# CHAPTER 1

**INTRODUCTION**

Global warming stands as one of the most pressing challenges of our time, with far-reaching consequences for the environment, ecosystems, and human societies. As the Earth's climate continues to undergo unprecedented changes, there is an urgent need for comprehensive analyses and accurate predictions to guide mitigation efforts. This research endeavors to address this imperative by conducting an in-depth investigation into global warming trends, utilizing three key datasets—Earth surface temperature, CO2 emissions, and world population data. The integration of advanced machine learning models, including SARIMAX, ARIMAX, VAR, and VECM, enhances our ability to discern intricate patterns and forecast future climate dynamics.

## 1.1 BACKGROUND

The acceleration of global warming is primarily attributed to human activities, particularly the emission of greenhouse gases such as carbon dioxide (CO2). Understanding the complex interactions between Earth surface temperature, atmospheric CO2 levels, and the burgeoning world population is essential for devising effective strategies to mitigate the impacts of climate change. The utilization of machine learning models in climate science has gained prominence due to their capacity to capture nonlinear relationships, temporal dependencies, and intricate dynamics present in climate datasets.

## 1.2 SIGNIFICANCE

This research holds significance as it contributes valuable insights to the ongoing discourse on global warming, providing evidence-based analyses and predictions that can inform climate change mitigation strategies. The utilization of machine learning models adds a novel dimension to our understanding of climate dynamics, facilitating more accurate and nuanced predictions for a sustainable future. The findings of this study aim to guide policymakers, researchers, and stakeholders in making informed decisions to address the challenges posed by global warming.

## 1.3 OBJECTIVES

The primary objectives of this research are:

* To conduct a thorough analysis of Earth surface temperature trends over a specified time period.
* To explore the relationship between CO2 emissions and climate dynamics, unraveling the impact of human activities on atmospheric CO2 concentrations.
* To investigate the correlation between world population growth and its implications for climate change.
* To employ SARIMAX, ARIMAX, VAR, and VECM for predictive modeling, offering accurate forecasts of future trends in Earth surface temperature, CO2 emissions, and world population.

# CHAPTER 2

**LITERATURE SURVEY**

A literature survey, also known as a literature review, is a critical analysis and synthesis of existing research and scholarly articles related to a particular topic or research question. It is an essential component of academic and research work, providing a comprehensive overview of the current state of knowledge in a specific field.

Harvey Zheng in “Analysis of Global Warming Using Machine Learning” , 2018, addresses the controversial topic of climate change, particularly in the US where skepticism exists. Recognizing the predicted dire consequences, such as mass ocean extinction and extreme weather events, the study focuses on constructing reliable statistical models based on 800,000 years of climate data. By comparing mainstream machine learning algorithms, including linear regression, lasso, support vector regression, and random forest, the research identifies that random forest outperforms others in accurately forecasting global atmosphere changes. The study highlights the significance of greenhouse gases, with CO2 identified as the primary contributor to temperature change, followed by CH4 and N2O. Controlling the release of these gases is crucial to mitigating temperature increase and preventing potential climate change impacts.

Adwait Prakash Mishra, Umesh Pratap Singh in "GLOBAL WARMING PREDICTION USING MACHINE LEARNING", 2021, aim to predict the concentration of greenhouse gases causing Earth's surface temperature increase. Utilizing a time series regression model in machine learning, forecast both Earth's surface temperature and the change in greenhouse gas concentrations. These predictions serve to gauge the urgency of the situation, informing strategies to cope with or reduce the use of greenhouse gas-emitting fuels.

D. Deva Hema, Anirban Pal, Vineet Loyer, Rajeev Gaurav in "Global Warming Prediction in India using Machine Learning ", 2019, assesses the performance of various machine learning algorithms (Linear Regression, Multi-Regression Tree, Support Vector Regression, Lasso) in predicting annual global warming based on past measurements over India. The study addresses challenges in creating a reliable statistical model on a large dataset, focusing on the relationship between average annual temperature and greenhouse gas concentrations like carbon dioxide, methane, and nitrous oxide. Linear regression is identified as the most accurate method for predicting greenhouse gases and temperature. The findings highlight the significant role of CO2 as the major contributor to temperature change, followed by CH4 and N2O. The analysis suggests that with a reduction in greenhouse gases, global warming can be mitigated within a few years, benefiting both humans and wildlife affected by rising temperatures.

M. Purushotham Reddy; A. Aneesh; K. Praneetha; S. Vijay in "**Global Warming Analysis and Prediction Using Data Science",** 2021, focuses on analyzing the application of the linear regression machine learning algorithm for predicting global temperature and carbon emissions based on historical data from India. Recognizing the broad significance of long-term global warming predictions across various sectors, the study emphasizes the precision of linear regression in forecasting these variables. The initial step involves constructing a consistent and reliable statistical model on a large dataset to capture the relationship between average annual temperature and factors contributing to global warming. The potential reduction in global temperature is underscored as a benefit for both humans and various animal species affected by the impacts of global warming.

Jiecheng Song et, al, in "**Data driven pathway analysis and forecast of global warming and sea level rise", 2023,** introduces a data-driven pathway analysis framework to understand the key processes influencing mean global temperature and sea level rise, projecting their magnitude until 2100. Utilizing historical data and dynamic statistical modeling, the study establishes causal pathways connecting rising greenhouse gas emissions to global temperature and sea level changes, involving humidity, sea ice coverage, and glacier mass. The findings indicate that without mitigating actions, the global temperature is projected to increase by approximately 3.28 °C above pre-industrial levels, with a sea level rise of 573 mm by 2100. However, adherence to the greenhouse gas emission regulations outlined in the 2021 United Nations Conference on Climate Change (COP26) could mitigate these impacts, resulting in a reduced temperature increase of 1.88 °C and a sea level rise of 449 mm by 2100.

Leon Wang et, al, in "A century-long analysis of global warming and earth temperature using a random walk with drift approach", 2023, investigates the alarming trend of global warming, utilizing temperature data analysis over the past century. Employing models, including the Random Walk with Drift approach implemented with the R programming language, the research compares different time horizons and scenarios. The study emphasizes the significance of advanced analytical techniques to comprehend the impact of climate change. The findings underscore the urgency of implementing effective climate policies to mitigate global warming's effects and ensure the well-being of our planet.

Ms. Nisha Bairagee, Mrs. Nitima Malsa, Dr. Jyoti Gautam in "Prediction on Global Warming ", 2016, addresses the critical issue of global warming and its environmental impacts, particularly focusing on the aftermath factors. The main concern is the release of carbon dioxide. The study employs classification and prediction techniques to categorize global warming factors and forecast their future effects on the atmosphere, consequently influencing the environment.

Lingyun Zhu in "**Global Warming: Temperature Prediction Based on ARIMA", 2023,** addresses concerns about global climate change, particularly focusing on global warming and frequent extreme weather events. Leveraging big data and information technology, the study emphasizes improved scientific and technical support for temperature prediction. Utilizing time series analysis, the paper examines global temperature trends over the past century and forecasts trends for the next century using data from BERKELEY EARTH (1900-2022). The ARIMA (12,1,5) model is applied, revealing an upward trend in global temperatures, indicating a continuation and intensification of global warming.

Navaneetha Krishnan M, Ranjith R, Lavanya B in "**Climate Change Prediction Using ARIMA Model**", 2022, addresses the challenging task of accurately forecasting weather data, emphasizing the importance of temperature change for business and economic activities. It utilizes univariate statistical techniques to model a world mean temperature dataset, aiming to develop a concise forecasting model for short-term managerial decision-making. The study employs the ARIMA algorithm in Python to create a prognostication tool. The methodology, though applied to global temperature data, can be extended to more localized levels. Statistical techniques include seasonal and non-seasonal unit root testing, structural breaks, as well as ARIMA and SARIMA modeling. The paper contributes to predicting air temperature, a crucial aspect of addressing global warming, and offers insights into the likely impact of climate change on monthly mean temperatures in Tamilnadu. The developed time-series model facilitates short-term forecasting (5-10 years) of global mean temperature change.

# CHAPTER 3

**SYSTEM DESIGN**

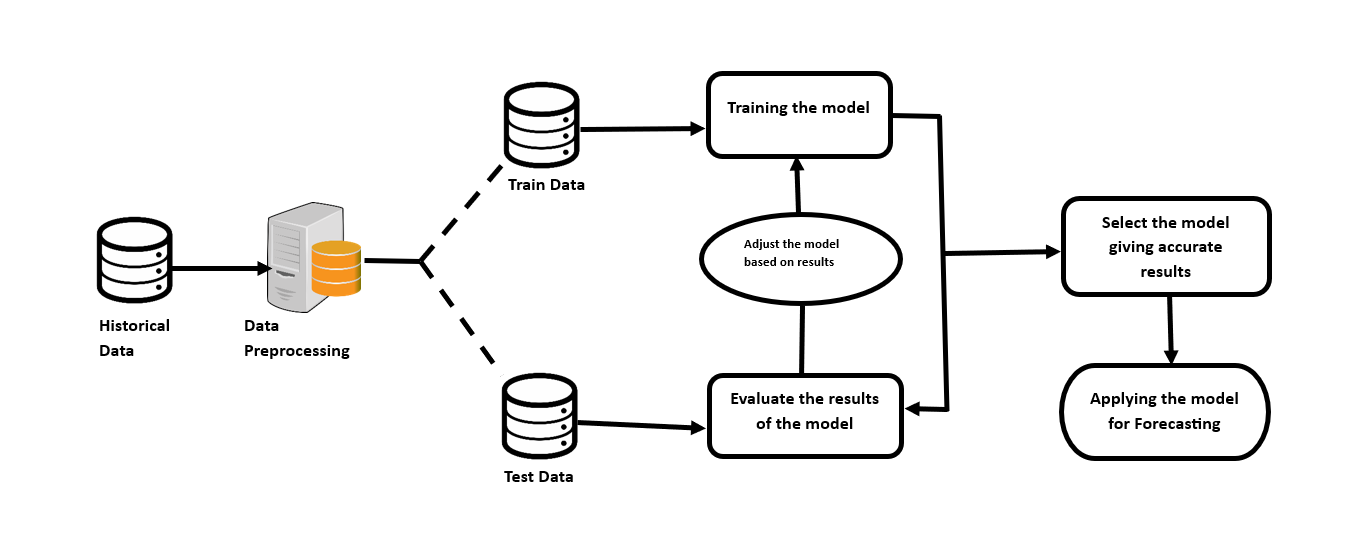
## 3.1 Existing System

* Machine learning models benefit from extensive historical data. However, in some cases, the availability of long-term, high-quality data may be limited.
* Climate systems are inherently complex and nonlinear. Traditional machine learning models may struggle to capture intricate relationships, leading to oversimplified representations of the climate dynamics.
* Climate predictions inherently involve uncertainties due to the complex nature of atmospheric processes. Predictions for extended timeframes may have higher uncertainties.
* Some models may not adequately incorporate socioeconomic factors and human behavior, which play a crucial role in shaping greenhouse gas emissions and climate change impacts.

## 3.2 Proposed System

* Develop time series models capable of accurately predicting Earth's surface temperature. These models will be designed to capture patterns and variations in temperature data over extended periods.
* Improve temperature forecasts by incorporating advanced multivariate time series techniques. This involves integrating exogenous features, conducting cointegration analyses, and leveraging Granger causality analysis.
* Investigate the relationship between Earth's surface temperature and various influencing factors. Utilize cointegration analysis to identify long-term associations and dependencies between temperature and these factors.
* Evaluate the causal relationships between temperature and identified factors, shedding light on how each element contributes to the global warming trend. Assess the magnitude and direction of these causal connections to better understand the dynamics of climate change.

## 3.3 System Architecture

An architecture diagram is a visual representation of the high-level structure and components of a system, application, or software. It provides an overview of the system's design and how different components interact with each other to achieve the overall functionality. Architecture diagrams are valuable for communication among stakeholders, including developers, architects, project managers, and other team members.

## 3.4 System Requirements

## 3.4.1 Hardware Requirements

* + Processor – Intel® Core™ i5-7200U CPU @ 2.50GHz 2.71GHz
  + Hard disk drive –256 GB
  + RAM – 8.00 GB
  + OS – Windows 10 Home ©2017 Microsoft corporation
  + Keyboard, Mouse.
  + Architecture: 32-bit or 64-bit

### 3.4.2 Software Requirements

* + Python 3.5 in Jupyter Notebook is used for data pre-processing, model training and prediction.
  + Operating System: windows 7 and above or Linux based OS or MAC OS
  + Visualization : Jupyter Notebook
  + Dataset : .csv file
  + Libraries : matplotlib, pandas, numpy, seaborn, sklearn

### 3.4.3 Functional Requirements

Functional requirements describe what the software should do (the functions). Think about the core operations. Because the “functions” are established before development, functional requirements should be written in the future tense.

**Python Library**

A Python library is a reusable chunk of code that you may want to include in your programs/ projects. The Python Standard Library is a collection of exact syntax, token, and semantics of Python. It comes bundled with core Python distribution.

**Pandas:**

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation.

**Matplotlib:**

Matplotlib is used for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers 10 for plotting as it provides features to control line styles, font properties, formatting axes, etc.

**Numpy:**

NumPy is a Python library for scientific computing that facilitates operations on large, multi-dimensional arrays and matrices. It includes mathematical functions and tools for tasks like linear algebra, statistical analysis, and more.

**Key Features:**

* **Array Operations:** Efficient handling of arrays and matrices.
* **Mathematical Functions:** Extensive set of mathematical functions.
* **Broadcasting:** Allows operations on arrays of different shapes.

**Seaborn:**

Seaborn is a statistical data visualization library built on Matplotlib. It simplifies the creation of attractive statistical graphics and provides color palettes for improved visualization.

**Key Features:**

* **Statistical Plots:** Easy creation of statistical plots.
* **Color Palettes:** Aesthetic color palettes for visual appeal.
* **Integration with Pandas:** Seamless integration with Pandas DataFrames.

**Scikit-learn:**

Scikit-learn is a popular machine learning library for Python that provides simple and efficient tools for data analysis and modeling. It is built on NumPy, SciPy, and Matplotlib and provides a wide array of machine learning algorithms for tasks such as classification, regression, clustering, and dimensionality reduction.

### 3.4.4 Non-Functional Requirements

1. **Performance:**

* The system should provide low-latency responses, especially for real-time adaptability and data visualization.
* It must handle large datasets efficiently, ensuring optimal performance during modeling and analysis.

1. **Scalability:**

* The system should be scalable to accommodate an increasing volume of data and user interactions.
* It must support scalability in terms of both computational resources and storage.

1. **Reliability:**

* The system must be reliable, ensuring minimal downtime and consistent performance.
* It should implement failover mechanisms to handle potential disruptions.

# CHAPTER 4

**METHODOLOGY**

## 4.1 Problem Definition

This project focuses on understanding and addressing global warming through data analysis and prediction. By collecting and preprocessing extensive climate data, the study aims to uncover patterns and correlations, particularly in greenhouse gas concentrations. Predictive modeling techniques, including machine learning algorithms, will be employed to forecast Earth's surface temperature and assess the impact of reducing greenhouse gas emissions.

## 4.2 Data Collection

Data collection is one of the important and basic things in our project. The right dataset must be provided to get robust results. Our data mainly consists of earth surface temperature dataset from kaggle.com, global population data and global urban population data from www.johnstonsarchive.net and carbon dioxide concentration data from esrl.noaa.gov.

## 4.3 Data Preprocessing

Human can understand any type of data but machine can’t our model will also learn from scratch so it’s better to make the data more machine readable. Raw data is usually inconsistent or incomplete. Data preprocessing involves checking missing values, splitting the dataset and training the machine etc.

## 4.4 Models Implementation

### 4.4.1 SARIMAX

SARIMAX(Seasonal Auto-Regressive Integrated Moving Average with exogenous factors) is an updated version of the ARIMA model. we can say SARIMAX is a seasonal equivalent model like SARIMA and Auto ARIMA. it can also deal with external effects. This feature of the model differs from other models.

One shorthand notation for SARIMA models is:



where *p* = non-seasonal autoregressive (AR) order, *d* = non-seasonal differencing, *q*= non-seasonal moving average (MA) order, *P* = seasonal AR order, *D* = seasonal differencing, *Q* = seasonal MA order, and *S* = length of repeating seasonal pattern. We will use this notation from now on. By adding those seasonal AR and seasonal MA components, SARIMA solves the seasonality problem.

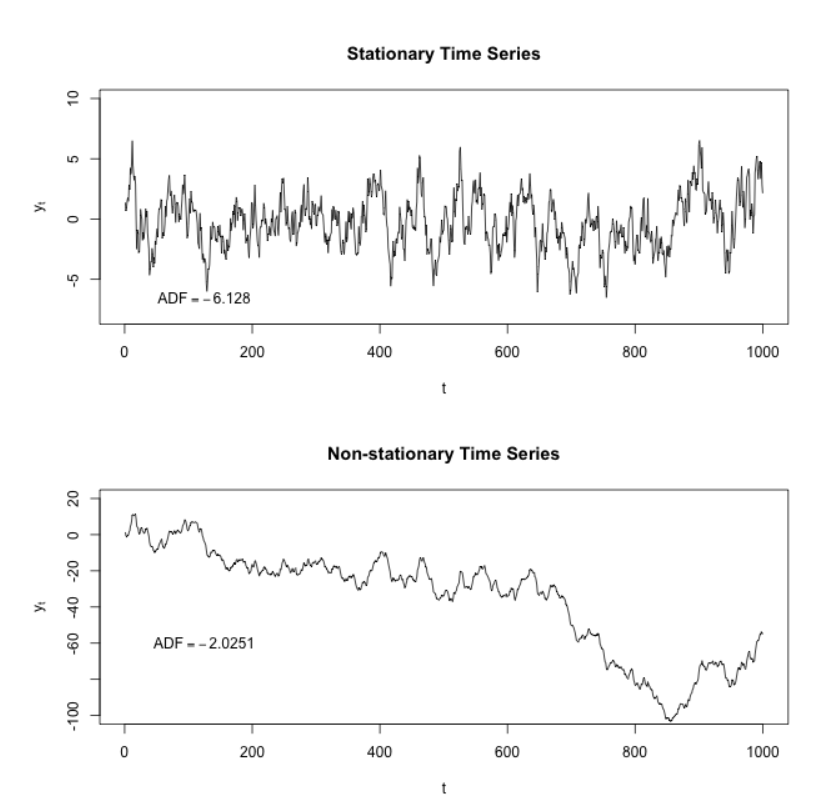
The SARIMAX implementation we use in the project also comes from the package statsmodels.

***Lag***

Lags are simply delays in time steps within a series. Consider a time index *t,*the lag 1 time index with respect to *t*is simply *t-1,*lag 2 is *t-2,*and so on.

***Stationarity***

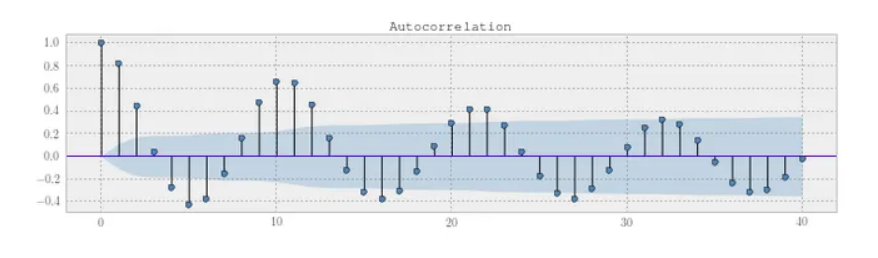
A stationary time series is one that has its mean, variance and autocorrelation structure unchanging overtime. In other words, it does not have any cycle/trend or seasonality. The ARMA models family is actually built on this concept.



***Autocorrelation function (ACF) and Partial Autocorrelation function (PACF)***

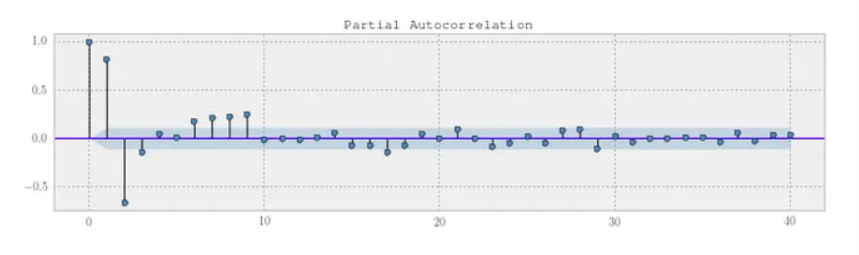
The ACF plot is a graphical representation of the correlation of a time series with itself at different lags. The correlation coefficient is a measure of how closely two variables are related. A correlation coefficient of 1 indicates a perfect positive relationship, while a correlation coefficient of -1 indicates a perfect negative relationship. A correlation coefficient of 0 indicates no relationship between the two variables.

The ACF plot can be used to identify the order of an AR model. The order of an AR model is the number of lags that are included in the model. The ACF plot will show spikes at the lags that are included in the model.



**Partial Autocorrelation Function (PACF)**

The PACF plot is a graphical representation of the correlation of a time series with itself at different lags, after removing the effects of the previous lags. The PACF plot can be used to identify the order of an MA model. The order of an MA model is the number of lags that are included in the model. The PACF plot will show spikes at the lags that are included in the model.



Both of these functions measure how correlated the data at time *t*is to its past values *t-1,t-2,…* There is one crucial difference, however. The ACF also measures indirect correlation up to the lag in question, while PCAF does not. In practice, their plots are vital for many tasks, especially choosing the parameters for the SARIMAX model.

### 4.4.2 ARIMAX

An Autoregressive Integrated Moving Average with Exogenous Variable (ARIMAX) model can be viewed as a multiple regression model with one or more autoregressive (AR) terms and/or one or more moving average (MA) terms. This method is suitable for forecasting when data is stationary/non stationary, and multivariate with any type of data pattern, i.e., level/trend /seasonality/cyclicity.

ARIMAX is related to the ARIMA technique but, while ARIMA is suitable for datasets that are univariate (see the article, entitled’ What is ARIMA Forecasting and How Can it Be Used for Enterprise Analysis?’). ARIMAX is suitable for analysis where there are additional exogenous variables (multivariate) in categorical and/or numeric format.

**ΔPt =c+βX+ϕ1 ΔPt-1 + θ1 ϵt-1+ϵt**

For starters, P t and P t-1 represent the values in the current period and 1 period ago respectively.

Similarly, ϵ t and ϵ t-1 are the error terms for the same two periods. And, of course, c is just a baseline constant factor.

The two parameters, ϕ 1 and θ 1, express what parts of the value P t-1 and error ϵ t-1 last period are relevant in estimating the current one.

Now, the two new additions to the model are “X” and its coefficient β. Just like ϕ, β is a coefficient which will be estimated based on the model selection and the data. But what about X?

**Exogenous Variable**

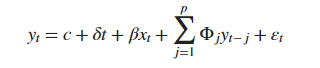
1. Well, X is the exogenous variable and it is also known as external or independent variables, are factors that influence the time series being studied but are not affected by the time series itself. These variables are considered to be independent of the time series and are often introduced into the analysis to improve the accuracy and predictive power of the model.

Exogenous variables can help explain the variation in the dependent variable (the time series) that cannot be accounted for by its own past values.

### 4.4.3 Vector Auto Regression (VAR)

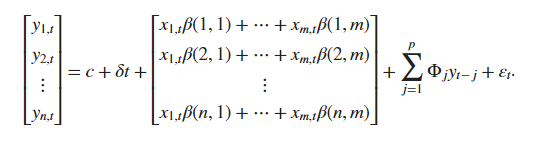
A *vector autoregression (VAR) model* is a multivariate time series model containing a system of *n* equations of *n* distinct, stationary response variables as linear functions of lagged responses and other terms. VAR models are also characterized by their degree *p*; each equation in a VAR(*p*) model contains *p* lags of all variables in the system.

The general equation for a VARX(*p*) model is



where

* *xt* is an *m*-by-1 vector of observations from *m* exogenous variables at time *t*. The vector *xt* can contain lagged exogenous series.
* *β* is an *n*-by-*m* vector of regression coefficients. Row *j* of *β* contains the regression coefficients in the equation of response series *j* for all exogenous variables. Column *k* of *β* contains the regression coefficients among the response series equations for exogenous variable *k*. This figure shows the system with an expanded regression component:

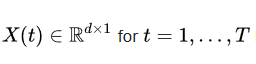


**Granger Causality test**

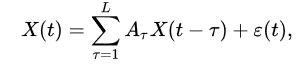
The Granger causality test is conducted to determine whether one time series is useful in forecasting another. A time series *X* is said to Granger-cause *Y* if it can be shown, usually through a series of t-tests and F-tests on lagged values of *X* (and with lagged values of *Y* also included), that those *X* values provide statistically significant information about future values of *Y*.

**Multivariate analysis**

Here, for multivariate Granger causality analysis performed by fitting a VAR to the time series. Considering below is a d-dimensional multivariate time series —



Granger causality is performed by fitting a VAR model with L time lags as follows:



where ε ( t ) is a white Gaussian random vector, and A τ is a matrix for every τ. A time series X i is called a Granger cause of another time series X j, if at least one of the elements A τ ( j , i ) for τ = 1 , … , L is significantly larger than zero.

**Lag order selection**

I have implemented Akaike’s Information Criteria (AIC) through the VAR (p) to determine the lag order value. In the fit function, I have passed a maximum number of lags and the order criterion to use for order selection.

### 4.4.4 Vector Error Correction Model (VECM)

Vector Error Correction Mechanism (VECM) models are a special application of VAR or Vector Autoregressive Models. The specification of VECM models involves the introduction of error correction terms into the VAR models. VECM methodology is used if the variables in the system have a long-run relationship, that is, they are cointegrated.

Every VAR model can be specified in the form of VECM by differencing the variables and introducing error correction terms. However, VECM is used only in the presence of cointegrating or long-run relationships. If there is no cointegration or if the variables are stationary, the VAR model should be applied.

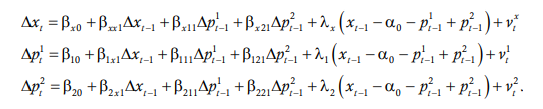
VECM imposes additional restriction due to the existence of non-stationary but co-integrated data forms. It utilizes the co-integration restriction information into its specifications. After the cointegration is known then the next test process is done by using error correction method. Through VECM we can interpret long term and short term equations. We need to determine the number of co-integrating relationships. The advantage of VECM over VAR is that the resulting VAR from VECM representation has more efficient coefficient estimates.

**Cointegration**

Non-stationary variables are said to be cointegrated if their linear combination is stationary. For example, if two variables in a system are I(1) or integrated of order 1, but their linear regression yields an error term that is stationary. In such a case, we can say that the two variables are cointegrated and have a long-run relationship. Such variables move together over time and share a common stochastic trend.

The VECM model is used to analyze cointegrated variables or cointegrating relationships. It provides a mechanism to understand the long-run as well as short-run behaviour of the variables in the system.

We could estimate a VEC system (with one lag, for simplicity) for the evolution of the three variables x, p1 , and p2 with one cointegrating relationship (with some known coefficients):



For those familiar with linear algebra, the term “rank” refers to the rank of a matrix characterizing the dynamic system. If a dynamic system of n variables has r cointegrating relationships, then the rank of the matrix is n – r. This means that the matrix has r eigenvalues that are zero and n – r that are not. The Johansen tests are based on determining the number of nonzero eigenvalues

Johansen proposed several related tests that can be used at each stage. The most common (and the default in Stata) is the trace statistic. The Stata command vecrank prints the trace statistic or, alternatively, the maximum-eigenvalue statistic (with the max option) or various information criteria (with the ic option).

Once the VEC system has been estimated, we can proceed to calculate IRFs and variance decompositions, or to generate forecasts as we would with a VAR.

### 4.4.5 Model Validation

Model validation is a crucial step in assessing the performance and reliability of predictive models. It involves evaluating how well a model generalizes to new, unseen data. The primary goal is to ensure that the model's predictions are accurate and reliable when applied to real-world scenarios.

**Key Steps:**

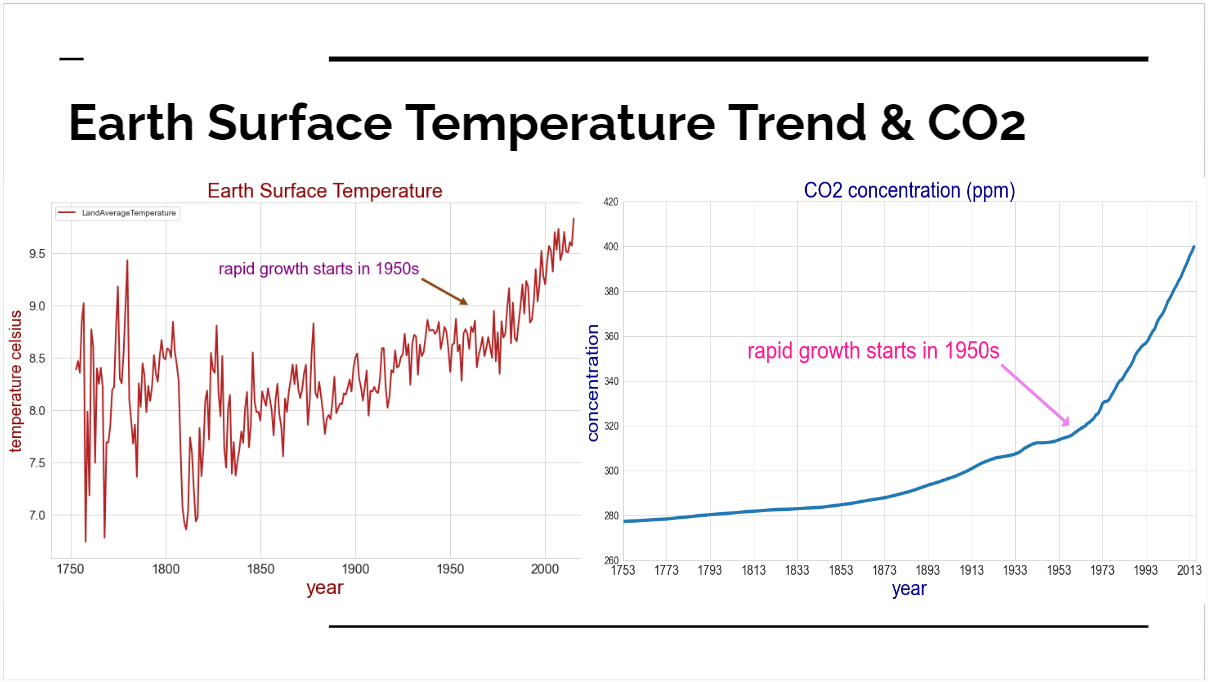
1. **Train-Test Split:**
   * The dataset is typically divided into two subsets: a training set used to train the model and a test set reserved for validation. Common splits include 70-30 or 80-20, ensuring an adequate amount of data for training and evaluation.
2. **Cross-Validation:**
   * For more robust validation, techniques like k-fold cross-validation can be employed. The dataset is divided into k subsets, and the model is trained and validated k times, with each subset serving as the test set once.
3. **Performance Metrics:**
   * Various metrics are used to assess model performance, depending on the nature of the problem. Common metrics include accuracy, precision, recall, F1 score for classification problems, and mean absolute error, mean squared error, or R-squared for regression problems.
4. **Overfitting and Underfitting Detection:**
   * Model validation helps identify overfitting (model performs well on training data but poorly on new data) and underfitting (model is too simple to capture the underlying patterns). This guides adjustments to model complexity.
5. **Hyperparameter Tuning:**
   * Model hyperparameters, which are settings not learned from data (e.g., learning rates in machine learning algorithms), can be fine-tuned based on validation results to improve model performance.

# CHAPTER 5

**RESULTS AND EVALUATION**

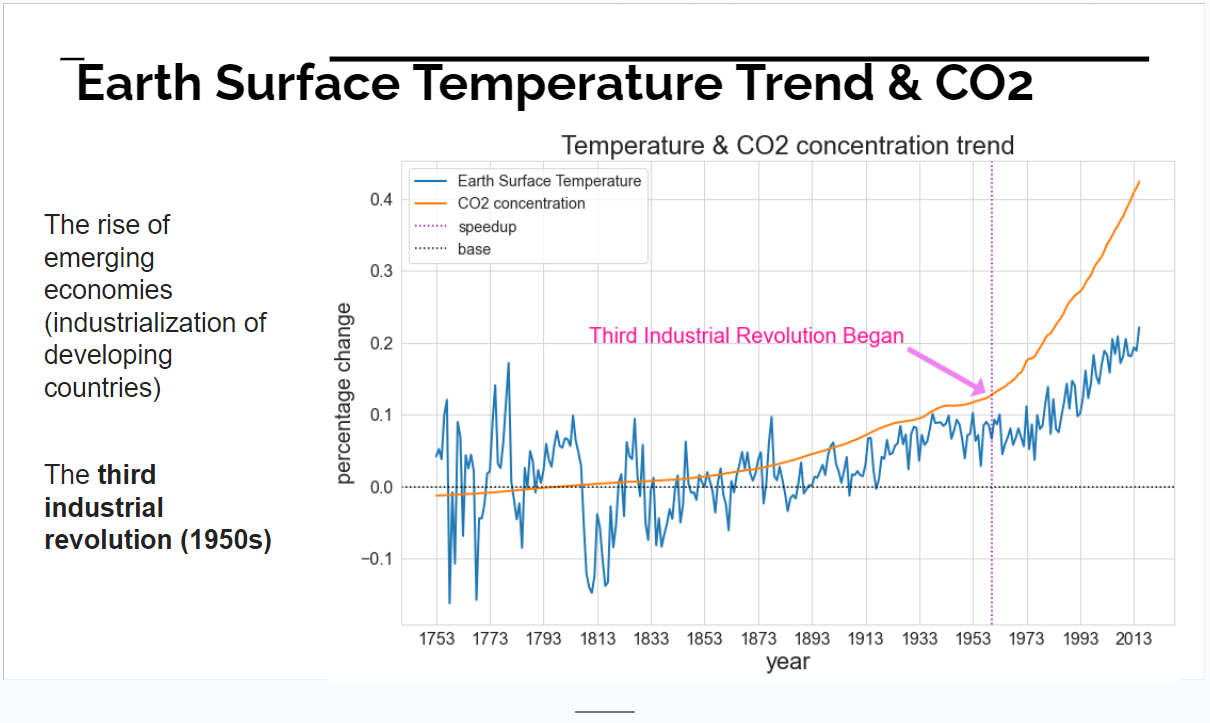
This is the main part for any implementation of the project. Evaluation of the key point to the success. All the categorization of the work and the best to know the resource fundaments and again establishing the same to check the validity and the work flow and check on the output is must. The evaluation, utilization and implementation undergo a various level of extraction and evaluation.

The main theme is to provide and come up with the output with an accuracy that can be used and implemented. From the starting to the final the process is categorized, supervised and efficiency is check and the working is undergone. Testing is tested and it’s evaluation are mentioned.

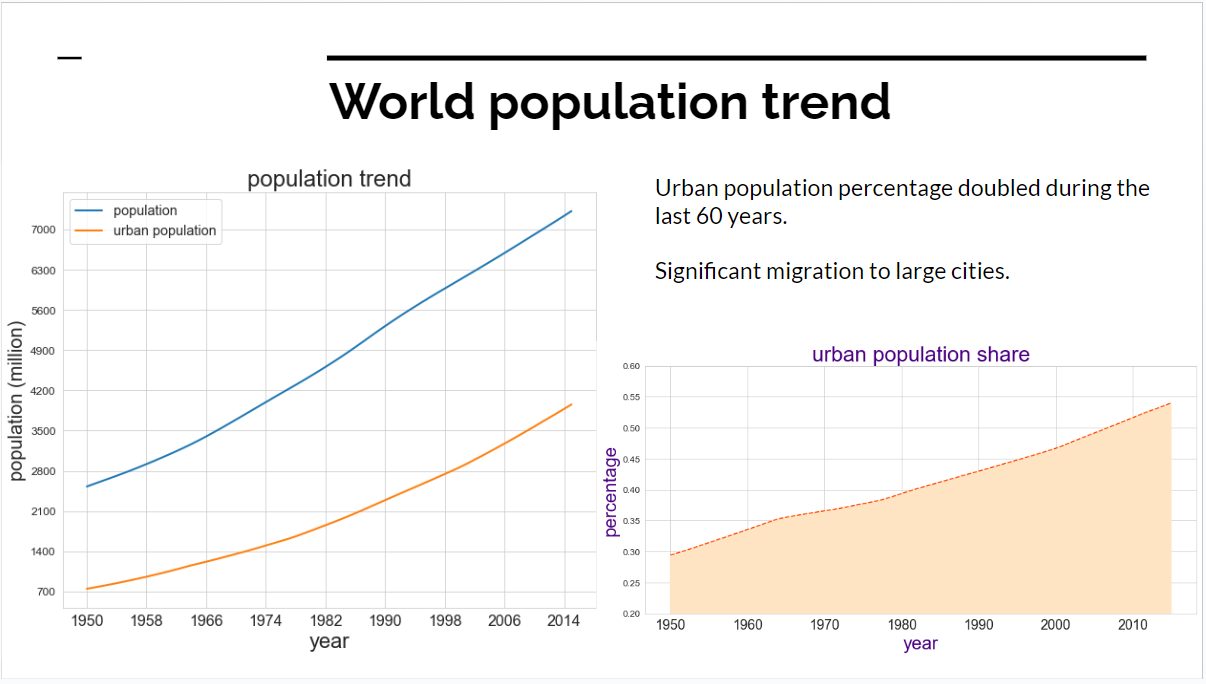
The process undergoes the same for various time and phase. Testing of the same undergoes sequential iteration for many more to meet up to the constituency. 

As we can see from the above plot, there is a clear trend showing global land average temperature is raising from 18th century to now. We can also see that before 1950s, temperature increase was slow for 200 years, ever since 1950s, rapid growth started.

In this plot, the carbon dioxide concentration in air also increases from 18th century to now. And it is very clear that the ascending rate keeps raising after the end of 19th century, especially after 1950s, we can see significant raise in ascending rate.



As shown in this plot, both the Earth Surface Temperaturere rising and CO2 concentration rising are speeding up around 1950 - 1960, when Third Industrial Revolution began. I believe CO2 concentration must have influence on the global warming.



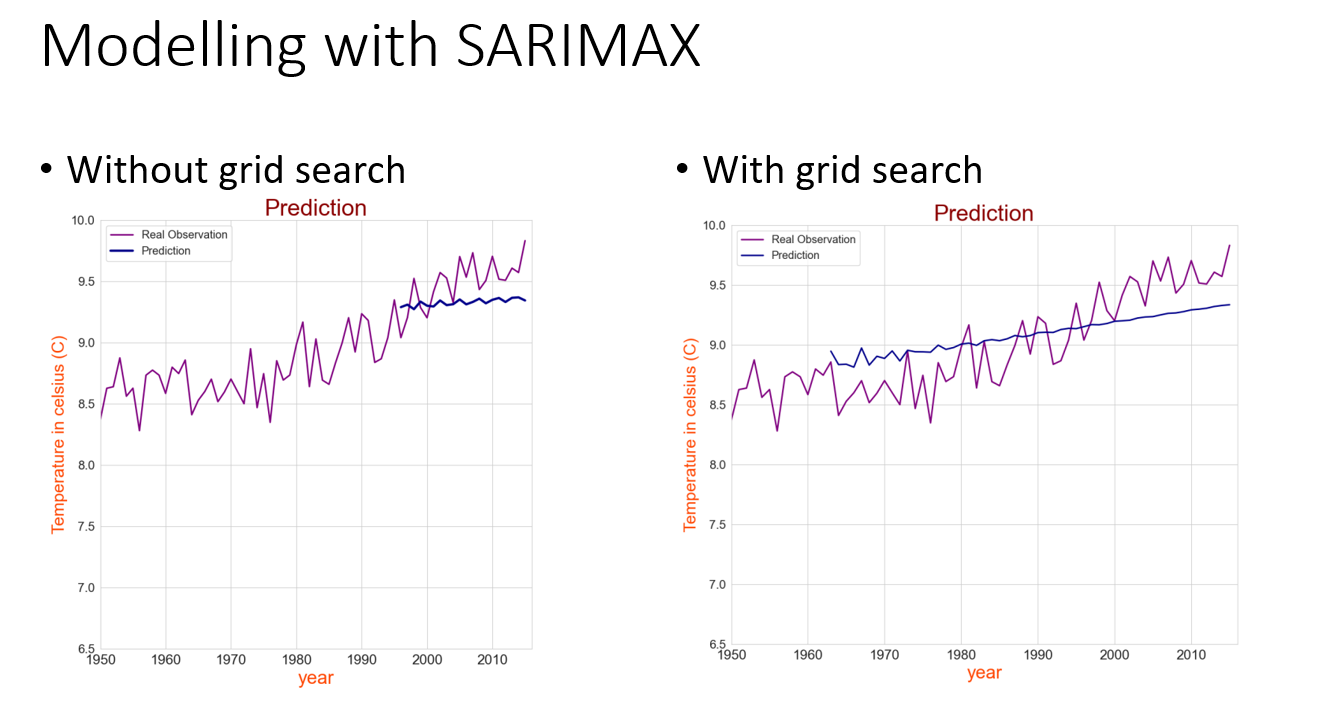
As we can see from the above plot, the world total population and urban population have been both steadily growing during past 70 years. **Ever since Third Industrial Revolution** (after world war II in 1940s), many countries were experiencing period of restoration and heavy production.

But **Ever since Third Industrial Revolution**, many countries (especially developing countries) has been through modern industrialization and urbanization, people migrated to urban areas, thus produced much more polution and harmful gases compared to rural life.

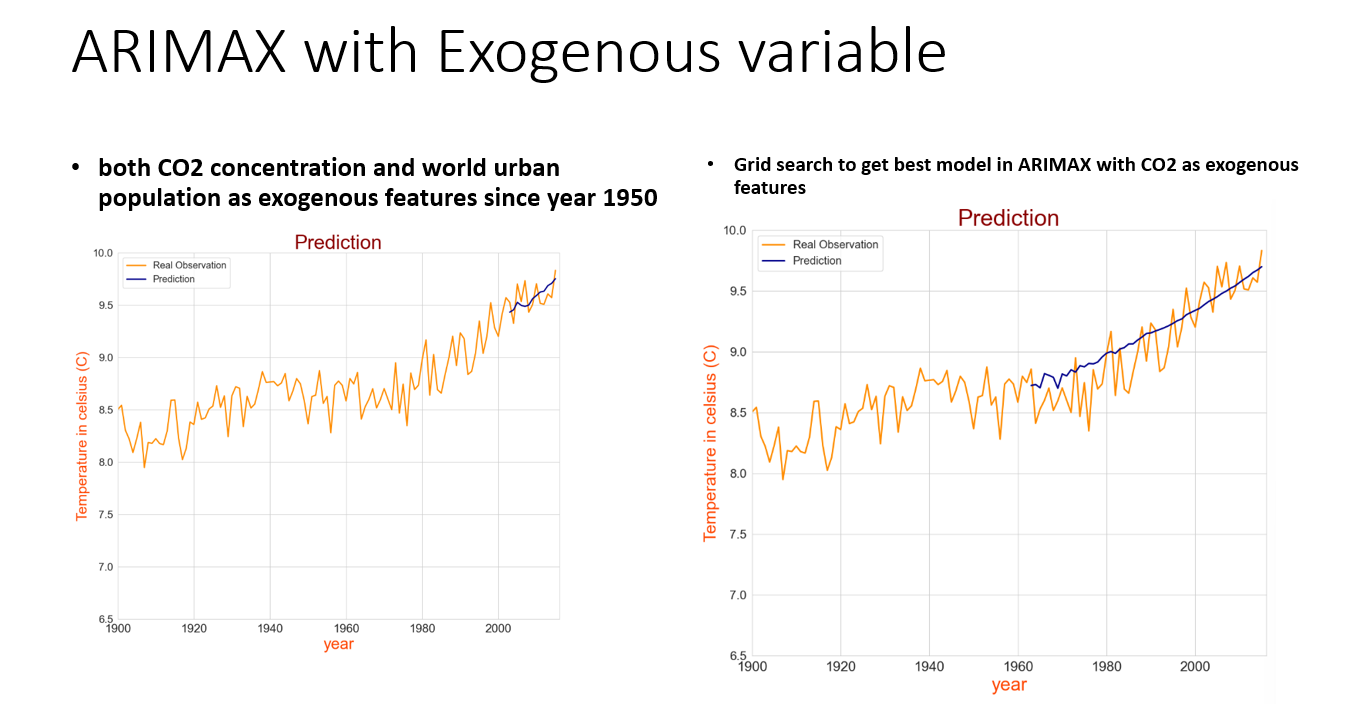
**Modelling with SARIMAX**

As we can see, when taking the whole series data from year 1753, SARIMAX model does not work better in yearly earth surface temperature prediction.

Using grid search of the hyperparameters, the best (p, d, q) combinations and MAE, RMSE, MSE results are shown in the result dataframe 'df\_sarimax'. The top are (3, 2, 15), (1, 2, 13), (7, 2, 12), (3, 2, 14) and (2, 2, 3), the RMSE error are similar for these orders. Best RMSE is 0.27, The above graph shows the prediction, definitely looks better than previous section which were not grid-search-selected model.

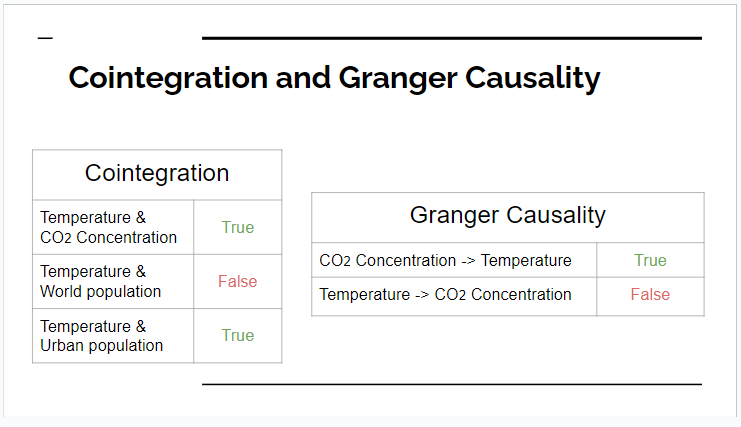


**Modelling with ARIMAX**



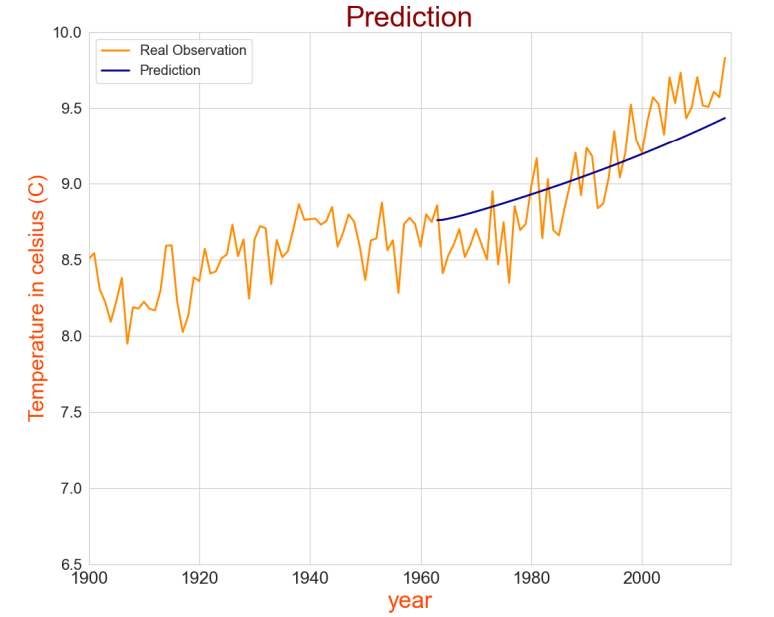
Comparing to the previous ARIMAX model with only CO2 exogenous feature, adding both CO2 and urban population features have reached similar minimum RMSE value but does improve the MAE, and this time I reached to the minimum MAE 0.09. Up to now, the ARIMAX model is very very accurate.

**Cointegration and granger Causality**

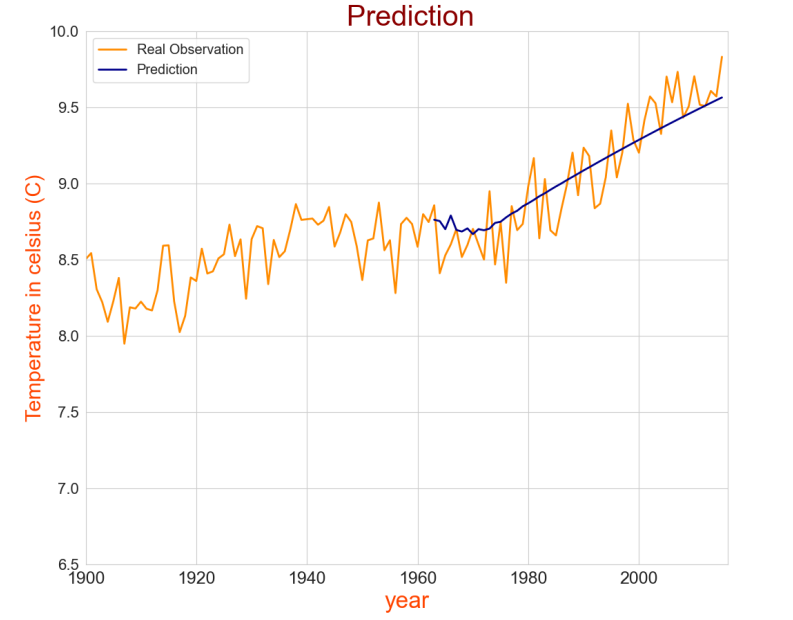
****

**VAR model with Earth surface temperature & CO2 concentration (multi-variables from year 1753)**

From the result in this section, we can see the prediction with Vector Auto Regression is not as good as (ARIMAX with exogenous features). Both RMSE and MAE are not as good as arimax. So this confirmed that my judgement that exogenous features will do better than Multivariate Time Series.

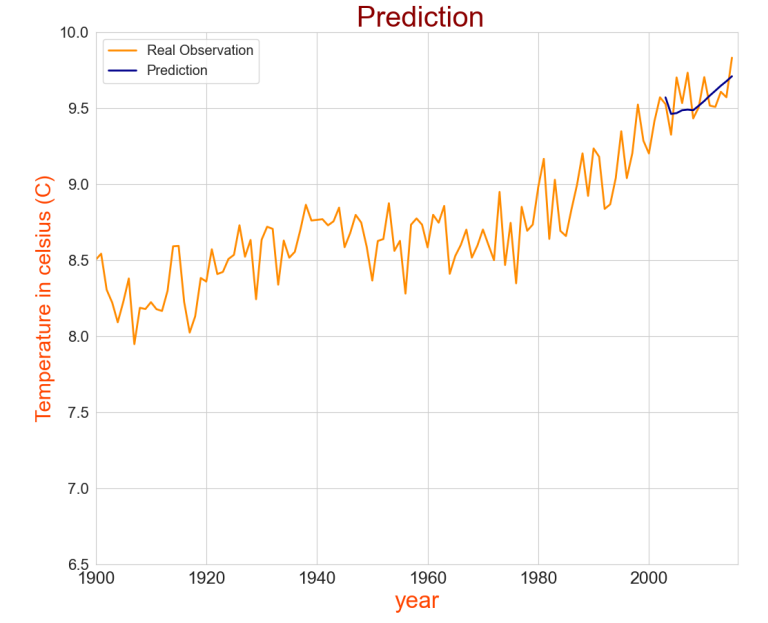


**VECM with Earth surface temperature & CO2 concentration**



From this VECM model (temperature with CO2 concentration) results, we can see that the RMSE and MAE are almost as good as the section (ARIMAX with exogenous features). The prediction graph derived by VECM model is good although not as good as ARIMAX. The VECM model gives much better results than VAR model

**VECM with Earth surface temperature & CO2 concentration & urban population from year 1950**

.

The result using VECM with temperature, CO2 concentration and urban population in VECM model give very accurate estimation of the 13 years earth surface temperature forecast (2003 -2015). The result is almost as good as the result in Section(ARIMAX models with exogenous features). This again verifies that the VECM model works much better than the VAR model in my case. Because VECM basically combines stationary levels and differences from all time series, so VECM mitigate the non-stationary influence and utilizes the cointegration between several non-stationary time series with similar trend. This is very important.

**Predicting Future Temperature**

Based on the results of the above models, The VECM, leveraging information on CO2 concentration and population, offers a precise and competitive forecasting capability for Earth's surface temperature compared to the ARIMAX model. So that using this model to predict the future temperature

# CHAPTER 6

**CONCLUSIONS AND FUTURE WORKS**

In this project, 4 different models are used to forecast the earth surface temperature. They are SARIMAX, ARIMAX, VAR, VECM. Also, grid search of hyperparameters are used for all of the models to find out best performance hyperparameters. Evaluation metrics used are RMSE and MAE, and, best forecast of each model are ploted and analyzed. The best model is ARIMAX(with exogenous variables. For 13 years forecasting (2003 - 2015), best results are (RMSE:0.13, MAE:0.09) and (RMSE:0.12, MAE:0.11). For 53 years forcasting (1963 - 2015), the best results are (RMSE:0.18, MAE:0.15). VECM models gives almost same good results, (RMSE:0.13, MAE:0.10) for 13 years forecasting and (RMSE:0.19, MAE:0.15) for 53 years forcasting. Comparing to the annual-mean real observations of the earth surface temperatures from 2013 to 2015, 9.58°C, this error is very tiny. Also, prediction graphes looks perfect in section 6(ARIMAX) and 9(VECM). Thus, the prediction models are successful.

Cointegration analysis and Granger causality analysis are done for time series of temperature, CO2 and population. Analysis show that CO2 concentration in air has strong influence to the earth surface temperature change (it is not scientifically rigorous to conclude that CO2 concentration cause the global warming, but from my statistical analysis, CO2 concentration greatly influenced temperature raising). When it comes to population, analysis in this project show that world population does not directly influence the earth surface temperature, but the world's urban population has influence to earth surface temperature change.

This point is important and makes perfect sense. Since carbon dioxide emission are mainly generated by industrial activities, human activities (daily life consumption, power generation industry, steel industry, car-driving, etc). Of course the urban people consumes much more energy and generate much more harmful garbage than rural people. There are more cars in metropolitan areas, more air-conditioning, more heat vapors, etc.

**FUTURE WORKS**

In future, this project aims to Enhance the integration of diverse climate datasets, including advanced satellite imagery, oceanic data, and atmospheric composition. Incorporate real-time data sources to capture dynamic environmental changes. Explore cutting-edge machine learning techniques and algorithms for more accurate predictive models. Investigate the integration of deep learning architectures and advanced ensemble methods to capture complex climate interactions. Extend the analysis to regional and local scales, considering specific geographical and socio-economic factors. Tailor predictions to address unique challenges faced by different regions.

# Appendix - A

**SAMPLE CODING**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from scipy import stats

import sklearn

from sklearn import metrics

from sklearn.metrics import mean\_absolute\_error

from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import r2\_score

from sklearn.model\_selection import GridSearchCV

from statsmodels.tsa.arima\_model import ARIMA

from statsmodels.tsa.arima\_model import ARMA

from statsmodels.tsa.stattools import adfuller

from statsmodels.tsa.vector\_ar.vecm import coint\_johansen

from statsmodels.tsa.vector\_ar.var\_model import VAR

from statsmodels.tsa.vector\_ar.vecm import VECM

import statsmodels.tsa.stattools as ts

from statsmodels.tsa.stattools import grangercausalitytests

import plotly

import statsmodels.api as sm

from pmdarima import model\_selection

import pmdarima as pm

import fbprophet

from pandas.api.types import is\_string\_dtype

from pandas.api.types import is\_numeric\_dtype

%matplotlib inline

sm.tsa.statespace.SARIMAX

import warnings

warnings.filterwarnings('ignore')

df = pd.read\_csv('GlobalTemperatures.csv')

df.head(5)

df.tail(5)

df.shape

df[['dt','LandAverageTemperature']].isna().sum()

df[['dt','LandAverageTemperature']].isna()[36:].sum()

df = df[36:]

# save original dataframe for future use

df1 = pd.read\_csv('GlobalTemperatures.csv')

type(df.dt.iloc[0])

df['date'] = pd.to\_datetime(df.dt)

df = df.set\_index('date')

df\_pop = pd.read\_csv('pop.csv', index\_col=0)

df\_ur = pd.read\_csv('ur.csv', index\_col=0)

df\_pop.tail()

df\_ur.tail()

df\_pop.shape, df\_ur.shape

df\_pop[['year','world']].isna().sum()

df\_ur[['year','urban','rural']].isna().sum()

df\_pop[['year','world']].dtypes

df\_ur[['year','urban','rural']].dtypes

df\_pop['date'] = pd.to\_datetime(df\_pop.year, format='%Y')

df\_pop = df\_pop.set\_index('date')

df\_ur['date'] = pd.to\_datetime(df\_ur.year, format='%Y')

df\_ur = df\_ur.set\_index('date')

# save original dataframe for future use

df\_pop1 = pd.read\_csv('pop.csv', index\_col=0)

df\_ur1 = pd.read\_csv('ur.csv', index\_col=0)

df\_com = pd.read\_csv('co2-mm-mlo\_csv.csv') # carbon dioxide data monthly

df\_coa = pd.read\_csv('co2-annmean-mlo.csv') # carbon dioxide data annually

df\_com.head(5)

df\_coa.head(5)

df\_com[['Date','Average']].isna().sum()

df\_coa[['Year','Mean']].isna().sum()

type(df\_com.Date[0]), type(df\_coa.Year[0])

df\_com['date'] = pd.to\_datetime(df\_com.Date)

df\_com = df\_com.set\_index('date')

df\_coa['date'] = pd.to\_datetime(df\_coa.Year)

df\_coa = df\_coa.set\_index('date')

df\_com[df\_com.Average < 313]

df\_coa[df\_coa.Mean < 315]

ave1 = np.mean([df\_com.loc['1963-11-01']['Average'],df\_com.loc['1963-12-01'

df\_com.loc['1964-01-01']['Average'],df\_com.loc['1964-05-01'

df\_com.loc['1964-06-01']['Average'],df\_com.loc['1964-07-01'

df\_com.loc['1964-02-01','Average'] = ave1

df\_com.loc['1964-03-01','Average'] = ave1

df\_com.loc['1964-04-01','Average'] = ave1

ave1 = np.mean([df\_com.loc['1975-09-01']['Average'],df\_com.loc['1975-10-01'

df\_com.loc['1975-11-01']['Average'],df\_com.loc['1976-01-01'

df\_com.loc['1976-02-01']['Average'],df\_com.loc['1976-03-01'

df\_com.loc['1975-12-01','Average'] = ave1

ave1 = np.mean([df\_com.loc['1984-01-01']['Average'],df\_com.loc['1984-02-01'

df\_com.loc['1984-03-01']['Average'],df\_com.loc['1984-05-01'

df\_com.loc['1984-06-01']['Average'],df\_com.loc['1984-07-01'

df\_com.loc['1984-04-01','Average'] = ave1

df\_com = df\_com[10:]

# save the original dataframe for future use

df\_com1 = pd.read\_csv('co2-mm-mlo\_csv.csv')

df\_coa1 = pd.read\_csv('co2-annmean-mlo.csv')

df\_coh = pd.read\_csv('co2history.csv')

df\_coh1 = pd.read\_csv('co2history.csv') # save an original dataframe for us

df\_coh.tail(5)

df\_coh[['year','data\_mean\_global']].isna().sum()

is\_numeric\_dtype(df\_coh['year']), is\_numeric\_dtype(df\_coh['data\_mean\_global

dfy = pd.DataFrame(df.groupby(df.index.year).LandAverageTemperature.mean())

sns.set\_style('whitegrid')

dfy.plot(figsize=(12,8), linewidth=2, color='firebrick')

plt.xlabel('year',fontsize=24, color='darkred')

plt.ylabel('temperature celsius', fontsize=20, color='darkred')

plt.xticks(fontsize=16)

plt.yticks(fontsize=15)

plt.title('Earth Surface Temperature', fontsize=24, color='darkred')

plt.annotate('rapid growth starts in 1950s', xy=(1960,9.0), xytext=(1828,9.

color='purple', fontsize=20, arrowprops=dict(facecolor = 'sadd

dfy[:100].mean()

len(df\_coh['data\_mean\_global'][1753:]), len(np.arange(1753,2015))

df\_coh['data\_mean\_global'][1753:][:100].mean()

plt.figure(figsize=(18,10))

df\_coh['data\_mean\_global'][1753:].plot(linewidth=5)

plt.xlim([1753,2016])

plt.ylim([260,420])

plt.xticks(np.arange(1753,2016,20),fontsize=20);

plt.xlabel('year',fontsize=32, color='darkblue')

plt.ylabel('concentration', fontsize=30, color='darkblue')

plt.title('CO2 concentration (ppm)', fontsize=33, color='darkblue')

plt.yticks(fontsize=16);

plt.annotate('rapid growth starts in 1950s', xy=(1958,320), xytext=(1810,35

arrowprops=dict(facecolor='violet',width=6,headwidth=24), font

df\_co1753 = df\_coh['data\_mean\_global'][1753:]

plt.figure(figsize=(13,8));

plt.plot((dfy - dfy[:100].mean())/(dfy[:100].mean()), linewidth=2);

plt.plot((df\_co1753 - df\_co1753[:100].mean())/(df\_co1753[:100].mean()), lin

plt.xticks(np.arange(1753,2015,20),fontsize=16);

plt.xlabel('year',fontsize=22)

plt.ylabel('percentage change', fontsize=20)

plt.yticks(fontsize=13)

plt.title('Temperature & CO2 concentration trend', fontsize=24)

plt.yticks(fontsize=16);

plt.axvline(1960, lw=1.5, c='darkmagenta', label='speedup', linestyle=':')

plt.axhline(lw=1.5, c='k', label='baseline', linestyle=':')

plt.legend(('Earth Surface Temperature','CO2 concentration','speedup','base

loc='upper left', fontsize='x-large');

plt.annotate('Third Industrial Revolution Began', xy=(1958,0.13), xytext=(1

arrowprops=dict(facecolor='violet',width=6,headwidth=24), font

from statsmodels.graphics.tsaplots import plot\_acf

from statsmodels.graphics.tsaplots import plot\_pacf

# plot autocorrelation for each lag (alpha is confidence interval) to deter

# yearly temperature for recent 57 years autocorrelation and partial autoco

xxx = plot\_acf(dfy1959, alpha=.05)

yyy = plot\_pacf(dfy1959, alpha=.05)

# yearly temperature from 1753 autocorrelation and partial autocorrelation

xxx = plot\_acf(dfy, alpha=.05)

yyy = plot\_pacf(dfy, alpha=.05)

# monthly temperature for recent 57 years autocorrelation and partial autoc

xxx = plot\_acf(dfm1959, alpha=.05)

yyy = plot\_pacf(dfm1959, alpha=.05)

# monthly temperature from year 1753 autocorrelation and partial autocorrel

xxx = plot\_acf(dfm1753, alpha=.05)

yyy = plot\_pacf(dfm1753, alpha=.05)

dfy1959\_train = dfy1959.LandAverageTemperature[:round(len(dfy1959)\*0.8)]

dfy1959\_test = dfy1959.LandAverageTemperature[round(len(dfy1959)\*0.8):]

len(dfy1959\_test)

# use p, d, q, value indicated in the autocorrelation and partial autocorre

m\_arima1959 = ARIMA(dfy1959\_train, order=(6,1,2))

m\_arima\_fit1959 = m\_arima1959.fit(disp=1)

print(m\_arima\_fit1959.summary())

# the prediction of earth surface temperature from 2005 to 2015

m\_arima\_fit1959.forecast(11)[0]

print('MSE:', mean\_squared\_error(dfy1959\_test, m\_arima\_fit1959.forecast(11)

print('RMSE:', np.sqrt(mean\_squared\_error(dfy1959\_test, m\_arima\_fit1959.for

print('MAE:', mean\_absolute\_error(dfy1959\_test, m\_arima\_fit1959.forecast(11

dfy1959.plot(figsize=(8,6), linewidth=2, color='firebrick')

plt.plot(range(2005,2016), m\_arima\_fit1959.forecast(11)[0], linewidth=2, co

plt.ylim([7,10])

plt.xlabel('year',fontsize=20, color='orangered')

plt.ylabel('temperature celsius', fontsize=18, color='orangered')

plt.xticks(fontsize=15)

plt.yticks(fontsize=12)

plt.title('Temperature Prediction', fontsize=20, color='darkred')

plt.legend(('Real Observation', 'Prediction'), loc='upper left', fontsize='

dfy\_train = dfy.LandAverageTemperature[:round(len(dfy)\*0.8)]

dfy\_test = dfy.LandAverageTemperature[round(len(dfy)\*0.8):]

df\_co\_train = df\_co1753[:round(len(df\_co1753)\*0.8)]

df\_co\_test = df\_co1753[round(len(df\_co1753)\*0.8):]

len(dfy\_test), len(df\_co\_test)

orderlist1 = []

mselist1 = []

rmselist1 = []

maelist1 = []

for p in range(1,16):

for d in range(1,3):

for q in range(1,16):

ordernow = (p,d,q)

m\_arimax\_exo\_now = sm.tsa.statespace.SARIMAX(endog = dfy\_train,

initialization='ap

orderlist1.append(ordernow)

foca\_now = m\_arimax\_exo\_now.forecast(steps=53, exog=df\_co\_test)

maelist1.append(mean\_absolute\_error(dfy\_test, foca\_now))

rmselist1.append(np.sqrt(mean\_squared\_error(dfy\_test, foca\_now)

mselist1.append(mean\_squared\_error(dfy\_test, foca\_now))

df\_arimax\_exo = pd.DataFrame()

df\_arimax\_exo['order'], df\_arimax\_exo['mse'], df\_arimax\_exo['rmse'], df\_ari

df\_arimax\_exo.sort\_values('rmse').head()

arimax\_exo\_best = sm.tsa.statespace.SARIMAX(endog = dfy\_train, exog=df\_co\_t

dfy.plot(figsize=(12,10), linewidth=2, color='firebrick')

plt.plot(range(1963,2016), arimax\_exo\_best.forecast(steps=53, exog=df\_co\_te

plt.ylim([6.5,10])

plt.xlabel('year',fontsize=24, color='orangered')

plt.ylabel('temperature celsius', fontsize=22, color='orangered')

plt.xticks(fontsize=18)

plt.yticks(fontsize=16)

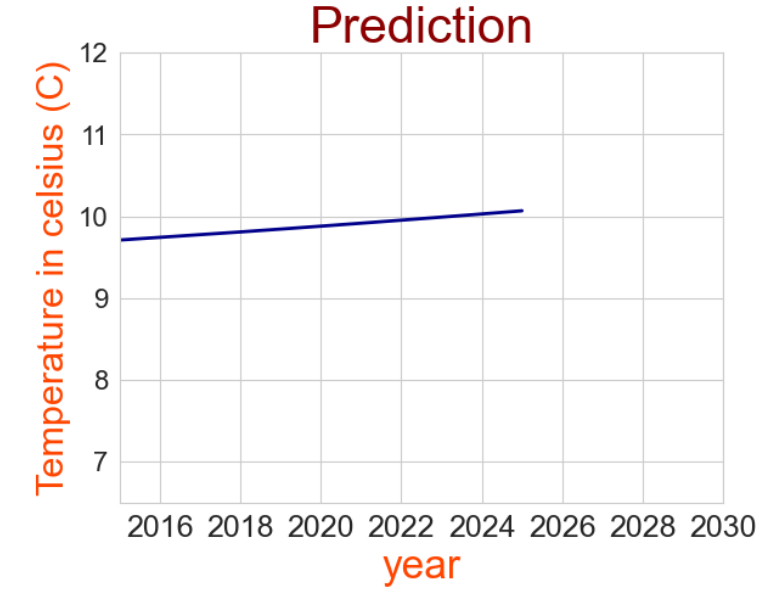
plt.title('Temperature Prediction', fontsize=30, color='darkred')

plt.legend(('Real Observation', 'Prediction'), loc='upper left', fontsize='x-large'

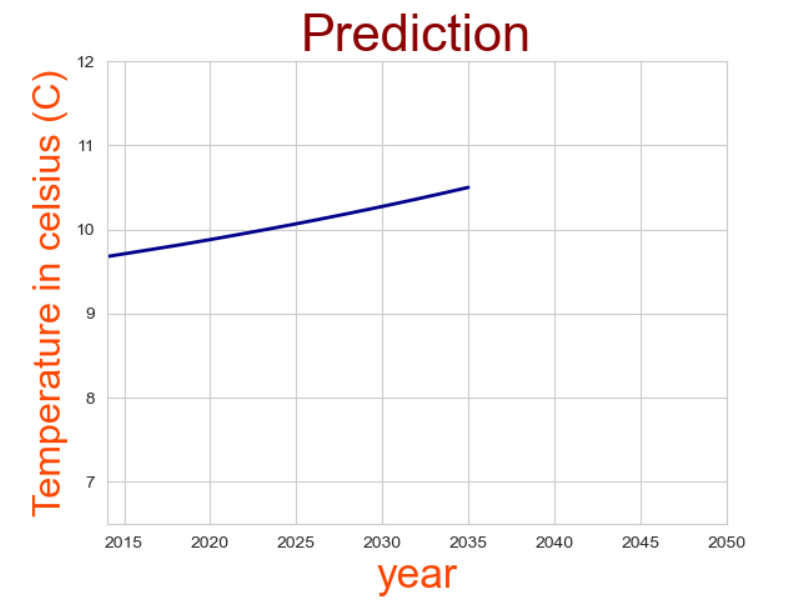
# APPENDIX - B

**OUTPUT**

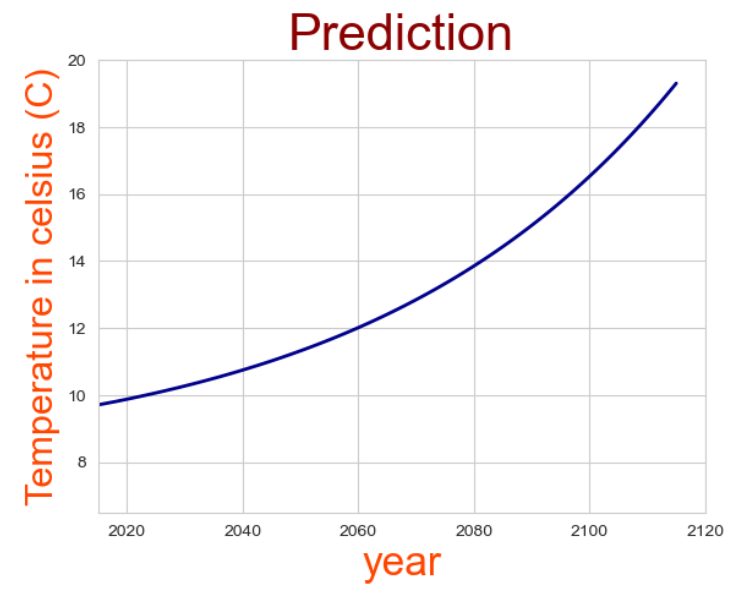
**a) forecast for 10 years**

****

**b) Forecast for 20 years**

****

**c) Forecast for next 100 years**

****

**Therefore, The mean Earth surface temperature is going to reach 10.06 degrees by 2026, 10.49 degrees by 2036 and 19.30 degrees by in next 100 years.**

# REFERENCES

**[1]** Zheng, H. (2018) Analysis of Global Warming Using Machine Learning. *Computational Water, Energy, and Environmental Engineering*, 7(3), pp. 127-141.

**[2]** Navaneetha Krishnan M, Ranjith R, Lavanya B. (2022) Climate Change Prediction Using ARIMA Model. Volume 10, Issue VI, June 2022 in International Journal for Research in Applied Science & Engineering Technology.

**[3]** Himanshu Vishwakarma.(2020) Climate Change Analysis Using Machine Learning. International Journal of Science and Research (IJSR), Volume 9 Issue 8, August 2020 .

**[4]** M. Purushotham Reddy, A. Aneesh, K. Praneetha and S. Vijay, "Global Warming Analysis and Prediction Using Data Science," *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2021, pp. 1055-1059, doi: 10.1109/I-SMAC52330.2021.9640944.

**[5] "**GLOBAL WARMING PREDICTION USING MACHINE LEARNING" by Adwait Prakash Mishra, Umesh Pratap Singh. International Research Journal of Modernization in Engineering Technology and Science, Volume:03, Issue:07/July-2021.

**[6]** Global Warming Prediction in India using Machine Learning” by D. Deva Hema, Anirban Pal, Vineet Loyer, Rajeev Gaurav International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019

**[7]** Song, J., Tong, G., Chao, J. *et al.* Data driven pathway analysis and forecast of global warming and sea level rise. *Sci Rep* 13, 5536 (2023).

**[8]** Leon Wang , Leigh Wang , Yang Li , John Wang. A century-long analysis of global warming and earth temperature usinDecision Analytics Journal, Volume 7, June 2023, 100237.

**[9]** Ms. Nisha Bairagee, Mrs. Nitima Malsa, Dr. Jyoti Gautam. Prediction on Global Warming. International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869 (O) 2454-4698 (P), Volume-5, Issue-3, July 2016.

**[10]** Rutvij Wamanse, Tushuli Patil in Analysis of various climate change parameters in India using machine learning.SciPy India Conference 2021, IIT Bombay