{j=1}^{J} \beta \cdot X_j + \epsilon \tag{12}$$

where y is the endogenous variable, output growth, and $X_{j,t}$ are the explanatory variables. In order to obtain the best linear models from a broad search, the specification is broadened so that the explanatory variables in Eq. (12) are allowed to enter individually at levels and with various lag combinations between 1 and 4

2.4 Forecast performance measures

To ascertain the forecast performance of alternative models, or more precisely how well the forecasting models are able to reproduce the already known data, we follow the common practise of keeping P observations apart in order to use it to ascertain the the h-step ahead forecast $\hat{y}_{T+h+i|T+i} = \mathbb{E}[y_{T+h+i|T+i}]$ for $i = 0, \ldots, P-h$ from the models based on data from the first T observations. In particular, we split the data into two overlapping parts, the first 75 percent of the data series, T, are used for training and estimation purposes and the latter 75 percent of the series (or hold-out sample), P, are used for testing the forecast performance of the models.

There are several formal and informal ways to measure the forecasting performance of a model (see Hyndman & Athanasopoulos, 2014). The most popular ones are those based on the loss function upon which the model is based—i.e., the forecast errors. We use one of the most popular criterion in this category—the mean squared prediction error [MSPE], which can be computed as

$$\begin{aligned} MSE(h) =& \frac{1}{P-h+1} \sum_{P-h}^{t=0} (y_{T+h+i} - \hat{y}_{T+h+i|T+i})^2 \\ =& \frac{1}{P-h+1} \sum_{P-h}^{t=0} e^2_{T+h+i|T+i}, \end{aligned} \tag{13}$$

where P is the number of forecasts, h is the forecast horizon, here one-step ahead forecast, and e are the forecast errors.

Although the MSE helps to facilitate comparison of models for one country, it does not necessarily help in comparing the forecast performance of different models between countries having different scales of input variables as in our case. To facilitate comparison among models and between countries, we also consider an alternative standardized forecast performance measures relative performance—i.e., the mean absolute percentage error (MAPE) given as

$$MAPE(h) = \frac{100}{P-h+1} \sum_{P-h}^{t=0} \left| \frac{y_{T+h+i} - \hat{y}_{T+h+i|T+i}}{y_{T+h+i}} \right| \tag{14}$$

Based on the MSE and MAPE criteria, the model with the smallest value is the preferred model as it gives the most accurate forecast among the competing models.

3 Data and sample

Data is mostly retrieved from the IMF EcOS (IFS) database and supplemented with data from publications by country authorities. We use quarterly data from 1970 to 2016, a total of 188 observations, for three of the five biggest economies in terms of the size of GDP in sub-Saharan Africa: South Africa, Nigeria, and Kenya. These countries account for about 54 percent of the weight distribution used by the African Development Bank (AfDB) for computing growth forecasts in Africa. Hence, correctly forecasting growth in these economies is crucial for the accuracy of overall growth forecasts in sub-Saharan Africa.

The output forecast variable for each country is the growth rate of GDP, while the input variables vary by country. The differences in input variables for each country is designed to capture the structural characteristics of each economy, especially as it relates to the composition of its merchandise export. In particular, the input variables that are consistent across countries are the interest rates (the repo rate for South Africa, the discount rate for Nigeria, and the deposit rate for Kenya), the CPI inflation and the volume of trade. As for the input variable that varies across countries, we use the most country-relevant commodity price series because of the differences in the nature of commodity dependence among countries: gold prices for South Africa, crude oil prices for Nigeria, and coffee prices for Kenya.

The data series are divided into two overlapping periods; data from 1970Q1 until 2004Q4 are used for training and estimation, while our out-of-sample forecasts start from the first quarter of 1982 until last quarter in 2016.

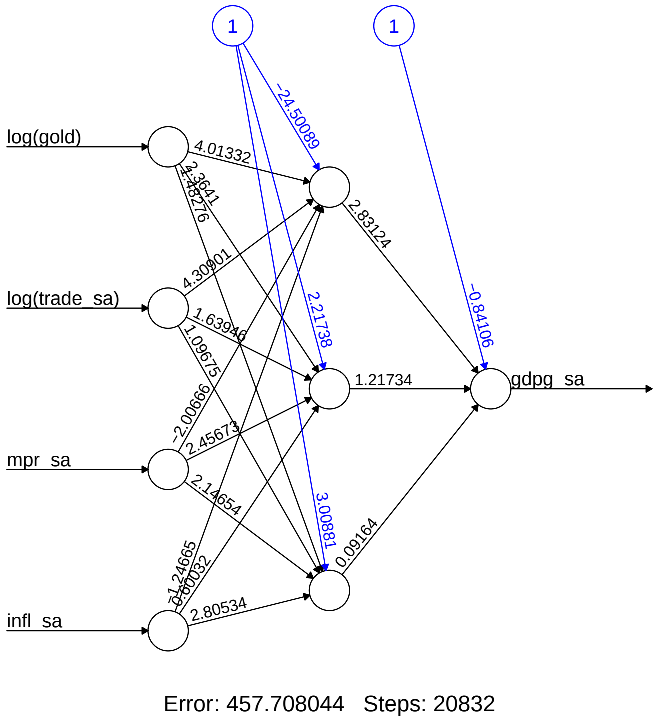
4 Results

In presenting and discussing the results from the forecasting exercise, we place more emphasis on the results from the artificial neural network forecasts and compare its performance with the traditional ARIMA and structural econometric model for each country. The neural network models are implemented in *R* using the "neuralnet" package by Günther and Fritsch (2016), the ARIMA models are also implemented in R using the "auto.arima" function in the "forecast" package, which implements a variation of the Hyndman and Khandakar algorithm combining unit root tests, minimization of the AICc and MLE to obtain the best ARIMA model (see the documentation in Hyndman & Khandakar, 2008; Hyndman & Athanasopoulos, 2014).

In Figures 4 to 6, we plot the trained neural network model for prediction of GDP growth rates in South Africa, Nigeria, and Kenya. They reflect the basic topology of the trained neural network. In particular, they are based on a single hidden layer feed-forward perceptron and three neurons for South Africa and Nigeria, and four neurons for Kenya. The logistic (sigmoid) activation function is used to relate the inputs to the outputs, and the synaptic weights are computed by resilient backpropagation with weight backtracking as in Riedmiller (1994). In Figures 4 to 6, the plots includes the trained synaptic weights, the black lines showing the connections between neurons in different layers; the bias term added to each step often interpreted as the intercept in a linear model, the blues lines; and the overall error and number of steps required to achieve convergence.

Because it is difficult to interpret the topology from a neural network, we are not able to put any structural interpretation on the weights and bias factors shown in Figures 4 to 6. Suffice it to state, however, that the training algorithms converge and therefore are

Figure 4: South Africa—trained neural network for GDP growth

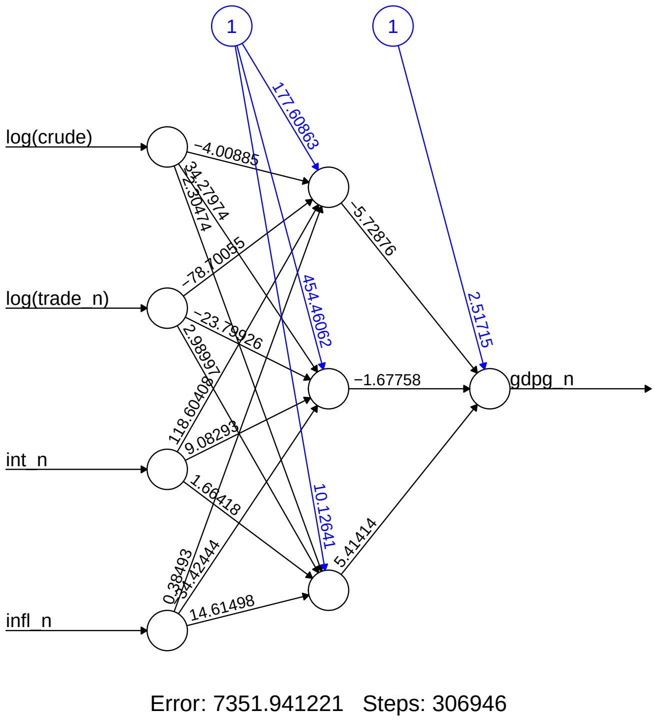


Note: Plot of trained neural network with a logistic activation function and synaptic weights computed by resilient backpropagation with weight backtracking as in Riedmiller (1994). The training process required 20832 steps until the absolute partial derivatives of the error function were smaller than 0.001

ready for use in forecasting. In particular, convergence is achieved after 20832, 306946, and 4182000 iterations in South Africa, Nigeria, and Kenya respectively. Although the error level for Nigeria is somewhat larger than those of South Africa and Kenya, the net for Kenya is particularly more complex as it required one additional neuron in the hidden layer (i.e. a total of four hidden neurons as opposed to three for the other countries) and many more iterations to achieve convergence

Interpretation of neural network models is usually done based on an examination of the effect of the input (or independent) variables on the prediction of the model. One approach is to study the effect of each input individually on each neuron in the network. This approach is problematic because it ignores the combined effect of one input on all units in the neural network. Therefore, to examine the relative importance of the inputs variables to the response variable in our neural network model, we follow the methodology in

Figure 5: Nigeria—trained neural network for GDP growth

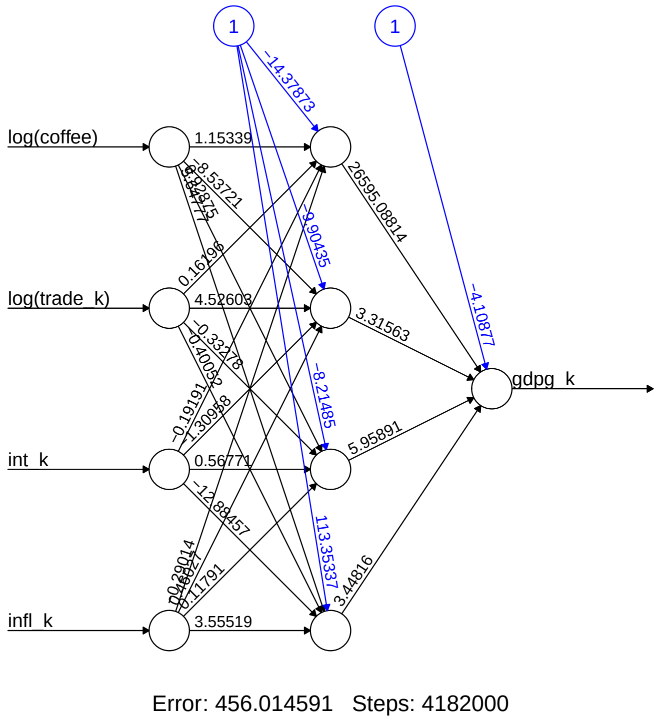


Note: Plot of trained neural network with a logistic activation function and synaptic weights computed by resilient backpropagation with weight backtracking as in Riedmiller (1994). The training process required 306946 steps until the absolute partial derivatives of the error function were smaller than 0.001

Intrator and Intrator (2001) which seeks to combine the effect of the inputs and the network architecture on all units in the network by studying the derivative of the prediction with respect to each covariate. This methodology involves the computation of the generalized weights which are then used to determine which variables have a linear effect, no effect, or a nonlinear effect on the predicted variable.

In Figures 7 to 9, we plot the generalized weights measuring the relative contribution of each covariate towards the prediction of the response variable, GDP growth. In particular, Panels 1 to 4 of Fig. 7 indicates that the price of gold, trade, interest rates and inflation rate are all important contributors to the prediction of GDP growth in South Africa. They are important because the generalized weights do not all cluster around point zero on the horizontal axis. Moreover, there is also evidence, though weak, to suggest that the effect of commodity prices, trade, interest rates, and inflation on GDP growth in South Africa is non-linear since the variance from zero of some generalized weights exceeds one.

Figure 6: Kenya—trained neural network for GDP growth

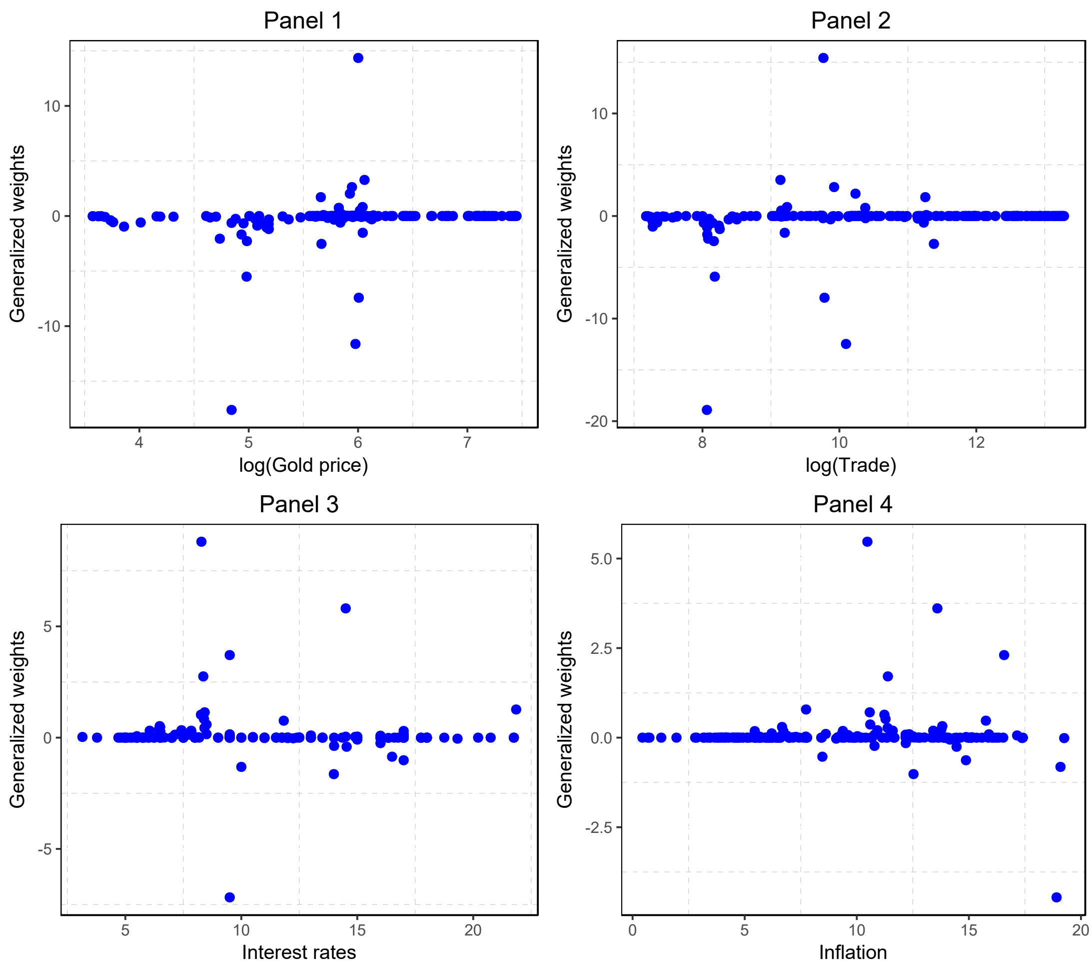


Note: Plot of trained neural network with a logistic activation function and with synaptic weights computed by resilient backpropagation with weight backtracking as in Riedmiller (1994). The training process required 4182000 steps until the absolute partial derivatives of the error function were smaller than 0.001

For Nigeria, the results are slightly different, apart from a few outliers, most of the generalized weights cluster around zero(see Panels 1 to 4 of Fig. 8), indicating that the impact of commodity prices, trade, interest rates, and inflation is mostly linear with weak evidence of significance. For Kenya, the results are quite fascinating, the generalized weights for the commodity price input and the trade input (Panels 1 and 2 of Fig. 9) are particularly almost evenly dispersed between -20 and 40 for commodity prices and -5 and 5 for the trade variable. The implication is that commodity prices and trade volumes are significant and nonlinear determinants of GDP growth in Kenya, the results for inflation in Panel 4 of Fig. 9 indicates that the influence of inflation is rather weak and nonlinear

The relative performance of the neural network model versus the linear and ARIMA models are assessed by visualizing the scatter plots of the respective model predictions versus the actual observed values. In Figures 10 to 12, we plot the actual versus model implied

Figure 7: South Africa—generalized weights for covariates

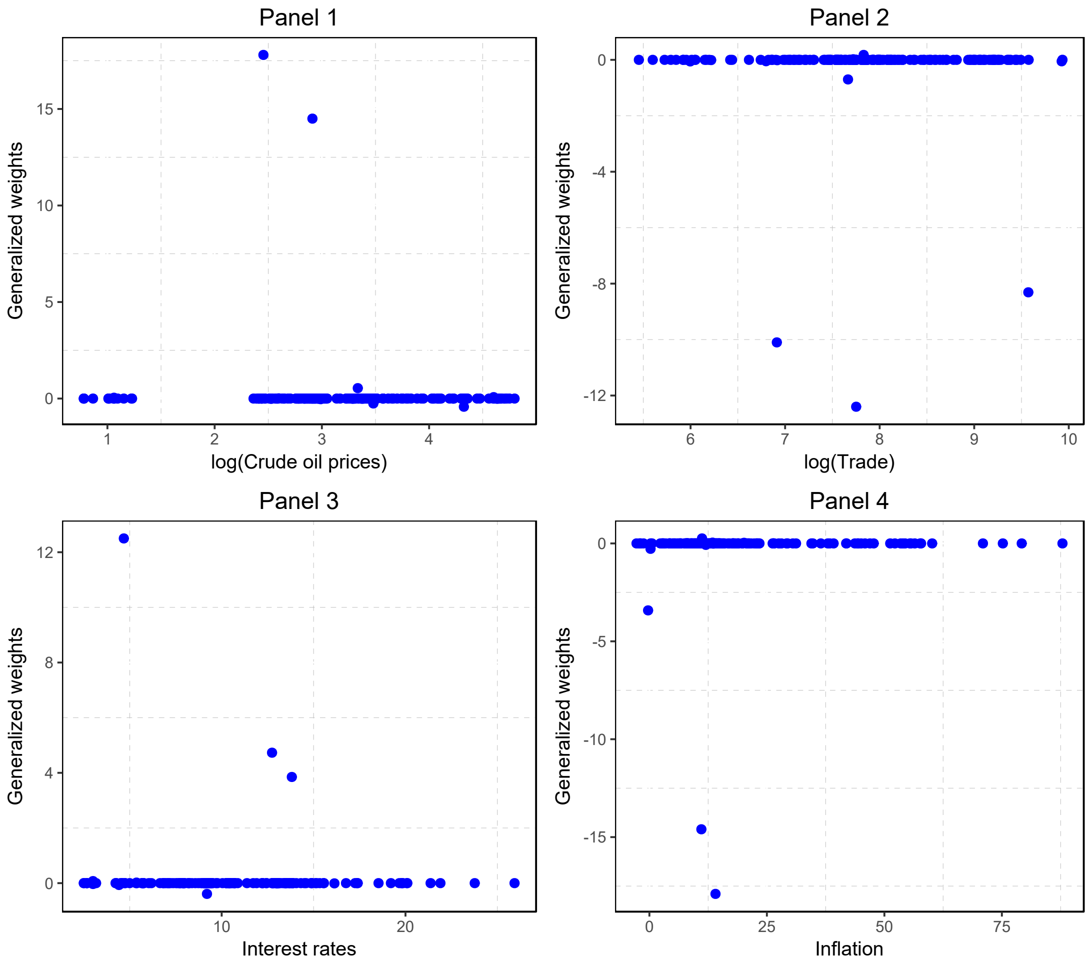


Note: A plot of the generalized weights for each covariate for the response variable GDP growth based on Intrator and Intrator (2001) which demonstrates the effect of each individual input on the response variable—GDP growth for South Africa.

forecasts for GDP growth in South Africa, Nigeria, and Kenya along with information on the line of best fit for the scatter plots. Figure 10 depicts the results for South Africa; Panels 1, 2, and 3 show the predictions from the neural network, the linear model, and the ARIMA model, where the selected optimal parsimonious model is an AR(3) and MA(2) combination with AICc 632.55; while Panel 4 is a plot of the scatter of all the predictions from the three models versus the actual values.

By visual inspection, we see that the predictions made by the neural network model (Panel 1 of Fig. 10) are, in general, more concentrated around the regression line of best fit than the results from other models. Where perfect alignment with the line of best fit would imply an MSE of 0, and thus an ideal perfect forecast. In particular, the slope of the

Figure 8: Nigeria—generalized weights for covariates

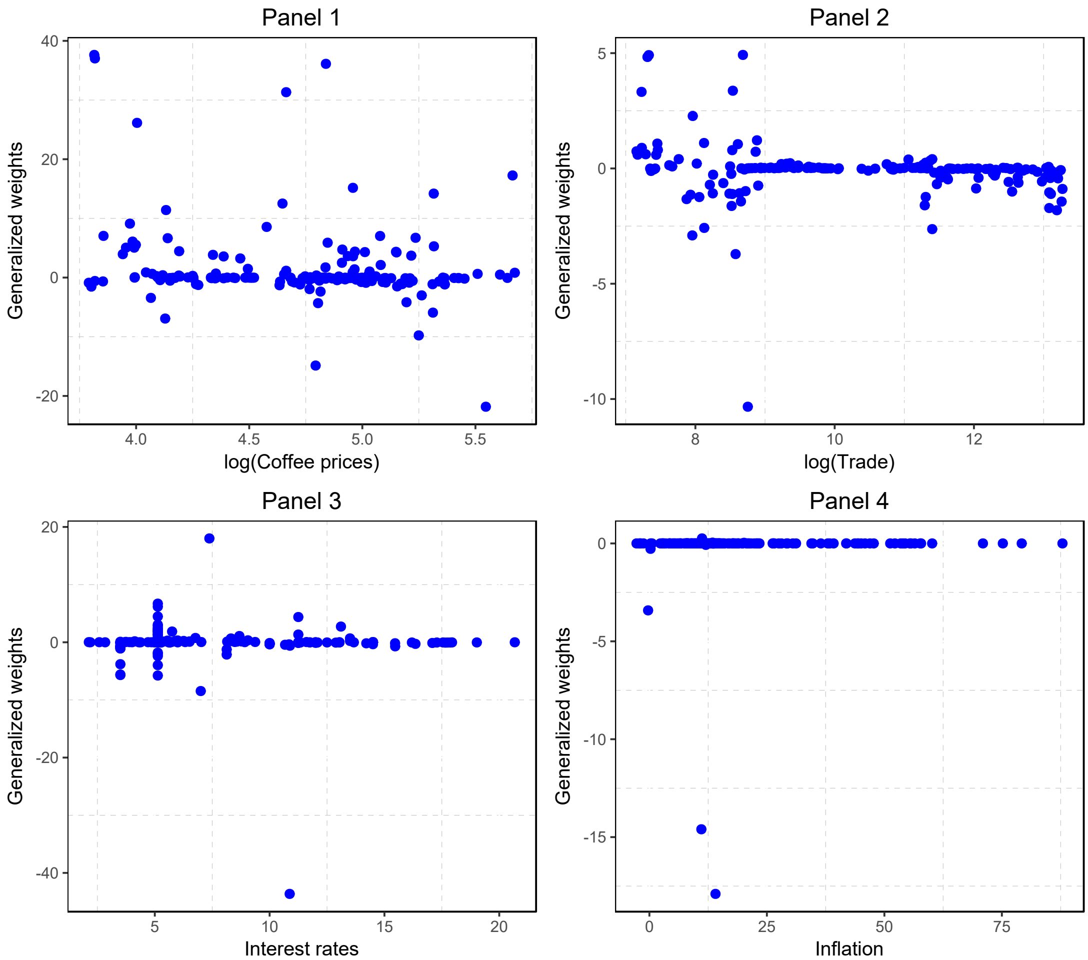


Note: A plot of the generalized weights for each covariate for the response variable GDP growth based on Intrator and Intrator (2001) which demonstrates the effect of each individual input on the response variable—GDP growth for Nigeria.

line of best fit for the ANN model is steepest at 1.67, compared to 1.65 in the linear model and 1.09 in the ARIMA model, and it also has the highest R-squared of 0.88, compared to 0.85 in the structural econometric model and 0.84 in the ARIMA model.

The results for South Africa show that although the neural network model improves the forecasting accuracy over the structural and ARIMA models, the improvement is only marginal. Specifically, in terms of improvements in the mean absolute percentage error, the ANN model only achieves a marginal 1.5 percentage points improvements over the structural model, and a 2.71 percentage points improvement over the ARIMA model (see Table 1). This is an indication that, for GDP growth forecasting in South Africa, although the structural and ARIMA model performs well in capturing the patterns of the data, the

Figure 9: Kenya—generalized weights for covariates

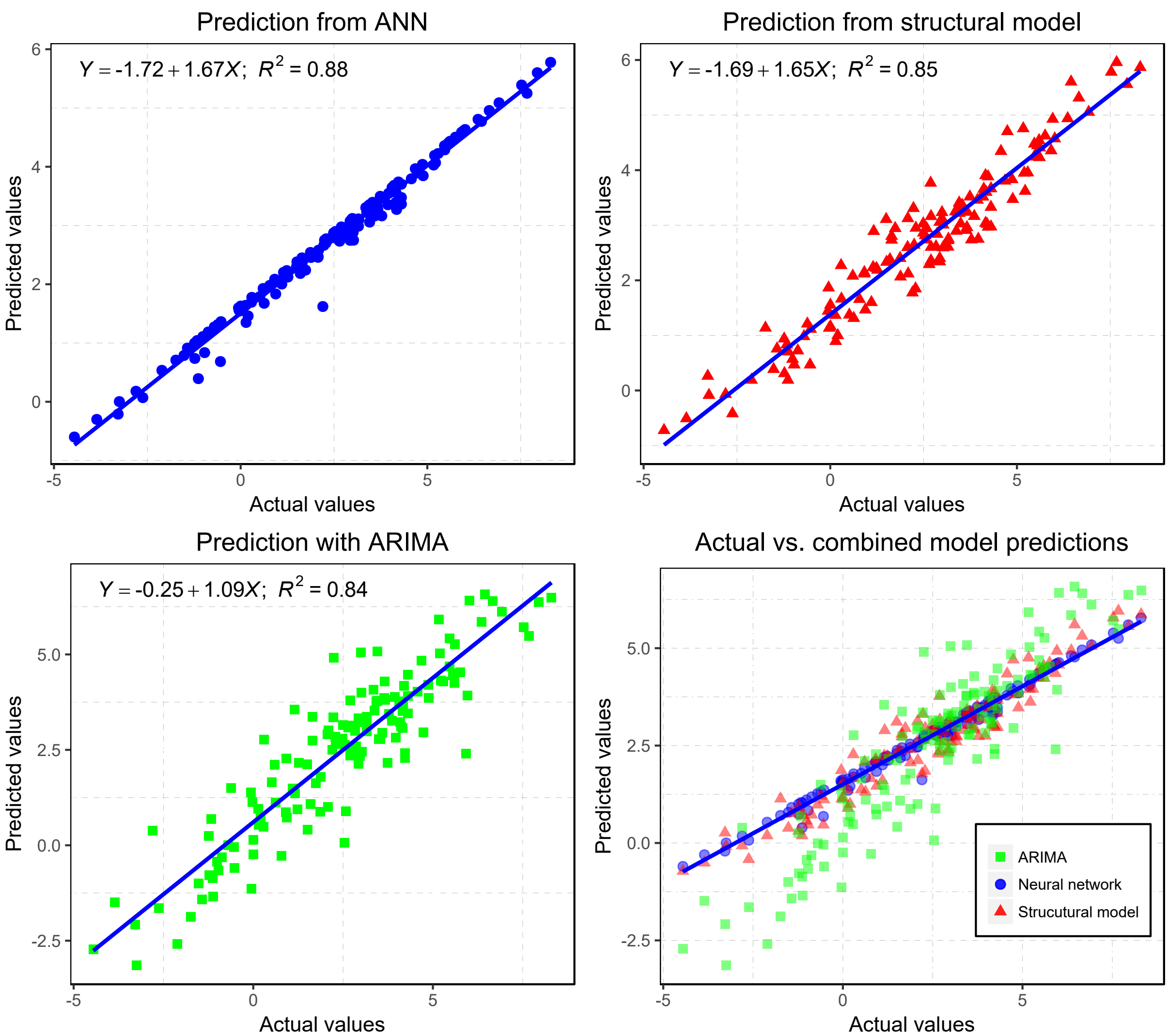


Note: A plot of the generalized weights for each covariate for the response variable GDP growth based on Intrator and Intrator (2001) which demonstrates the effect of each individual input on the response variable—GDP growth for Kenya .

neural network model delivers a much more precise forecast, especially when the inputs include commodity prices, trade, inflation and interest rates. The forecast performance measures—the mean square error (MSE), the mean absolute percentage error (MAPE), and the R-squared—presented in Table 1 for South Africa corroborate the conclusions from the visual inspection of Fig. 10. As the ANN model has the lowest MSE and MAPE followed by the structural model and the ARIMA model.

For Nigeria, although the ranking of the forecast performance of different models is similar to that of South Africa, there are, however, some important differences. The selected ARIMA model is an AR(3) and MA(3) combination which returned an AICc of 1089.24. In Panel 1 of Figure 11, we observe that although the artificial neural network

Figure 10: South Africa—Actual versus model predictions

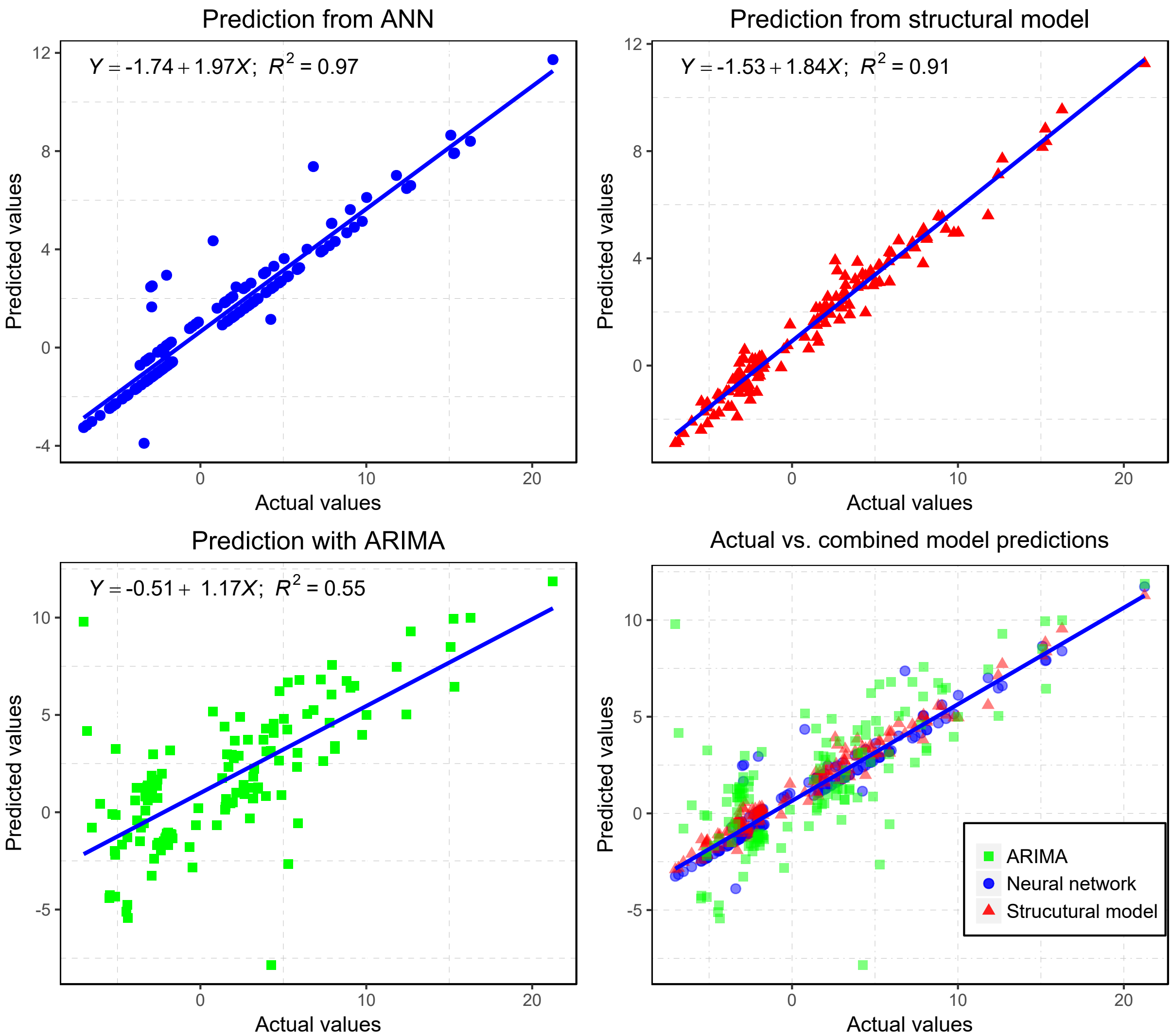


Note:

returns worse predictions for a few data points (outliers) its overall forecasting capability is superior to the structural and time series models. In particular, the R-squared for the ANN model is 0.97, much higher than the coefficient for the ARIMA model, 0.55, and the structural model 0.91.

The surprising thing about the performance of the ARIMA model in forecasting GDP growth in Nigeria is that it performs so poorly that it can hardly stand a chance in any competition with the ANN and structural model. In precise terms, the difference between the MSE for the ANN model of 0.93 and the ARIMA model of 19.78 is significant and using the MAPE criterion, the ARIMA model predictions are about 18 percentage points worse than the ANN predictions. For practical purposes, however, it seems most appropriate to

Figure 11: Nigeria—Actual versus model predictions

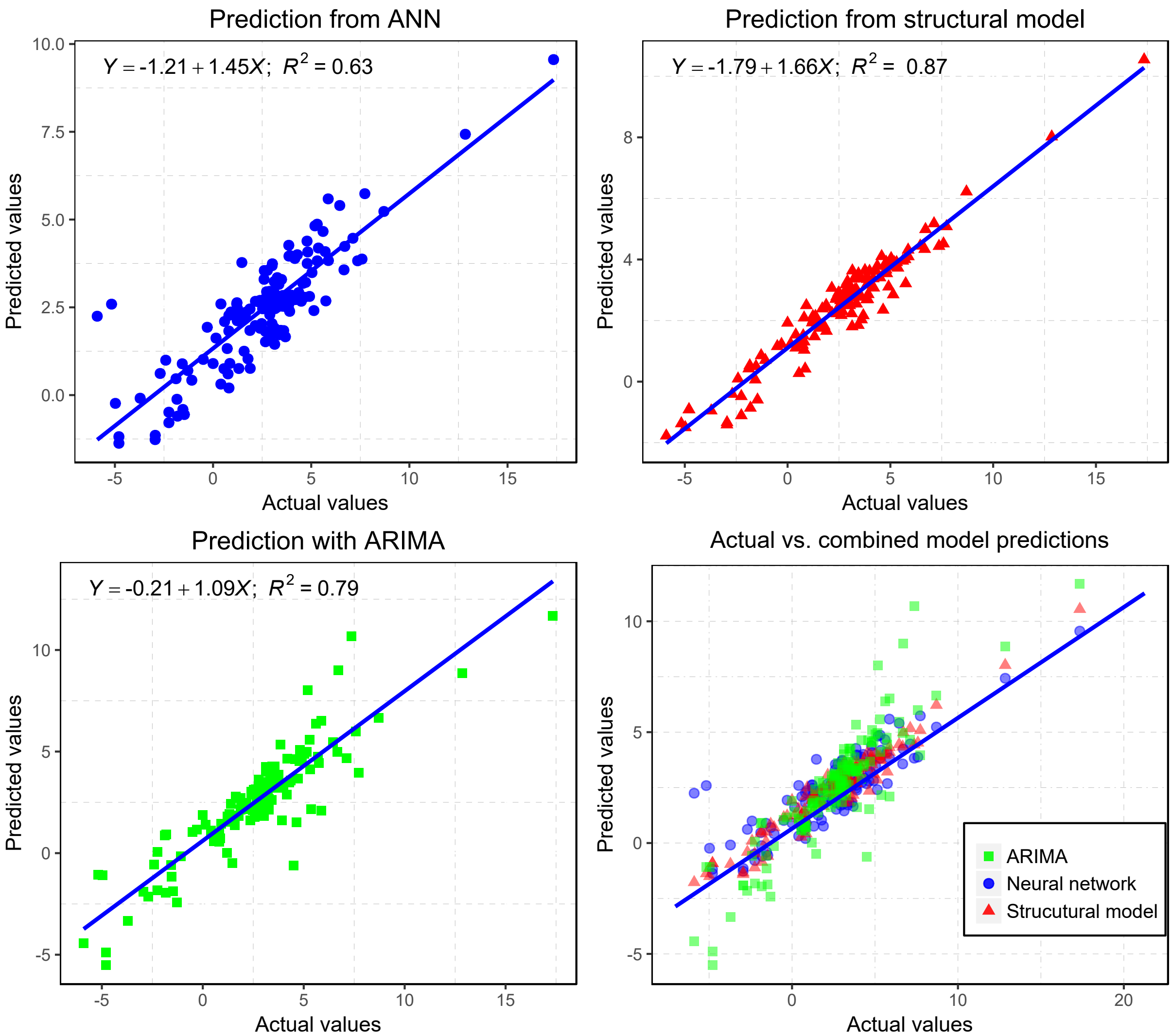


Note:

recommend the structural model for forecasting in Nigeria because of its overall regularity, instead of the ANN model with better performance statistics but occasional erratic and outlier predictions.

For Kenya, the results from the forecast competition are at stark difference with the others. Specifically, like the other countries, the neural network architecture for Kenya has four inputs, but unlike the other countries, it has four neurons in the hidden layer. The selected optimal parsimonious ARIMA has an AR(3) and MA(2) specification with an AICc of 741.09. From the visual inspection of Figure 12, we observe that the ANN model performs poorly when compared to the performance of the structural econometric model and the ARIMA model.

Figure 12: Kenya—Actual versus model predictions



Note:

In general, the ranking of the forecast performance of the three models for Kenya is as follows: first—structural econometric model, second—ARIMA model, and third—artificial neural network model with their receptive R-squared coefficients being 0.87, 0.79 and 0.63. It is important to notice from Panels 1 and 3 of Fig. 12 that even though the performance measures seem to favour the ARIMA model over the ANN model, there is a sense in which it would be better to use the ANN model over the ARIMA model for practical applications; because for selected data points, the ARIMA model gives predictions that are significantly different from the actual values.

Overall, the results show that artificial neural network systems perform slightly better in forecasting GDP growth in developing economies that depend highly on primary commodity

Table 1: Forecast performance comparison: ANN, Structural model and ARIMA

	South Africa			Nigeria			Kenya		
	MSE	MAPE	R^2	MSE	MAPE	R^2	MSE	MAPE	R^2
ARIMA	0.971	37.22	0.84	19.784	74.815	0.55	1.849	49.343	0.79
Structural model	0.928	36.01	0.85	3.623	58.538	0.91	1.187	36.731	0.87
Neural network	0.725	34.51	0.88	0.937	56.182	0.97	3.334	61.589	0.63

exports with its attendant price volatility. The reason that neural network models perform better than ARIMA models and some structural econometric models is most probably because it is able to capture the non-linear and chaotic behaviour of major input variables like commodity prices, economic instability and interest rates for forecasting the response variable—GDP growth—in these kinds of environment.

5 Conclusion

How to improve forecasts accuracy in an economic environment with potential for sudden change and volatility often inherited from input variables is an important question that forecasters of economic and financial variables in developing economies are faced with. This study examines the relative performance of artificial neural networks versus traditional time series models and structural econometric models in forecasting GDP growth for selected frontier economies in Africa, using quarterly data from 1970 until 2016.

Using absolute and relative forecast performance measures, the results show that artificial neural networks, in many cases, perform better than structural econometric models and ARIMA models. Our results are somewhat consistent with the results for developed economies obtained in, for examples, Tkacz (2001), Qi (2001), and Heravi et al. (2004). At least, three practical implications can be drawn from the results. First, parsimonious artificial neural networks models can be used to provide benchmark forecasts for GDP growth rates in developing economies that are exposed to potential chaotic influences from commodity prices, external factors, and even political economy factor because this class of model is better able to learn the system and capture the nonlinearities inherent in the input variables. Second, neural networks systems are also capable of misleading and producing outlier forecasts at certain data points, as we see in the cases of Nigeria and Kenya. Therefore, we recommended that forecasts from neural network models should be revalidated with forecasts from a structural econometric model. And finally, time series ARIMA models should only be considered as a last resort for forecasting in these kinds of environment, as they perform poorly mainly because of the sudden changes and chaotic patterns of macroeconomic variables in developing economies.

References

Box, G. E., & Jenkins, G. M. (1976). *Time series analysis: Forecasting and control* (Vol. 1). San Francisco: Holden-Day.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, *6*(4), 911–917.

Clements, M. P., Franses, P. H., & Swanson, N. R. (2004). Forecasting economic and financial time-series with non-linear models. *International Journal of Forecasting*, *20*(2), 169–183.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? empirical evidence from the nn3 competition on time series prediction. *International Journal of Forecasting*, *27*(3), 635–660.

De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International journal of forecasting*, *22*(3), 443–473.

Feng, L., & Zhang, J. (2014). Application of artificial neural networks in tendency forecasting of economic growth. *Economic Modelling*, *40*, 76–80.

Franses, P. H., Dijk, D. v., & Opschoor, A. (2014). *Time series models for business and economic forecasting*. Cambridge university press.

Ghiassi, M., Saidane, H., & Zimbra, D. (2005). A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting*, *21*(2), 341–362.

Giusto, A., & Piger, J. (2017). Identifying business cycle turning points in real time with vector quantization. *International Journal of Forecasting*, *33*(1), 174 - 184. Retrieved from http://www.sciencedirect.com/science/article/pii/S0169207016300590 doi: http://dx.doi.org/10.1016/j.ijforecast.2016.04.006

Günther, F., & Fritsch, S. (2016). neuralnet: Training of neural networks. *The R Journal*, *2*(1), 30–38.

Heravi, S., Osborn, D. R., & Birchenhall, C. (2004). Linear versus neural network forecasts for european industrial production series. *International Journal of Forecasting*, *20*(3), 435–446.

Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on power systems*, *16*(1), 44–55.

Hyndman, R. J., & Athanasopoulos, G. (2014). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, *27*(i03).

Intrator, O., & Intrator, N. (2001). Interpreting neural-network results: a simulation study. *Computational statistics & data analysis*, *37*(3), 373–393.

Jain, A., & Kumar, A. M. (2007). Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, *7*(2), 585–592.

Jiang, X., & Wah, A. H. K. S. (2003). Constructing and training feed-forward neural networks for pattern classification. *Pattern recognition*, *36*(4), 853–867.

Kaytez, F., Taplamacioglu, M. C., Cam, E., & Hardalac, F. (2015). Forecasting electricity consumption: a comparison of regression analysis, neural networks and least squares support vector machines. *International Journal of Electrical Power & Energy Systems*, *67*, 431–438.

Khashei, M., & Bijari, M. (2010). An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications*, *37*(1), 479–489.

Leski, J., & Czogała, E. (1999). A new artificial neural network based fuzzy inference system with moving consequents in if–then rules and selected applications. *Fuzzy Sets and Systems*, *108*(3), 289–297.

Lippmann, R. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, *4*(2), 4–22.

Ma, L., & Khorasani, K. (2003). A new strategy for adaptively constructing multilayer feedforward neural networks. *Neurocomputing*, *51*, 361–385.

Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (2008). *Forecasting methods and applications*. John Wiley & Sons.

Qi, M. (2001). Predicting us recessions with leading indicators via neural network models. *International Journal of Forecasting*, *17*(3), 383–401.

Reiss, P. C., & Wolak, F. A. (2007). Structural econometric modeling: Rationales and examples from industrial organization. *Handbook of econometrics*, *6*, 4277–4415.

Riedmiller, M. (1994). Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms. *Computer Standards & Interfaces*, *16*(3), 265–278.

Sałabun, W., & Pietrzykowski, M. (2016). Neural networks in economic problems. In *Selected issues in experimental economics* (pp. 245–266). Springer.

Sokolov-Mladenović, S., Milovančević, M., Mladenović, I., & Alizamir, M. (2016). Economic growth forecasting by artificial neural network with extreme learning machine based on trade, import and export parameters. *Computers in Human Behavior*, *65*, 43–45.

Teräsvirta, T., Van Dijk, D., & Medeiros, M. C. (2005). Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *International Journal of Forecasting*, *21*(4), 755–774.

Tkacz, G. (2001). Neural network forecasting of canadian gdp growth. *International Journal of Forecasting*, *17*(1), 57–69.

Tseng, F.-M., Yu, H.-C., & Tzeng, G.-H. (2002). Combining neural network model with seasonal time series arima model. *Technological Forecasting and Social Change*, *69*(1), 71–87.

Williams, D., & Hinton, G. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Wold, H. (1938). *A study in the analysis of stationary time series*. Almgrist and Wiksell, Stockholm.

Yule, G. U. (1926). Why do we sometimes get nonsense-correlations between time-series?–a study in sampling and the nature of time-series. *Journal of the royal statistical society*, *89*(1), 1–63.

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175.

Zhang, G. P., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, *14*(1), 35–62.

Zhang, G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European journal of operational research*, *160*(2), 501–514.