AIR COMPONENT ANALYSIS USING SARIMA MODEL

Author Name's Arvind Chaudhary

Department of ***Computer Science and Engineering***
Raj Kumar Goel Institute of Technology,
Ghaziabad 201003, India
E-mail Id: -
arvindchaudhary5050@gmail.com

Authors Name's Ashutosh Tripathi

Department of ***Computer Science and Engineering***
Raj Kumar Goel Institute of Technology,
Ghaziabad 201003, India
E-mail Id: -
tashutosh43@gmail.com

Authors Name's Dhruv Vats

Department of ***Computer Science and Engineering***
Raj Kumar Goel Institute of Technology,
Ghaziabad 201003, India
E-mail Id: -
dhruvvats061@gmail.com

Author's Name's Mohd. Siraj Ali

Department of ***Computer Science and Engineering***
Raj Kumar Goel Institute of Technology,
Ghaziabad 201003, India
E-mail Id: -
coolali1301@gmail.com

Abstract—

Air is one of the most important fundamental constituents for the sustenance of life on earth. The meteorological, traffic factors, consumption of non-renewable energy sources, and industrial parameters are steadily increasing air pollution. Our main aim is to find the most polluted cities in recent years and analyze the levels of pollutants here. Understanding the impact of COVID-19 induced lockdowns on Air Quality in some of the major cities: analyzing which cities underwent the most drastic improvement in Air Quality and which cities showed a spike in AQI levels despite a stringent lockdown. We do a time-series analysis of the data and fit a SARIMA model with computed orders to forecast India's AQI in 2021.

Keywords - Air Quality Index, Air Pollution Prediction, ARIMA time series model, Machine Learning.

I. INTRODUCTION

In developing nations such as India, the fast growth in population and economic boom in urban areas has resulted in environmental concerns such as air pollution and a lot of other issues. The health of people is directly impacted by air pollution. In Delhi, there has been a rise in the general public's awareness of the issue. Air pollution has several long-term effects, some of

which include global warming, acid rain, and a rise in the number of asthma sufferers.

According to the World Health Organization (WHO), there are about 2.4 million deaths worldwide because of poor air quality compared to other causes (Ganesh et al., 2018; World Health Organization, 2016). The AQI has been proposed to address the measure of air quality in a region.

In time series analysis our goal is to predict a series that typically is not deterministic since it contains a random component. If this random component is stationary, then we can develop powerful techniques to forecast its future values. In the linear framework, the ARIMA method has been extensively studied in the past and has proven to be effective in forecasting. If "long memory" persists in data, then ARIMA models can be considerably improved upon by employing ARFIMA models. The improvement in the models can be visualized in terms of various factors like Mean Absolute Error, Mean Absolute Percentage Error, Root Mean Square Error, and AICC. A model with the smaller values of these errors is considered as most appropriate.

In the present study, three different time series models have been fitted to predict and forecast the air pollution level in terms of AQI of the pollutant RSPM in the city Chandigarh, located in the north zone of India. The pollutant RSPM is chosen for the time series forecasting as it was found to be the responsible pollutant in 2010 with an AQI of 658.26. For this purpose, the data for the AQIs of the RSPM is taken for the years 2009 and 2010

II. LITERATURE REVIEW

Several air quality forecasting studies are being conducted. To estimate the severity of air pollution, Ishan Verma and his colleagues developed a bi-directional LSTM model. Three

The bidirectional LSTM represents the short, long, and immediate effects of the severity degree of PM 2.5 in this system to increase the prediction accuracy.

This study aimed to use the ARIMA model to estimate the amount of pollution in the air in Zhengzhou. Air pollution from particulate matter from smoke, maa te and out, her sources is a major concern for Chinese cities today. According to WHO et al, lung cancer has been linked to air pollution. (2018).

III. PROPOSED METHODOLOGY

Numpy

NumPy *is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Num array into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy is a Num FOCUS fiscally sponsored project.*

NumPy targets the Python reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms are written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays; using these requires rewriting some code, mostly inner loops, using NumPy.

Dickey-Fuller Test

- *In statistics, the* ***Dickey-Fuller test*** *tests the null hypothesis that a unit root is present in an autoregressive time series model. The alternative hypothesis is different depending on which version of the test is used but is usually stationarity or trend-stationarity. The test is named after the statisticians David Dickey and Wayne Fuller, who developed it in 1979.*

- ***Explanation***

A simple AR(1) model is

yt = pyt-1+ut

where ***yt*** *is the variable of interest,* ***t*** *is the time index,* ***p*** *is a coefficient, and* ***ut*** *is the error term (assumed to be white noise). A unit root is present if p = 1. The model would be non-stationary in this case.*

The regression model can be written as

$$\Delta y_t = (\rho - 1)y_{t-1} + u_t = \delta y_{t-1} + u_t$$

where Δ is the first difference operator and $\delta = \rho - 1$This model can be estimated and testing for a unit root is equivalent to testing $\delta = 0$. Since the test is done over the residual term rather than raw data, it is not possible to use standard t-distribution to provide critical values. Therefore, this statistic ***t*** *has a specific distribution simply known as the Dickey-Fuller.*

There can be three versions of the test.

- $\Delta y_t = \delta y_{t-1} + u_t$ *test for a unit root*
- $\Delta y_t = a_0 + \delta y_{t-1} + u_t$ *test for a unit root with constant*
- $\Delta y_t = a_0 + a_1 t + \delta y_{t-1} + u_t$ *test for a unit root with the constant and deterministic trends with time*

Each version of the test has its critical value which depends on the size of the sample. In each case, the null hypothesis is that there is a unit root, $\delta = 0$. The tests have low statistical power in that they often cannot distinguish between true unit-root processes ($\delta = 0$) and near-unit-root processes (δ is close to zero). This is called the "near observation equivalence" problem.


```
diff = cities['India_AQI'].diff(periods=1)
diff.dropna(inplace=True)
fig = seasonal_decompose(diff, model='additive').plot()
```

```
dftest = adfuller(diff)
dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Nu
for key,value in dftest[4].items():
    dfoutput['Critical Value (%s)'%key] = value
dfoutput
```

```
Test Statistic                -8.385232e+00
p-value                        2.448599e-13
#Lags Used                     9.000000e+00
Number of Observations Used    5.600000e+01
Critical Value (1%)           -3.552928e+00
Critical Value (5%)           -2.914731e+00
Critical Value (10%)          -2.595137e+00
dtype: float64
```

From the p-value and the Test Statistic, we can conclude that with one differencing, the time series becomes stationary. Therefore, d=1.

```
fig, ax = plt.subplots(2,figsize=(13, 8))
ax[0] = plot_acf(diff, lags=30, ax=ax[0])
ax[1] = plot_pacf(diff,lags=30, ax=ax[1])
```

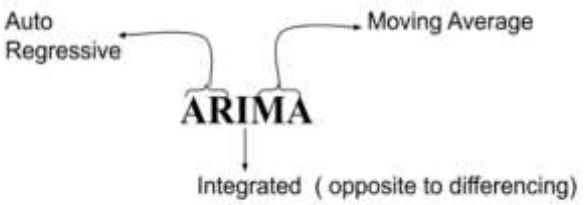
IV. Modules Used

Autoregressive Integrated Moving Average, or ARIMA, is one of the most widely used forecasting methods for univariate time series data forecasting.
Although the method can handle data with a trend, it does not support time series with a seasonal component. An extension to ARIMA that supports the direct modeling of the seasonal component of the series is called SARIMA.
We will discover the Seasonal Autoregressive Integrated Moving Average, or SARIMA, a method for time series forecasting with univariate data containing trends and seasonality.
We will know:
The limitations of ARIMA when it comes to seasonal data.
The SARIMA extension of ARIMA explicitly models the seasonal element in univariate data.
How to implement the SARIMA method in Python using the Statsmodels library.

ARIMA (p,q,d)

ARIMA is an acronym that stands for Auto-Regressive Integrated Moving Average. It's a class of models that captures a suite of different standard temporal structures in time series data. It explicitly caters to a suite of standard structures in time-series data, and as such provides a simple, powerful method for making skillful time-series forecasts. It's a generalization of the simpler Auto-Regressive Moving Average, with the added notion of integration.

- ***AR**: Autoregression. A model that uses the dependent relationship between observation and some number of lagged observations.*
- ***I**: Integrated. The use of differencing of raw observations (e.g. subtracting an observation from observation at the previous time step) to make the time series stationary.*
- ***MA**: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model is applied to lagged observations.*



We've covered the pieces, now we'll combine them to form the ARIMA model.

Each of these components is explicitly specified in the model as a parameter. Standard notation is used for ARIMA (p,d,q), where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used.

The parameters of the ARIMA model are defined as follows:

- ***p**: AR. The number of lag observations included in the model is also called the lag order.*
- ***d**: I. The number of times that the raw observations are differenced is also called the degree of difference.*
- ***q**: MA. The size of the moving average window is also called the order of moving average.*

How does the ARIMA model work?

Simply put, we have 3 parameters in ARIMA (p,q,d).
p is from Auto Regression, q is from Moving Averages, and d is from differencing.
d can be any order of differencing.
All three parameters are hyper-parameters that need to experiment with and figure out which fits best, just like K in K-NN. If d = 2 instead of predicting y_t we will use y_t'' to model.
Now we have ARMA(p,q). What is ARMA? ARMA is ARIMA without the I, the Integrating part.
p corresponding to AR and q corresponding to MA.
The model looks like this:

$$y_t = \mu + (\alpha_1 y_{t-1} + \ldots\ldots\ldots + \alpha_p y_{t-p}) + (\theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \theta_3 \epsilon_{t-3} + \ldots\ldots\ldots + \theta_q \epsilon_{t-q}) + \epsilon_t$$

*Here, μ is some constant + linear combination of the previous **p** + linear combination of the previous **q** errored terms + the error this time(ϵ_t).*

What happens if we take ϵ_t to the other side of the equation? This becomes:

$$\hat{y_t} = \mu + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{q} \theta_j \epsilon_{t-j}$$

This equation is the same as the previous equation. We have a constant + linear combination of previous P values + linear combination of previous errored terms.

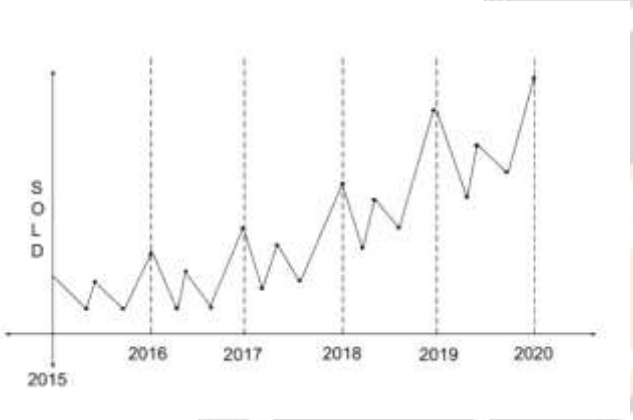
Imagine how we can model this into Linear Regression? We'll take the previous p values as features, and previous q errors as features, α_i, and θ_i will be linear regression. This is also a linear regression problem. We can say ARIMA is nothing but a linear regression.

What's wrong with ARIMA

Autoregressive Integrated Moving Average, or ARIMA, is a forecasting method for univariate time series data.
As its name suggests, it supports both autoregressive and moving average elements. The integrated element refers to differencing allowing the method to support time-series data with a trend.
A problem with ARIMA is that it does not support seasonal data. That is a time series with a repeating cycle.
ARIMA expects data that is either not seasonal or has the seasonal component removed, e.g. seasonally adjusted via methods such as seasonal differencing.
An alternative is to use SARIMA.

What is SARIMA?

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component.
It adds three new hyperparameters to specify the autoregression (AR), differencing (I), and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.
A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA. The seasonal part of the model consists of terms that are very similar to the non-seasonal components of the model, but they involve backshifts of the seasonal period.
Let's incorporate seasonality into the ARIMA model.



How do we know we should use the seasonal ARIMA(SARIMA) model?
The above is drawn to show the seasonality. We see a very clear ***W-type*** *pattern repeating, so we have seasonality.*
In ***SARIMA (P, Q, D)*** $_m$***:*** *m is the seasonal factor. It's the number of time steps for a single seasonal period.*
In the above graph, consider each year has 4 quarters. Now we'll have an m value equal to 4.
The (P, D, Q) are the analogs of (p, q, d), except for the seasonal component

V. ***RESULT AND DISCUSSION***

The section shows the overall accuracy of the ARIMA and SARIMA models. So, this works gives better Air Pollution prediction compared to the existing method.

SARIMAX Results

Dep. Variable:	India_AQI	**No. Observations:**	48
Model:	SARIMAX(0, 1, 2)x(1, 0, [1], 12)	**Log Likelihood**	-229.813
Date:	Sat, 21 May 2022	**AIC**	469.625

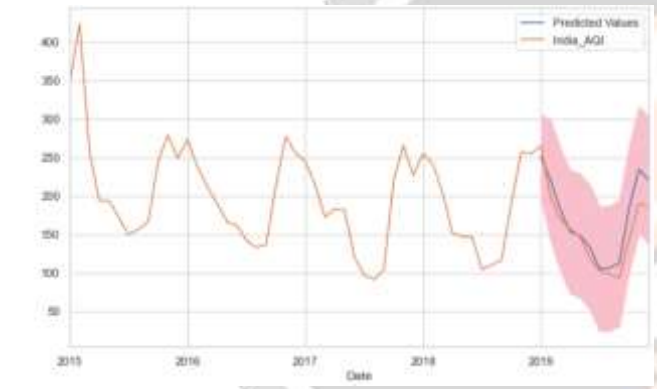
Time:	11:10:21	**BIC**	478.876
Sample:	01-01-2015	**HQIC**	473.106
	- 12-01-2018		
Covariance Type:	opg		

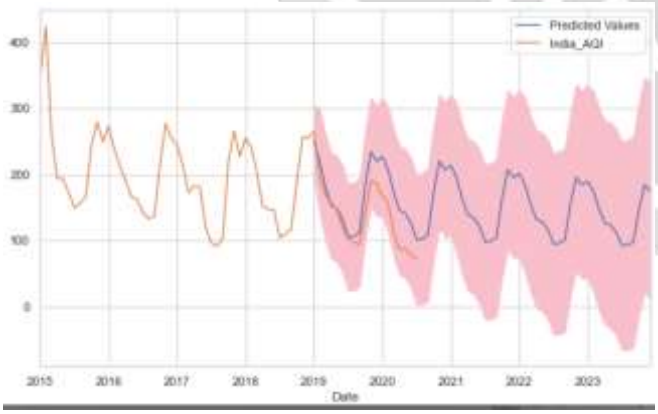
	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	0.0647	0.660	0.098	0.922	-1.229	1.359
ma.L2	-0.9326	0.646	-1.444	0.149	-2.198	0.333
ar.S.L12	0.9183	0.097	9.439	0.000	0.728	1.109
ma.S.L12	-0.4473	0.301	-1.484	0.138	-1.038	0.143
sigma2	766.9859	487.993	1.572	0.116	-189.464	1723.435

Ljung-Box (L1) (Q):	0.23	**Jarque-Bera (JB):**	3.61
Prob(Q):	0.63	**Prob(JB):**	0.16

Ljung-Box (L1) (Q): 0.23 **Jarque-Bera (JB):** 3.61
Prob(Q): 0.63 **Prob(JB):** 0.16

Heteroskedasticity (H):	0.24	**Skew:**	-0.66
Prob(H) (two-sided):	0.01	**Kurtosis:**	3.28





VI. CONCLUSION

1. *Air pollution is one of the most serious environmental problems in urban societies. Since prediction is essential for planning for adopting policies and taking necessary measures for reducing and preventing pollution increase and critical air quality*

VII. REFERENCES

- *We proposed a predictive data feature exploration-based air quality prediction approach to note the results to avoid it in future situations Valipour, Mohammad. "Long-term runoff study using SARIMA and ARIMA models in the United States." Meteorological Applications 22.3 (2015): 592-598.*
- *Nobre, Flávio Fonseca, et al. "Dynamic linear model and SARIMA: a comparison of their forecasting performance in epidemiology." Statistics in Medicine 20.20 (2001): 3051-3069.*
- *Cheng, Wan-Li, et al. "Comparison of the revised air quality index with the PSI and AQI indices." Science of the Total Environment 382.2-3 (2007): 191-198.*
- *Gilbert, Kenneth. "An ARIMA supply chain model." Management Science 51.2 (2005): 305-310.*
- *Kalpakis, Konstantinos, Dhiral Gada, and Vasundhara Puttagunta. "Distance measures for effective clustering of ARIMA time-series." Proceedings 2001 IEEE international conference on data mining. IEEE, 200.*
- *Lee MH, Rahman NHA, Latif MT, Nor ME, Kamisan NAB (2012) Seasonal ARIMA for forecasting air pollution index: a case study. Am J Appl Sci 9(4):570–578.*
- *Naveen V, Anu N (2017) Time series analysis to forecast air quality indices Thiruvananthapuram District, Kerala, India. Int J Eng Res Appl. 7(6)(Part-3):6684. ISSN: 2248-9622. Kumar A Goyal P (2011) Forecasting of daily air quality*