Comprehensive Review of Nonstationary Seasonal Timeseries Models through Python Simulations

Venkata Krishna Rao Maddela
Dept.of Computer Science & Engineering (Data Science)
Institute of Aeronautical Engineering
Hyderabad, India

***Abstract*— Timeseries analysis has been a popular bundle of techniques for analysis and prediction of a parameter that evolves over time. Simple to complex models have been considered for huge number of real-world applications. The challenges in this analysis are the timeseries attributes such as nonstationarity, seasonality, cyclic nature and trend. Several software tools were developed to carry out timeseries analysis. Several statistical tests were proposed to validate the identified model. The fitted model on real time data has no ground truth available for true validation except for statistical statements based on some Hypothesis. This prompts for an analysis applied on simulated timeseries data with known characteristics. Despite the abundant literature available on the analysis and prediction of timeseries, the information is widely scattered and needs a comprehensive review of the same. This paper derives an explicit expression for simulating a nonstationary seasonal timeseries. Using this expression the data is simulated using various timeseries models using open-source python language. The performance of models is evaluated using statistical tests. Thus, this review is not just a theoretical one but also includes independent simulation experiments. Moreover, except for few statistical tests, the python code is highly customized and developed from fundamentals without much built-in packages, leading to more authentic review of the topic through lot of transparency in the code functionality.**

Keywords*— *nonstationarity tests, integrated order, seasonality, SARMA, SARIMA, timeseries models, python programming.

I. INTRODUCTION

A time series is the time evolution of a parameter under observation. Examples include stock market prices, product sales [1], web traffic [2,3], electrocardiogram, electroencephalogram [4], weather profiles and many others. With the advent of new systems and advances in the existing systems such as cognitive radio, distributed sensors, internet of things (IOT) applications, unmanned vehicles, and electronic warfare, the amount of time series data is exploding in recent times. Speech is a natural timeseries and time evolution of phonetic-acoustics events play an important role in speech-based applications [5, 6]. In phonetic-acoustic analysis time evolution of short-time Fourier spectrum was also studied [7, 8]. In view of a data scientist or machine learning engineer, the time series is considered as an *unstructured data.* A timeseries is said to be nonstationary if its statistics vary over time. A timeseries is said to be seasonal if the data has a repetitive component with a specific period. If the period is not the same, then the series is said to be cyclic. Nonseasonal stationary timeseries modelling is a bit straightforward compared to nonstationary and seasonal series. [9]. In majority cases, exact seasonal period is not known. Several tests have been devised for testing the nonstationarity of the time series [10].

A seasonal nonstationary time series model is denoted as $ARIMA\,(p,d,q)(P,D,Q)_m$ and is mathematically given by

$$\emptyset(B)\emptyset_s(B)(1-B)^d(1-B)^D x_n = \Theta(B)\Theta_s(B)e_n \qquad (1)$$

where,
x is the time series,
e is the gaussian white innovation process $N(0,\sigma^2)$
p is the degree of nonseasonal AR polynomial
q is the degree of nonseasonal MA polynomial
d is the nonseasonal integrated order
P is the degree of seasonal AR polynomial
Q is the degree of seasonal MA polynomial
D is the seasonal integrated order
B is the backshift operator; $Bx_n = x_{n-1}, B^2x_n = x_{n-2}$ and so on.
$\emptyset(B) = 1 - \varphi_1 B - \varphi_2 B^2, \dots, -\varphi_p B^p$ is nonseasonal AR polynomial
$\Theta(B) = 1 + \theta_1 B + \theta_2 B^2, \dots, +\theta_q B^q$ is nonseasonal MA polynomial
$\emptyset_s(B) = 1 - \varphi_{1s} B^m - \varphi_{2s} B^{2m}, \dots, -\varphi_{Ps} B^{Pm}$ is seasonal AR polynomial
$\Theta_s(B) = 1 + \theta_{1s} B^m + \theta_{2s} B^{2m}, \dots, +\theta_{Qs} B^{Qm}$ is seasonal MA polynomial

Several tools and programming languages are available to model the real world complex timeseries data. The validation of these models is based on statistical tests in the absence of ground truth. Thus, for educative purposes or for validating a new algorithm, simulated data is more useful.

The purpose of this study is to review the nonstationary seasonal timeseries models in terms of generation and prediction. Here timeseries data with different model parameters *p, d, q, P, D, Q* is generated, and processing is carried out on the simulated data, review of timeseries models is carried out using illustrations of results.

The rest of the paper is organized as follows. Section II presents the recent developments in the time series analysis. In Section III a generic recursive expression for simulating seasonal and nonstationary timeseries is derived from fundamentals. The proposed algorithms for reviewing the nonstationary seasonal timeseries and the details of the

simulated dataset are also discussed in this section. Section IV discussed the analysis results with the help of illustrations, screenshots, tables and graphs. Finally, the conclusions are drawn, and future scope of work is presented in section V.

II. LITERATUTRE REVIEW

In [11] linear ARIMA and Holt's model were studied among other non-linear methods like neural network auto-regressive model. The system was aimed at arriving the best model to predict the risk of deaths and infections. The second task is the implementation of the third wave of infections and deaths in Russia. The study prompted about the limitations of study during time-changing conditions, and state that time-series forecasting can be accurate only in the short term and suggests for nonlinear timeseries models. In [12] a study was done to model the monthly foot and mouth disease outbreaks in Thailand using fitting timeseries models especially SARIMA model. The results have shown the importance of seasonality in timeseries modelling. However, estimation of seasonal period is still a challenge and in general grid search is used. In [13] an ARMA (1,1) model and an ARIMA (1, d, 1) were fitted to annual mean temperature data of land and sea surface. The authors have analyzed the ACF, PACF, residual and residual Q-Q plots. The study states that ARMA (1,1) fits better than ARIMA (1,1,1) and ARIMA (1,2,0). It prompts the importance of selecting model order, which in turn prompts for the stationarity tests before carrying out any timeseries analysis.

In [14] an explorative study was performed on malaria data of Adamawa Region, Cameroon from 2018 to 2022. Statistical analysis includes rank statistics like median and quartiles. Dispersion analysis and shape analysis was based on third and fourth statistical moments i.e., Skewness and Kurtosis. The ACF and PACF were also explored. The study also used the ADF and KPSS tests for verifying the stationarity of data. In [15] the study aimed at analysing the data of skilled and certified construction workers data in the period 2013-2020 of Indonesia. The relationship between construction workers counts and absorption in construction industry was studied. The projection modelling was carried out using ARIMA. Forecasting of number skilled construction workers for years 2022-2025 for all island provinces of Indonesia are provided. In [16] the study discusses univariate and vector ARMA models and dynamic linear models in epilepsy research. Examples are provided using ambulatory electrocorticography data. in the R programming environment. The review study of [17] broadly includes linear and nonlinear timeseries methods in theory. Some analytical expressions for ARMA and GARCH models were presented. However, no simulations or experiments on real data were conducted to support the theory.

In a very recent paper [18], energy trade analysis of different countries was conducted by using ARIMA models and LSTM models. The analysis was done on the data from 2000 to 2020. Primary energy consumption per capita was forecasted for India, Germany and China. It was stated that the analysis works well for some countries and for some other countries, the results are not encouraging. It means the insights into the timeseries models are required to understand complex patterns especially seasonality and nonstationarity.

In another recent work [19], life expectancy of India, USA and China was conducted by using ARMA and ARIMA models. The analysis was based on the data from 2000 to 2015. Predictions were carried out for the years 2016-2025 for different countries and results were presented for India, USA and China. When ARMA model was fitted all forecasts showed downward trend after 2015. When ARMA after differencing the timeseries once or twice, the results have improved. However, seasonality was not considered in this study. Hence, the results were not encouraging for the countries where there are cyclic and seasonal changes. Hence this phone is not included in the ensemble studies.

In another very recent paper [20], a study was carried out on global health expenditures versus gross domestic product for almost 200 countries in 2000 and 2022. Forecasting health expenditure was carried out using AR, MA and ARMA models and using Long Short-Term Memory networks. Error measures such as Mean Absolute Error, Mean Squared Error and RMSE were used to estimate the model performance. Error plots were presented for different (p, q) values of ARMA model. The error plots were cyclic indicating a cyclic component that is left uncaptured by the model. The results of fitting LSTM methods were also presented. The LSTM model could capture the long-term trend and thus missing the local temporal variations. In fact, the Current Health Expenditure as a percentage curve is cyclic in nature with an upward trend.

In the present review study, the nonstationary seasonal timeseries models are explored. The unique feature of this review is that independent simulations are also carried out along with the theory. An explicit expression for simulating a generic nonstationary seasonal timeseries is derived from fundamentals. This expression is used to simulate the timeseries data of different model orders (*p, d, q, P, D, Q)* and model parameters: AR, MA, SAR, SMA using open-source python language. This avoided the problems associated with built-in methods such as version mismatch, deprecated methods, deprecated options, and so on, faced by the author several occasions of simulations. Moreover, except for few statistical tests, the python code is highly customized without using any python timeseries packages. This makes the review more authentic as there is a lot of transparency in the code functionality.

III. METHODOLOGY

In this section, a generic expression for simulating a nonstationary and seasonal time series is derived. We start with differencing the time series represented by the model in (1) d times and D times seasonally, we get a seasonal stationary series that is denoted as $ARMA\,(p,q)(P,Q)_m$ and is mathematically given by

$$\emptyset(B)\emptyset_s(B)x_n = \Theta(B)\Theta_s(B)e_n \tag{2}$$

The ARMA model in (2) can also be stated by a recursive equation as

$$\begin{aligned} x_n &= (\varphi_1 x_{n-1} + \varphi_2 x_{n-2}, \dots, +\varphi_p x_{n-p}) \\ &+(\varphi_{1s} x_{n-m} + \varphi_{2s} x_{n-2m}, \dots, +\varphi_{Ps} x_{n-Pm}) \\ &+e_n + (\theta_1 e_{n-1} + \theta_2 e_{n-2}, \dots, +\theta_q e_{n-q}) \\ &+(\theta_{1s} e_{n-m} + \theta_{2s} e_{n-2m}, \dots, +\theta_{Qs} e_{n-Qm}) \end{aligned} \quad (3)$$

Using the eq (3), we can generate an ARMA stationary timeseries samples. Now we derive a similar recursive expression for generating samples from an ARIMA nonstationary timeseries. By changing the subscript in eq (3), we get

$$\begin{aligned} x_{n-1} &= (\varphi_1 x_{n-2} + \varphi_2 x_{n-3}, \dots, +\varphi_p x_{n-p-1}) \\ &+(\varphi_{1s} x_{n-m-1} + \varphi_{2s} x_{n-2m-1}, \dots, +\varphi_{Ps} x_{n-Pm-1}) \\ &+e_{n-1} + (\theta_1 e_{n-2} + \theta_2 e_{n-3}, \dots, +\theta_q e_{n-q-1}) \\ &+(\theta_{1s} e_{n-m-1} + \theta_{2s} e_{n-2m-1}, \dots, +\theta_{Qs} e_{n-Qm-1}) \end{aligned} \quad (4)$$

Subtracting eq (4) from (3), we get

$$\begin{aligned} &x_n - x_{n-1} \\ &= (\varphi_1[x_{n-1} - x_{n-2}] + \varphi_2[x_{n-2} - x_{n-3}], \dots, \\ &+\varphi_p[x_{n-p} - x_{n-p-1}]) + (\varphi_{1s}[x_{n-m} - x_{n-m-1}] \\ &+\varphi_{2s}[x_{n-2m} - x_{n-2m-1}], \dots, +\varphi_{Ps}[x_{n-Pm} - x_{n-Pm-1}] \\ &+(e_n - e_{n-1}) + (\theta_1[e_{n-1} - e_{n-2}] + \theta_2[e_{n-2} - e_{n-3}], \dots, \\ &+\theta_q[e_{n-q} - e_{n-q-1}]) + (\theta_{1s}[e_{n-m} - e_{n-m-1}] \\ &+\theta_{2s}[e_{n-2m} - e_{n-2m-1}], \dots, +\theta_{Qs}[e_{n-Qm} - e_{n-Qm-1}]) \end{aligned} \quad (5)$$

Defining the new variables

$$\begin{aligned} y_n &= x_n - x_{n-1} \\ g_n &= e_n - e_{n-1} \end{aligned} \quad (6)$$

and substituting in eq (5) results in

$$\begin{aligned} y_n &= (\varphi_1 y_{n-1} + \varphi_2 y_{n-2}, \dots, +\varphi_p y_{n-p}) \\ &+(\varphi_{1s} y_{n-m} + \varphi_{2s} y_{n-2m}, \dots, +\varphi_{Ps} y_{n-Pm}) \\ &+g_n + (\theta_1 g_{n-1} + \theta_2 g_{n-2}, \dots, +\theta_q g_{n-q}) \\ &+(\theta_{1s} g_{n-m} + \theta_{2s} g_{n-2m}, \dots, +\theta_{Qs} g_{n-Qm}) \end{aligned} \quad (7)$$

$$\begin{aligned} &y_n - y_{n-1} = \\ &(\varphi_1[y_{n-1} - y_{n-2}] + \varphi_2[y_{n-2} - y_{n-3}], \dots, + \\ &\varphi_p[y_{n-p} - y_{n-p-1}]) + (\varphi_{1s}[y_{n-m} - y_{n-m-1}] + \\ &\varphi_{2s}[y_{n-2m} - y_{n-2m-1}], \dots, +\varphi_{Ps}[y_{n-Pm} - y_{n-Pm-1}] \\ &+(g_n - g_{n-1}) + (\theta_1[g_{n-1} - g_{n-2}] + \theta_2[g_{n-2} - g_{n-3}], \dots, \\ &+\theta_q[g_{n-q} - g_{n-q-1}]) + (\theta_{1s}[g_{n-m} - g_{n-m-1}] + \\ &\theta_{2s}[g_{n-2m} - g_{n-2m-1}], \dots, +\theta_{Qs}[g_{n-Qm} - g_{n-Qm-1}]) \end{aligned} \quad (8)$$

Again defining the new variables

$$\begin{aligned} z_n &= y_n - y_{n-1} \\ w_n &- g_n - g_{n-1} \end{aligned} \quad (9)$$

rearranging the terms and continuing in the same manner, and substituting back from eq (9) and (6), we have the compact expression in terms of original variable as

$$\begin{aligned} &x_n = \\ &2Bx_n - B^2 x_n + \Delta_2(B)\big(1 - \emptyset(B)\big) + \Delta_2(B)(1 - \emptyset_s(B)) \\ &+\Delta_2(B)\,\Theta(B)e_n + \Delta_2(B)\Theta_s(B)e_n \end{aligned} \quad (10)$$

where $\Delta_2(B) = 1 - 2B + B^2$ is differentiating polynomial of order d with coefficients (1, -2, 1) derived from $(2 \mid 1)^{th}$ row of pascal triangle. It may be noted that this corresponds to the polynomial $(1-B)^d$ for $d-2$ and seasonal terms correspond to $(1-B)^D$ for $D=2$. Here it is assumed that $d = D = 2$, however it will be true for the case of $d \neq D$ also. Continuing further to find d differences, and substituting for difference samples of original series, we can show that the stationary ARMA series using d-difference series, is equivalent to the original nonstationary ARIMA series, and we have a recursive equation to generate the series for any *d* and *D*.

A. Algorithm for Dataset generation

The dataset used for reviewing the SARIMA models in this work is simulated using eq (1). The algorithm to generate the data is given below.

1. Assume the zero mean innovation process and specify the variance of the process and *n* the number of observations to be genefrated.
2. Select the nonseasonal AR order and nonseasonal AR coefficients. Verify the stationarity of the selected AR process using pole-zero diagram.
3. Select the nonseasonal MA order and nonseasonal MA coefficients. Verify the invertibility of the selected MA process using pole-zero diagram.
4. Select the seasonal AR order and seasonal AR coefficients. Verify the stationarity of the selected AR process using pole-zero diagram.
5. Select the seasonal MA order and seasonal MA coefficients. Verify the invertibility of the selected MA process using pole-zero diagram.
6. Plug in the parameters into the eq (1) and iteratively compute the timeseries of length *n*.

B. Algorithm for Timeseries Analysis

The analysis algorithm is given below.

1. Display the timeseries as a line plot using *matplotlib* package of python. It is necessary to ascertain that valid data is generated.
2. Compute the autocorrelation and partial autocorrelations using the expressions in [1, 2, 3].
3. Display the autocorrelation and partial autocorrelations using *matplotlib* package to get gross idea about the nature of the data model (AR/MA/nonstationary/ seasonal).
4. Conduct the unit root tests: Augmented Dickey-Fuller (ADF)test, Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, and Phillips-Perron (PP) test to check for nonstationarity.
5. If timeseries is stationary, go to step 9, else go to step 6.
6. As the timeseries is nonstationary, estimate the nonseasonal integrated order *d*, using .ndiffs() method of pmdarima.arima.utils python module.
7. Estimate the seasonal integrated order *D* using .nsdiffs() method of pmdarima.arima.utils python module.

8. Difference the timeseries nonseasonally *d* times and seasonally *D* times to convert the series into a stationary series.
9. Estimate the ARMA model orders *(p,d,q)(P,D,Q)* using autoarima or any other time series method of python.
10. Fit the best model to the data, and compute the residual (i.e., the innovation/driving white noise process that is responsible for the given series.
11. Evaluate the residual for whiteness and normality using QQ plots, and moment measures.
12. If not satisfied with normality tests, go to step 9, else to step 13.
13. Predict the test data using the best fitted model, and compute error measures. If not satisfied go to step 9, else go for forecasting optionally.

C. Dataset

A highly customized python code without using any timeseries (ARIMA/ARMA) packages is developed by the author to generate any generalized *ARIMA(p, d, q)(P, D, Q, m)* process with a deterministic or stochastic trend, nonseasonal and/or seasonal ARMA, and integrated orders: 1 and 2. As per the experience of the author, there are some problems with built-in methods in simulating a Timeseries because of version mismatch, deprecated methods, deprecated options, and so on. The code can generate timeseries for any nonseasonal orders: (p, q) and any seasonal orders: (P,Q) and any season period (m). Simulations are carried out for wide varieties of models to verify the code authenticity. Here the results are presented for (i). nonseasonal model: $[\varphi_1,\varphi_2,\varphi_3] = [0.5, -0.3, 0.2]\,, \mathrm{p} = 3$ and $[\theta_1,\theta_2,\theta_3,\theta_4] = [0.35, -0.45, 0.2, 0.65], \mathrm{q} = 4$ (ii). Seasonal model: $\varphi_{1s} = [0.65], P = 1$ and $\theta_{1s} = [0.65], Q = 1$ (iii). nonseasonal integrated orders: d = 0, 1, 2 and (iv). seasonal integrated orders: D = 0, 1. The SARIMA model for this set of $\emptyset, \Theta, \emptyset_s, \text{and } \Theta_s$ are verified for invertibility and stationarity using built-in attributes of *.isinvertible* and *.isstationary* of *statsmodels.tsa.arima_process.ArmaProcess* class. The poles and zeros are also visualized using customized python code as shown in Fig 3. The poles and zeros are computed by using the built-in method *.roots()* of numpy package.

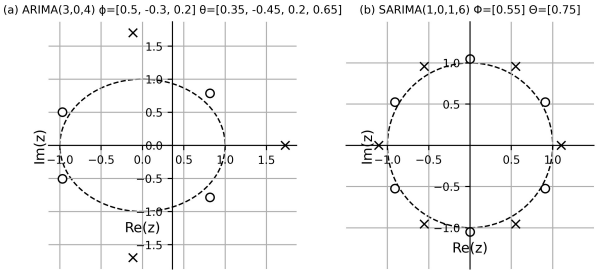

Fig. 1. Pole-Zero diagram of the ARIMA process used to simulate in ths paper. (a) Nonseasonal ARIMA (b). Seasonal ARIMA.

In Fig 3, the zeros are represented as 'o' and poles as 'x'. Fig 3(a) shows the roots of nonseasonal AR (poles) and MA (zeros) polynomials. Fig 3(b) shows the roots of seasonal AR (poles) and MA (zeros) polynomials. As all the roots are outside the unit circle in z-plane, the system is stable and invertible. Invertibility means the model can be estimated from the timeseries. Thus, it is worth noting down that random selection of AR and MA coefficients may not always result in a stable and predictable system.

IV. Simulation Experiments & Results

The spider IDE under Anaconda distribution is used for python programming. The stationary (d=0) and nonstationary (d=1,2) timeseries of $ARIMA\,(p,d,q)(P,D,Q)_m$ model for the orders given in Table I, II and III respectively are generated using the customized python code *myARIMAsimulate.py*.

TABLE I. DESCRIPTION OF SIMULATED TIMESERIES (STATIONARY)

Sl. No.	Model Specification	Model Name
1	ARIMA(3,0,0)(0,0,0)	3-AR model
2	ARIMA(0,0,4)(0,0,0)	3-MA model
3	ARIMA(3,0,4)(0,0,0)	(3,3)-ARMA model
4	ARIMA(0,0,0)(3,0,0)	3-SAR model
5	ARIMA(0,0,0)(0,0,3)	3-SMA model
6	ARIMA(0,0,0)(3,0,3)	(3,3)-SARMA model
7	ARIMA(3,0,0)(3,0,0)	(3)(3)-SAR model
8	ARIMA(0,0,4)(0,0,3)	(3)(3)-SARMA model
9	ARIMA(3,0,4)(3,0,3)	(3,3)(3,3)-SARMA model

TABLE II. DESCRIPTION OF SIMULATED TIMESERIES (NONSTATIONARY, INTEGRADED ORDER =1)

Sl. No.	Model Specification	Model Name
1	ARIMA(3,1,0)(0,0,0)	3-IAR model
2	ARIMA(0,1,4)(0,0,0)	3-IMA model
3	ARIMA(3,1,4)(0,0,0)	(3,3)-ARIMA model
4	ARIMA(0,0,0)(3,1,0)	3-SIAR model
5	ARIMA(0,0,0)(0,1,3)	3-SIMA model
6	ARIMA(0,0,0)(3,1,3)	(3,3)-SARIMA model
7	ARIMA(3,1,0)(3,1,0)	(3)(3)-SIAR model
8	ARIMA(0,1,4)(0,1,3)	(3)(3)-SARIMA model
9	ARIMA(3,1,4)(3,1,3)	(3,3)(3,3)-SARIMA model

TABLE III. SIMULATED TIMESERIES (NONSTATIONARY, INTEGRADED ORDER =2)

Sl. No.	Model Specification	Model Name
1	ARIMA(3,2,0)(0,0,0)	3-IAR model
2	ARIMA(0,2,4)(0,0,0)	3-IMA model
3	ARIMA(3,2,4)(0,0,0)	(3,3)-ARIMA model
4	ARIMA(0,0,0)(3,2,0)	3-SIAR model
5	ARIMA(0,0,0)(0,2,3)	3-SIMA model
6	ARIMA(0,0,0)(3,2,3)	(3,3)-SARIMA model
7	ARIMA(3,2,0)(3,2,0)	(3)(3)-SIAR model
8	ARIMA(0,2,4)(0,2,3)	(3)(3)-SARIMA model
9	ARIMA(3,2,4)(3,2,3)	(3,3)(3,3)-SARIMA model

The timestep is taken as 1 for all simulations. The line plots of the timeseries generated for the models given in Tables I, II and III are shown in Figs 2, 3 and 4 respectively. The autocorrelation (ACF) and partial autocorrelation (PACF) are computed for the above 9 x 3 = 27 timeseries are computed using customized python scripts: *myACF(X,maxlags)* and *myPACF(X, maxlags)* where the arguments are timeseries (X) and maximum lags for

correlation computation. The plots are shown in Figs 5, 6 and 7. It may be observed that the correlation plots peak at lag 6, which is the season period of the series. The partial autocorrelation up to 3 lags are significant for pure AR process. Similarly, the autocorrelations up to 4 lags are significant for pure MA process. Here the 3 and 4 are the orders of AR and MA parts respectively.

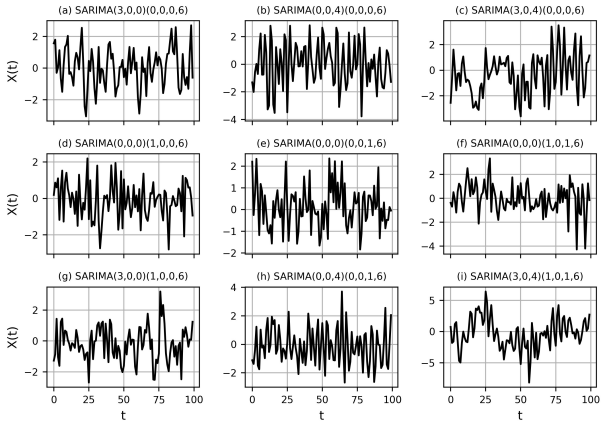

Fig. 2. Timeseries plots of first 100 samples of simulated data (d=0)

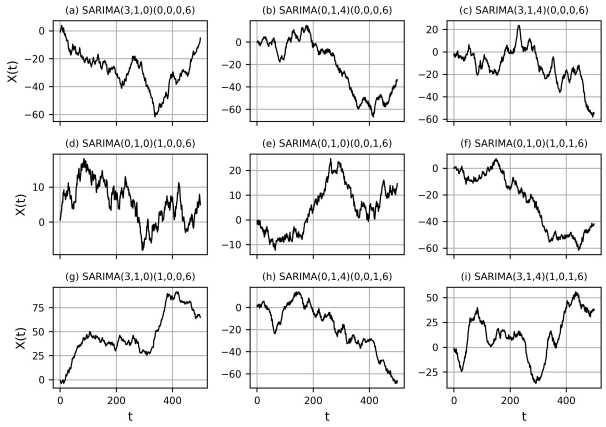

Fig. 3. Timeseries plots of first 500 samples of simulated data (d=1)

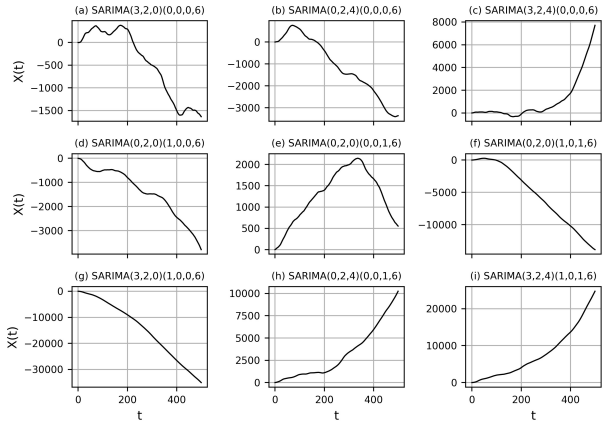

Fig. 4. Timeseries plots of first 500 samples of simulated data (d=2)

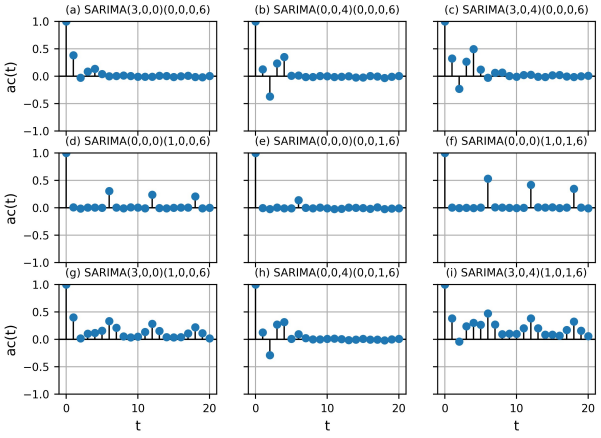

Fig. 5. Autocorrelations of 8000 samples of simulated data (d=0)

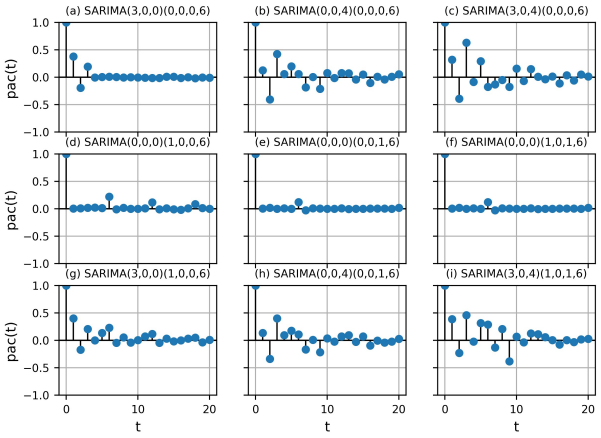

Fig. 6. Partial autocorrelations of 8000 samples of simulated data (d=0)

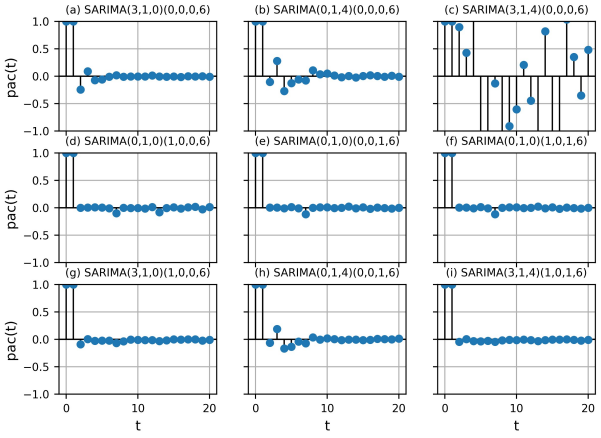

Fig. 7. Partial autocorrelations of 8000 samples of simulated data (d=1)

The nonstationarity of the timeseries is assessed by using ***ndiffs()*** method selecting the ADF, KPSS and PP tests separatelt. The results are shown in Fig 8. The d=1 or 2 are correctly estimated by all methods. This conforms to the theory as well as the customized code authenticity. The p-value for both ADF test as well as PP test for d=1,2 are

above the significant level $\alpha = 0.05$; this fails to reject the null hypothesis H_0 meaning that the series is nonstationary as expected.

```
In [3]: runfile('D:/00TimeSeriesModelling/myARIMAStationarityTests1.py', wdir='D:/
00TimeSeriesModelling')
Reloaded modules: myFunctions3
============================================
p(ADF) p(PP) d(ADF) d(KPSS) d(PP) D(ocsb) D(ch):
[0.1735, 0.59, 1, 1, 1, 0, 0]

In [4]: runfile('D:/00TimeSeriesModelling/myARIMAStationarityTests1.py', wdir='D:/
00TimeSeriesModelling')
Reloaded modules: myFunctions3
============================================
p(ADF) p(PP) d(ADF) d(KPSS) d(PP) D(ocsb) D(ch):
[0.485, 0.81, 2, 2, 2, 0, 0]
```

Fig. 8. Estimation of integrated order d of timeseries by ADF, KPSS and PP tests.

Then the builtin method of python: ***pmdarima.auto_arima(params)*** is used estimate the model orders (p,d,q)(P,D,Q) of a simulated nonseasonal nonsationary timeseries, where the argument ***params*** is the set of appropriate input parameters required. The screenshot of results is shown in Fig 9. Here grid search is used meaning all combinations of (p,d,q,P,D,Q) are searched for the identifying the best model. Total 172 cominations are searched minimizing the AIC cost. The best model is ARIMA(3,2,4)(0,0,0) with no seasonal component. The model parameters of best model are also close to true values.

```
Best model:  ARIMA(3,2,4)(0,0,0)[0] intercept
Total fit time: 37.178 seconds
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 4000
Model:               SARIMAX(3, 2, 4)   Log Likelihood               -5667.004
Date:                Sat, 19 Apr 2025   AIC                          11352.008
Time:                        21:35:41   BIC                          11408.650
Sample:                             0   HQIC                         11372.086
                                 - 4000
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      0.0694      0.028      2.456      0.014       0.014       0.125
ar.L1          0.4995      0.020     24.396      0.000       0.459       0.540
ar.L2         -0.3070      0.021    -14.508      0.000      -0.348      -0.266
ar.L3          0.2043      0.020      9.989      0.000       0.164       0.244
ma.L1          0.3580      0.016     22.727      0.000       0.327       0.389
ma.L2         -0.4519      0.018    -25.701      0.000      -0.486      -0.417
ma.L3          0.2045      0.018     11.422      0.000       0.169       0.240
ma.L4          0.6564      0.015     44.103      0.000       0.627       0.686
sigma2         0.9960      0.022     44.852      0.000       0.952       1.039
==============================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                 1.27
Prob(Q):                              0.89   Prob(JB):                         0.53
Heteroskedasticity (H):               1.01   Skew:                             0.04
Prob(H) (two-sided):                  0.88   Kurtosis:                         3.05
==============================================================================
```

Fig. 9. Final output of applying *autoarima()* method on a simulated nonseasonal nonsationary timeseries.

The AIC loss is also plotted for 172 combinations of orders in Fig 10. Please observe the decreasing loss pattern until th ebest order is obtained.

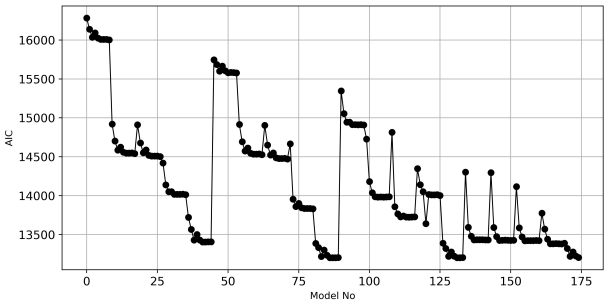

Fig. 10. AIC loss is also plotted for 172 combinations of orders.

The result of applying ***auto_arima(params)*** on a simulated seasonal nonsationary timeseries with properly selected *max_order* parameter and *gridsearch* is shown in Fig 11.

```
                                SARIMAX Results
==========================================================================================
Dep. Variable:                                   y   No. Observations:      800
Model:             SARIMAX(3, 0, 4)x(1, 0, [1], 6)   Log Likelihood    -11417.22
Date:                             Sun, 20 Apr 2025   AIC                22854.45
Time:                                     00:16:09   BIC                22924.32
Sample:                                          0   HQIC               22878.37
                                               - 8000
Covariance Type:                               opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.4970      0.015     32.164      0.000       0.467       0.527
ar.L2         -0.3162      0.022    -14.193      0.000      -0.360      -0.273
ar.L3          0.2088      0.015     13.601      0.000       0.179       0.239
ma.L1          0.3532      0.012     28.585      0.000       0.329       0.377
ma.L2         -0.4244      0.020    -21.263      0.000      -0.464      -0.385
ma.L3          0.2148      0.019     11.488      0.000       0.178       0.251
ma.L4          0.6177      0.013     49.301      0.000       0.593       0.642
ar.S.L6       -0.5492      0.260     -2.114      0.035      -1.058      -0.040
ma.S.L6        0.5784      0.248      2.337      0.019       0.093       1.064
sigma2         1.0160      0.016     62.254      0.000       0.984       1.048
==============================================================================
Ljung-Box (L1) (Q):                   0.03   Jarque-Bera (JB):                 1.43
Prob(Q):                              0.87   Prob(JB):                         0.49
Heteroskedasticity (H):               1.09   Skew:                             0.02
Prob(H) (two-sided):                  0.02   Kurtosis:                         2.94
==============================================================================
```

Fig. 11. Final output of applying *autoarima()* method on a simulated seasonal nonsationary timeseries

The optimization algorithm estimated the nonseasonal ARMA model accuartely and seasonal ARMA model less accuarately. Here algorithm assumes that correct season period (m) is known apriopri.

V. Conclusion

In this review study, the nonstationary seasonal timeseries models are explored using simulated data. Customized python code is used for simulating the data, for computing (and plotting) the correlation functions, and for pole-zero plots. The model identification and estimation were done by built-in method. The saliency of this review study is the use of independent simulations and without losing the theoretical nuances. Another important aspect is the derivation of an explicit recursive equation for simulating a generic nonstationary seasonal timeseries from fundamentals derived from. Moreover, except for few statistical tests, the python code is highly customized without using any python timeseries packages. This makes the review more authentic as there is a lot of transparency in the code functionality.

VI. Future Scope of Work

In the present review study, the nonstationary seasonal timeseries models are explored. The unique feature of this review is that independent simulations are also carried out along with the theory. An explicit expression for simulating a generic nonstationary seasonal timeseries is derived from fundamentals. This expression is used to simulate the timeseries data of different model orders (*p, d, q, P, D, Q)* and model parameters: AR, MA, SAR, SMA using open-source python language. This avoided the problems associated with built-in methods such as version mismatch, deprecated methods, deprecated options, and so on, faced by the author several occasions of simulations. Moreover, except for few statistical tests, the python code is highly

customized without using any python timeseries packages. This makes the review more authentic as there is a lot of transparency in the code functionality.

Conclusions

The limitation of the study is the consideration of only the stationary (SARMA) and nonstationary (SARIMA, SARIMX) models for the review. All these models are assumed to be homogenous models in the sense that the underlying innovation process has constant variance. However, some practical data, especially econometric data, is heteroscedastic i.e., has nonuniform variance. Reviewing such data modelling techniques can be interesting and useful and can be the future scope of work in this direction. However, the review presented in this work paves the way to easy understanding of advanced timeseries models like nonlinear and/or heteroscedastic models (GARCH) and deep learning timeseries models like LSTM networks.

REFERENCES

[1] Ruey S. Tsay, Analysis of Financial Time Series, 3rd ed., A John Wiley & Sons, Inc., 2010.

[2] Mulkkalla, Summanth Redde, "Web Traffic Time Series Forecasting", Master's Projects. 1342, San Jose State University, Dec 2023

[3] H. Nunnagoppula, K. Katragadda and M. Ramesh, "Website Traffic Forecasting Using Deep Learning Techniques," 2023 International Conference on Artificial Intelligence and Smart Communication, Greater Noida, India, 2023, pp. 531-536. https://doi.org/10.1109/AISC56616.2023.10085005

[4] Aditya Amancharla, et.al., "Analysis of EEG and ECG time series in response to olfactory and Cognitive tasks", Procedia Computer Science, vol. 235, pp. 745-756, 2024. https://doi.org/10.1016/j.procs.2024.04.071

[5] Krao MV, "Machine analysis and synthesis of spoken Telugu vowels", Third International Conference on Computational Intelligence and Information Technology, CIIT-2013, 18-19 Oct. 2013, Published by IET., pp.93-104, INSPEC Acc. No: 14542316 @IEEExplore, DOI: https://doi.org/10.1049/cp.2013.2577

[6] Krao MV, and Bhaskararao Peri, "Realization of Telugu r-phoneme in intervocalic position: An acoustic study", 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), pp. 776-782 https://doi.org/10.1109/ICEECCOT52851.2021.9707967

[7] Krao MV and B Rao Peri, "Durational and Formantshift characteristics of Telugu alveolar and bilabial nasal phonemes", MysuruCon-2021, IEEE Mysore Subsection Flagship International Conference, IEEE Conference Record #52639, pp. 386–393, October 2021 *https://doi.org/10.1109/MysuruCon52639.2021.9 641674*

[8] Maddela, V.K.R., Bhaskararao, P. Phonetic–Acoustic Characteristics of Telugu Lateral Approximants. Circuits Syst Signal Process, vol 41, pp.3508–3546, Feb 2022 https://doi.org/10.1007/s00034-021-01949-6

[9] Robert H. Shumway, and David S. Stoffer, Time Series Analysis and Its Applications With R Examples, 4th ed., Springer Texts in Statistics, 2016.

[10] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed., John Wiley & Sons, Inc., 2016.

[11] M. Abotaleb, and T. Makarovskikh, "System for forecasting covid-19 cases using time-series and neural networks models", Engineering Proceedings, vol.5 (1) 46. pp. 1-8, July 2021.

[12] Punyapornwithaya V., et. al., "Time-series analysis for the number of foot and mouth disease outbreak episodes in cattle farms in Thailand using data from 2010–2020", Viruses, 1999-4915, vol. 14 (7), June 2022.

[13] Mahmud Hasan. et. al., "ARMA model development and analysis for global temperature uncertainty", Frontiers in Astronomy and Space Sciences, vol. 10, pp. 1-11, April 2023

[14] Apollinaire B.B., "A comprehensive statistical analysis of Malaria dynamics in the Adamawa region of Cameroon, from 2018 to 2022", Brazilian Journal of Biometrics, vol. 42, pp.289-306, Aug 30, 2024.

[15] Rachim, F., et. al.,"Research on predicting skilled labor availability to enhance sustainability building practices. International Journal of Sustainable Development and Planning, Vol. 19, No. 11, pp. 4183-4192, Nov 2024.

[16] Chiang, S., Zito, J., et. al., Time-series analysis. In Statistical Methods in Epilepsy, pp. 166-200, 2024.

[17] Dhawani Shah and Manishkumar Thaker, "A Review of Time Series Forecasting Methods", International Journal of Research and Analytical Reviews, vol. 11(2), April 2024.

[18] Satheeshna S, Sadul S, Jonathan A, and Krishna Rao M.V., Global Energy Trade Analysis Using ARMIA and Deep Learning Models", Journal of Emerging Trends and Novel Research, vol.03, Issue 03, pp.292-302, March 2025. https://rjpn.org/jetnr/viewpaperforall.php?paper=JETNR2503032

[19] Sushmitha A, Shreya R, Sharmila D, Krishna Rao M.V. "Global Life Expectancy Prediction Using Nonstationary Time Series Models", International Journal of Emerging Technologies and Innovative Research, vol.12, Issue 3, pp.369-377, March 2025 Available: http://www.jetir.org/papers/JETIR2503853.pdf

[20] Vedha Sri N, Vandhana D, Sneha K, and Krishna Rao M.V., "Global Health Expenditure Analysis and Predictions", International Research Journal of Engineering and Technology", vol.12, Issue 03, pp.691-699, March 2025 . Available: www.irjet.net/archives/V12/i3/IRJET-V12I3107.pdf