Covid – 19 Forecasting

Nithish Kumar S[1], Sodam Shreya Reddy[2], Shruthi Ravindra[3]

[1,2,3] **Department of Information Technology, VIT University, Vellore, Tamil Nadu, India**

Abstract

This paper presents a comparative study of forecasting techniques for the prediction of Covid – 19 deaths. Various techniques have been analyzed after going through literature surveys and the best technique have been implemented to accurately predict various factors such as new deaths and new cases. These techniques have been implemented assuring the major factors being capacity, robustness and the precision. The proposed methods have been compared against statistical parameters like mean absolute error (MAE). The experimental results suggest that using a particular method is subjective to its application.

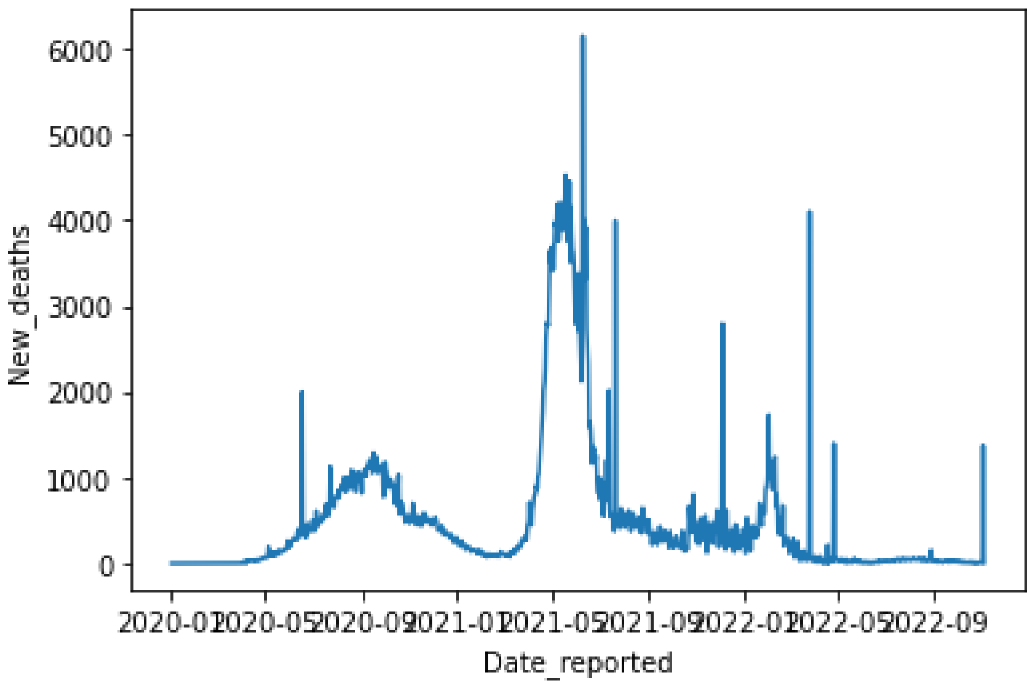
Keywords – COVID – 19, Forecasting, ARIMA, Moving Average, Prediction, MAE

I. INTRODUCTION

Integration of artificial intelligence (AI) techniques in wireless infrastructure, real-time collection, and processing of end-user devices is now in high demand. It is now superlative to use AI to detect and predict pandemics of a colossal nature. The Coronavirus disease 2019 (COVID-19) pandemic, which originated in Wuhan China, has had disastrous effects on the global community and has overburdened advanced healthcare systems throughout the world. he dominant symptoms include fever and cough, while gastrointestinal symptoms are uncommon. In COVID-19 infected patients the absence of fever is more frequent than in patients infected by similar viruses. The initial patients infected by COVID-19, reportedly indicated an association with a large seafood and animal market in Wuhan that demonstrated an animal-to-person spread. Per contra, a burgeoning number of patients have not displayed any association with the animal markets, revealing the fact of human-to-human transmission of COVID-19. This pandemic has been declared a global health emergency and is spreading at an alarming rate. The dependencies for the project include the following packages and libraries: Datetime, Numpy, Pandas, SciPy, Scikit Learn, and Matplotlib. This project has been implemented on the Google Colab platform using the CPU runtime.

II. BACKGROUND

SARS-coronavirus-2 is a new virus infecting people and causing COVID-19 disease. The disease is causing a worldwide pandemic. Although some people never develop any signs or symptoms of disease when they are infected, other people are at very high risk for severe disease and death. If we’re able to intervene to prevent even some transmission, we can dramatically reduce the number of cases. And this is the public health goal for controlling COVID-19. So, this is an approach for comparatively accurate values prediction of new cases for a particular day in order to be considered for preventive measures. Forecasting, however, requires ample historical data with no perfect prediction. Forecasts depend upon data reliability and variables of interest for prediction. The availability of techniques for forecasting infectious disease can make it easier to fight COVID-19. We include methods for forecasting future cases based on existing data.



III. LITERATURE SURVEY

The following table suggest the various techniques available for Covid – 19 forecasting:

Authors & Year	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Hamadini & 2020	Deep Learning	Predicts the increase	Insufficient Dataset	WHO	MAE
Reshi & 2020	CNN model	Predicts the increase	Insufficient Dataset	Kaggle	RMSE
Bayuut & 2020	ARIMA model	Reduces Time complexity	Inadequate metrics	Github	MSE
Mvall & 2020	ANN model	Predicts the increase	Improve Accuracy	WHO	MAE
Steffy & 2020	Decision Tree	Accuracy: 85%	Improve Accuracy	WHO	MAE

Authors & Year	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Rodriguez 2020	ANN model	Predicts the increase	Insufficient Dataset	Kaggle	MAE
Burdick & 2020	Polymerase Chain Reaction	Accuracy: 81%	Insufficient Accuracy	WHO	MAE
Santosh & 2020	SEIR model	Predicts the increase	Insufficient Dataset	WHO	RMSE
Ayman & 2020	Multinomial logistic regression	Accuracy: 82%	Insufficient Accuracy	Kaggle	MLR
D.Haritha & 2020	Transfer learning	Error is minimized	Insufficient Dataset	Kaggle	MAE
Howard & 2020	ANN model	Error is minimized	Insufficient Accuracy	WHO	PIBA
Tomar A & 2020	RNN model	Accuracy: 82%	Insufficient Accuracy	WHO	Accuracy
Alberti 2020	Logistic model	Accuracy: 81%	Insufficient Accuracy	Kaggle	MAE
Ardabili & 2020	ANN model	Error is minimized	Insufficient Accuracy	worldometer.info	MAE
Lixiang Li 2020	Gaussian Distribution Theory	Error is minimized	Inadequate metrics	Kaggle	Accuracy
Wynants & 2020	ARIMA model	Reduces Time complexity	Inadequate metrics	-	Recall

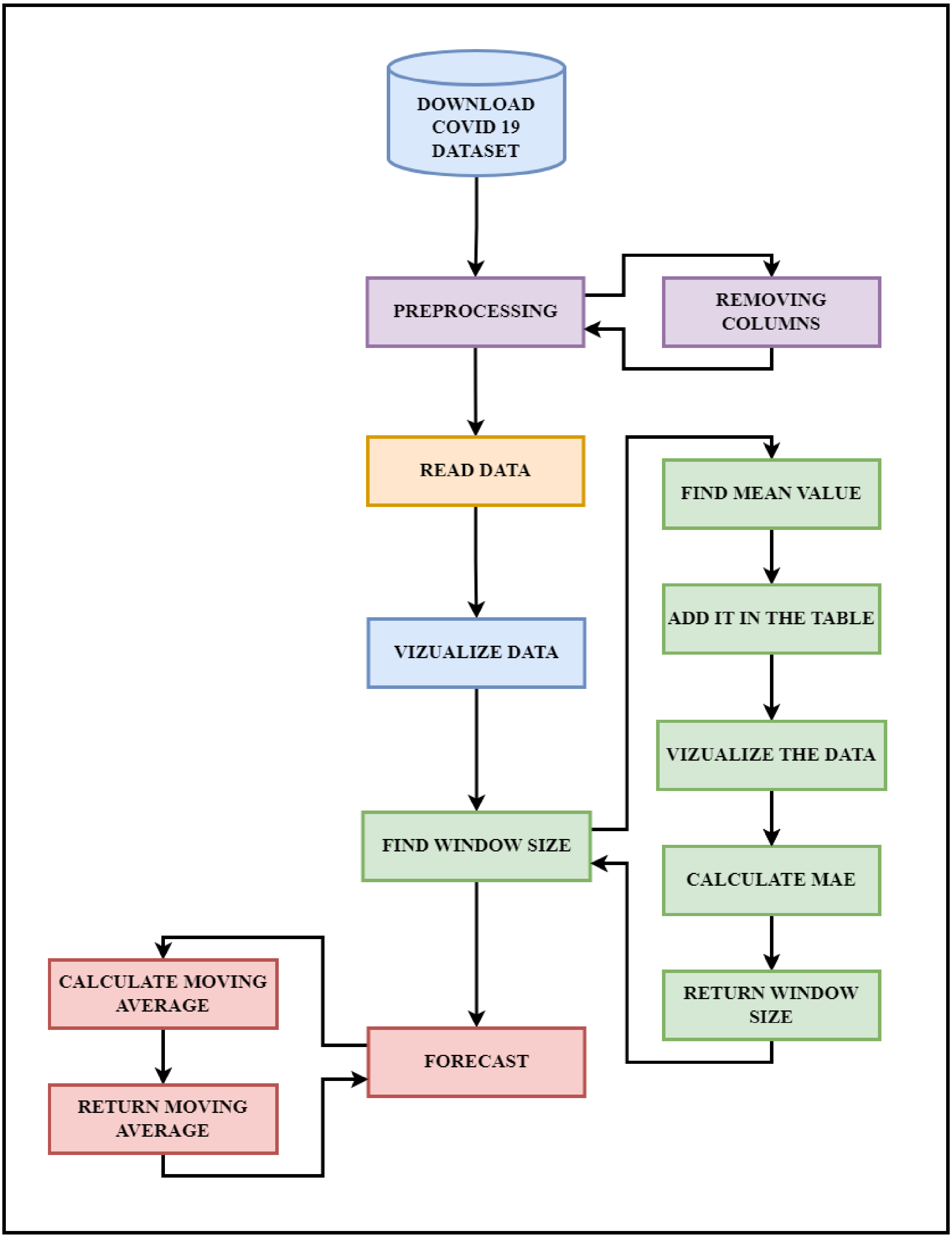
Authors & Year	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Radcliff & 2020	CNN model	Accuracy: 80%	Insufficient Accuracy	WHO	NPV
Zifeng & 2020	SIER model	Accuracy: 86%	Inadequate metrics	WHO	Recall
Subhas & 2020	SAIUQR model	Error is minimized	Inadequate metrics	Kaggle	MLR
Faulstick & 2020	Genetic Algorithm	Error is minimized	Insufficient Accuracy	WHO	RMSE
Shuhaida 2020	RF-SSA model	Accuracy: 82%	Inadequate metrics	WHO	Recall
PeterBaker 2021	logistic regression	Accuracy: 85%	Insufficient Accuracy	WHO	MAE
Nandi & 2021	CNN model	Error is minimized	Insufficient Accuracy	Kaggle	Accuracy
XiZhang & 2021	Evolution Algorithm	Accuracy: 86%	Insufficient Accuracy	WHO	MSE
Saurabh & 2021	Transfer Learning	Accuracy: 80%	Improve Accuracy	Kaggle	Accuracy
Shakeel & 2021	ARIMA model	Accuracy: 87%	Inadequate metrics	WHO	RMSE

Authors & Year	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Saif & 2021	ARIMA model	Accuracy: 85%	Insufficient Dataset	WHO	MSE
Moreira & 2021	PROMETHEE-GAIA	Error is minimized	Improve Accuracy	WHO	NPV
Podolski & 2021	Cellular Automata	Error is minimized	Improve Accuracy	WHO	MAE
Oliveira & 2021	ANN model	Accuracy: 80%	ANN is less efficient	WHO	MSE
Erlisa & 2021	LSTMNN model	Error is minimized	Data type to be improved	WHO	MSE
Khodeir & 2021	PICO model	Accuracy: 82%	Improve Accuracy	WHO	MAE
Srivastava 2021	Deep Learning	Accuracy: 87%	Inadequate metrics	Kaggle	MAPE
Adimoolam & 2021	RF-SSA model	Error is minimized	Improve Accuracy	Kaggle	MAE
Shareef & 2022	ANN model	Accuracy: 86%	Improve Accuracy	Github	MSE
Nagaraj & 2022	multilayer perceptron	Accuracy: 80%	Data type to be improved	WHO	MSE
Tulshyan & 2022	Prophet model	Accuracy: 82%	Inadequate metrics	Kaggle	MAPE

Authors & Year	Methodology or Techniques used	Advantages	Issues	Dataset used	Metrics used
Chen & 2022	Transfer Learning	Accuracy: 80%	Improve Accuracy	Kaggle	Accuracy
Mydukuri & 2022	LSRGNFM-LDC	Error is minimized	Inadequate metrics	WHO	Accuracy
Haritha & 2022	Fuzzy C-Means	Accuracy: 84%	Inadequate metrics	WHO	Accuracy
Niteesh & 2022	FCM Clustering	Accuracy: 85%	Insufficient Accuracy	Kaggle	MSE RMSE
Guohui & 2022	ARIMA model	Error is minimized	Inadequate metrics	WHO	RMSE
Mehar & 2022	CNN (Image based)	Accuracy: 90%	Insufficient Accuracy	Kaggle	Accuracy
Vibhor & 2022	Polynomial Regression	Accuracy: 93%	Improve Accuracy	GitHub, Kaggle	MAE

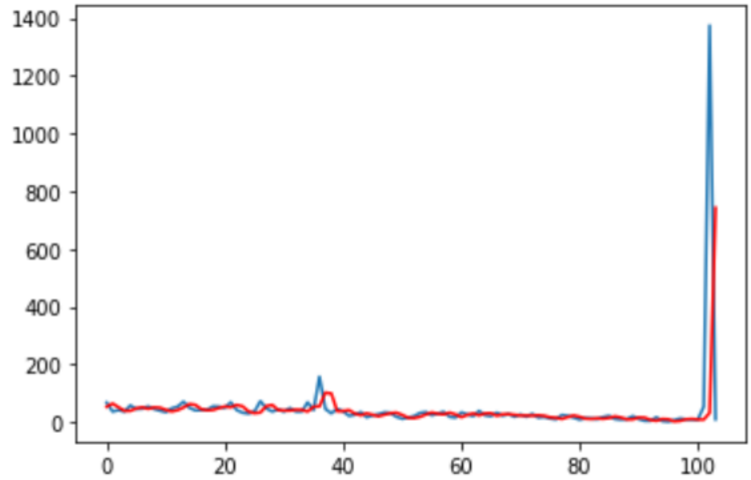
We conclude from the literature survey that Polynomial regression is one of the best method for COVID – 19 Forecasting which can be implemented using ARIMA model. But, the accuracy of the prediction can be further improved to best forecast the new COVID cases.

PROPOSED ALGORITHM



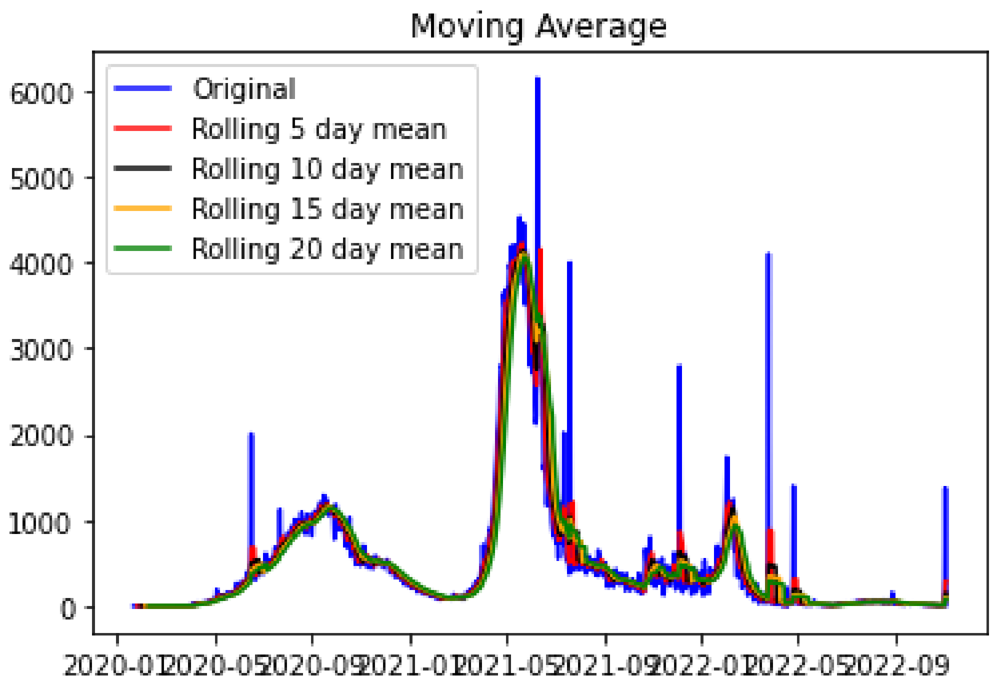
IV. EXPERIMENT RESULTS

ARIMA MODEL:




```
Mean Absolute Error from ARIMA Model: 30.851
```

MOVING AVERAGE:




```
Mean Absolute Error from 5day moving average prediction: 21.838
Mean Absolute Error from 10day moving average prediction: 22.352000000000004
Mean Absolute Error from 15day moving average prediction: 22.985333333333333
Mean Absolute Error from 20day moving average prediction: 23.5555
```

V. COMPARATIVE STUDY

In this project, we are comparing two prediction techniques for COVID – 19 Forecasting which are ARIMA model and Moving Average method. ARIMA model predicts the number of death cases with the error of 30.851, whereas the 5 – day Moving Average method predicts the number of death cases with the error of 21.838. Thus, we can conclude that 5 – day Moving Average method can forecast the number of new cases / death cases accurately and closely approximate the data.

VI. CONCLUSION AND FUTURE WORK

In summary, COVID-19 has put some significant and unprecedented strain on global supply chains across most product categories. Literatures on forecasting and on supply chain disruption has been able to provide some indication of the factors that can lead to it. However, and at the same time, it has exposed some of the challenges associated with identifying and responding to significant changes in the demand patterns during a pandemic. In this project, we have successfully compared and implemented both ARIMA and Moving Average method. Therefore, we can conclude that Moving Average method predicts new deaths better than ARIMA method with significantly less error.

VII. REFERENCES

a. https://www.sciencedirect.com/science/article/pii/S2090447921000502
b. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8408413/
c. https://www.sciencedirect.com/science/article/pii/S1877050922000527
d. https://www.sciencedirect.com/science/article/pii/S1877050921018494
e. https://www.sciencedirect.com/science/article/pii/S1877050921002970
f. https://www.proquest.com/docview/2652966618?accountid=38207
g. https://onlinelibrary.wiley.com/doi/full/10.1111/exsy.12694
h. https://ieeexplore.ieee.org.egateway.vit.ac.in/document/9276753
i. https://ieeexplore.ieee.org.egateway.vit.ac.in/document/9579985
j. https://ieeexplore.ieee.org.egateway.vit.ac.in/document/9823691
k. https://ieeexplore.ieee.org.egateway.vit.ac.in/document/9792709
l. https://www.sciencedirect.com/science/article/pii/S2090447921000502
m. https://journals.sagepub.com.egateway.vit.ac.in/doi/pdf/10.1177/09720634211050425
n. https://www.cambridge.org/core/journals/disaster-medicine-and-public-health-preparedness/article/an-eye-on-the-future-of-covid19-prediction-of-likely-positive-cases-and-fatality-in-india-over-a-30-days-horizon-using-prophet-model/0A68D6479CC1149F7E2EE726527AE762
o. https://www.sciencedirect.com.egateway.vit.ac.in/science/article/pii/S0957582021005917
p. https://www.sciencedirect.com.egateway.vit.ac.in/science/article/pii/S0019057821003700

VIII. CODING

```
import pandas as pd
import numpy as np
import datetime as dt
from matplotlib import pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.preprocessing import PolynomialFeatures
from pandas.plotting import autocorrelation_plot
from statsmodels.tsa.arima.model import ARIMA
```

```
data = pd.read_csv("
      https://drive.google.com/uc?id=1i5mp18QvgKP_Esq2WM765c8Q4jV7v_9H")

data.head()

plt.xlabel('Date_reported')
plt.ylabel('New_deaths')
plt.plot(data['Date_reported'],data['New_deaths'])
plt.show

rolling_mean5 = data['New_deaths'].rolling(window = 5).mean()
rolling_mean10 = data['New_deaths'].rolling(window = 10).mean()
rolling_mean15 = data['New_deaths'].rolling(window = 15).mean()
rolling_mean20 = data['New_deaths'].rolling(window = 20).mean()
plt.plot(data['Date_reported'],data['New_deaths'], color = 'blue', label =
'Original')
plt.plot(data['Date_reported'],rolling_mean5, color = 'red', label = 'Rolling
5 day mean')
plt.plot(data['Date_reported'],rolling_mean10, color = 'black', label =
'Rolling 10 day mean')
plt.plot(data['Date_reported'],rolling_mean15, color = 'orange', label =
'Rolling 15 day mean')
plt.plot(data['Date_reported'],rolling_mean20, color = 'green', label =
'Rolling 20 day mean')
plt.legend(loc = 'best')
plt.title('Moving Average')
plt.show()

data['moving_avg_5day'] = rolling_mean5
data['moving_avg_10day'] = rolling_mean10
data['moving_avg_15day'] = rolling_mean15
data['moving_avg_20day'] = rolling_mean20
data.head(100)

data.dropna(inplace=True)

X = data['New_deaths'].values
size = int(len(X) * 0.90)
train, test = X[0:size], X[size:len(X)]
traindf, testdf = data[0:size], data[size:len(X)]
history = [x for x in train]
predictions = list()
for t in range(len(test)):
    model = ARIMA(history, order=(1,1,0))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat = output[0]
    predictions.append(yhat)
```

```python
    obs = test[t]
    history.append(obs)
predictions = np.array(predictions)

plt.plot(test)
plt.plot(predictions, color='red')
plt.show()

mae = mean_absolute_error(test, predictions)
print('Mean Absolute Error from ARIMA Model: %.3f' % mae)


mae5day = mean_absolute_error(testdf['New_deaths'], testdf['moving_avg_5day'])
mae10day = mean_absolute_error(testdf['New_deaths'],
testdf['moving_avg_10day'])
mae15day = mean_absolute_error(testdf['New_deaths'],
testdf['moving_avg_15day'])
mae20day = mean_absolute_error(testdf['New_deaths'],
testdf['moving_avg_20day'])

print(f'Mean Absolute Error from 5day moving average prediction: {mae5day}')
print(f'Mean Absolute Error from 10day moving average prediction: {mae10day}')
print(f'Mean Absolute Error from 15day moving average prediction: {mae15day}')
print(f'Mean Absolute Error from 20day moving average prediction: {mae20day}')
```