With the support of the
Erasmus+ Programme
of the European Union

An Explorative Study on Greenhouse Gas Emissions & Polar Ice Cap Melting

NAFIL MAHMUD
ID: 77361293
Data Analytics and Visualisation

Supervisor
Dr. Ah-Lian Kor
School of Built Environment, Engineering, and Computing
Leeds Beckett University,England
May, 2023

Abstract

Greenhouse gas emission is one of the most dangerous threats to the existence of life on Earth. Continuous expansion of industrial evolution makes earth like a chimney of releasing gases like carbon dioxide, methane, nitrous oxide etc. The temperature gets warmer and as a result global warming is caused. One of the major effects of global warming is that polar ice caps are melting rapidly. In this research emissions from last three decades and ice-covered areas are studied to figure out the correlation between them and tried to find a common predictive model for them. Five machine learning techniques are compared as well as three deep learning techniques and found satisfactory results in terms of predicting greenhouse gas emissions and ice-covered areas of earth in a year. For Predicting Greenhouse gas emissions MLP algorithm works best while for predicting ice-covered areas of the world random forest algorithm outperforms other algorithms. Descriptive statistical and inferential statistical analysis is also used to find informative trends in the data used. The total research was done keeping in mind 2 of the 17 sustainable development goals which are climate change(SDG 13) and life below water(SDG 14) as greenhouse gas emission affects both of them.

Keywords: Greenhouse, GHG, Ice-Caps, Northern Hemisphere, Southern Hemisphere, Machine Learning, Deep Learning, Inferential Analysis, Descriptive Analysis, Data Mining.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Full Form
SDG	Sustainable Development Goals
CO_2	Carbon dioxide
MLP	Multi-layer Perceptronl
GHG	Greenhouse Gas
CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
CPU	Central Processing Unit
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
SVR	Support Vector Regression
RCP	Representative Concentration Pathways
CSV	Comma-Separated Values
H_0	Null Hypothesis
H_a	Alternate Hypothesis

1 Introduction

1.1 Background of Context

Air is an integral part of the existence of every species on Earth. Without air, the world would turn into vacuum and make the atmosphere disappear. As a result, air quality is a matter of concern for public health as well as the environment. The air we breathe contains various pollutants like nitrogen oxide, particulate matter sulpher dioxide which have dangerous effect on human health and natural ecosystem (Jorgenson 2007). In addition to that greenhouse gas (GHG) like carbon dioxide, methane and nitrous oxide is a significant contributors of climate change which is threatening to sustainable lifestyle (Tuckett 2019). Green-house gases trap heat in the earths atmosphere which has severe consequences like temperature increase, rising sea level, change in weather pattern etc which have adverse impacts over biodiversity and natural ecosystem (James G Titus et al. 1991a). Recent observation clearly illustrates that ice cap is not in equilibrium and out of balance with the current climate (James G. Titus et al. 1991b).

1.2 Rationale

GHG gas emission is one of the leading causes of climate change and is severely affecting our world. As the concentration of greenhouse gas increases in the atmosphere, the earth's surface temperature increases (Manabe 2019). Greenhouse gases like carbon dioxide, nitrogen dioxide, methane etc are directly related to the increase of temperature around the world. Reducing the emission of greenhouse gases is now more of a necessity than just a concern. Many of the world's leading countries are taking steps to reduce it which should be promoted more by making proper guidelines. For that, we need a vast analysis of GHG emission trends of countries and identify the cause of origin to take proper steps.

Ice cap melting is a result of global warming. The continuous sea level rise is a major problem for both land and sea ecosystems. Extinction of species is more of a threat now more than ever and loss of sea ice has increased the risk of diseases in marine biota (Prakash 2021). The polar ice caps are continually decreasing every year and there are lots of data to show this trend. According to research data about 75% of the ice mass loss has been recorded in Antarctica and Greenland in the last 10 years and resulted in 0.3mm increase in sea level per year(Baqi, Abbas, and Imran 2021). So, it is high time to take deep dive into the relation of GHG and ice cap melting and make an informed analysis.

1.3 Problem Statement

Since the Industrial revolution the world has seen rapid changes in terms of creativity and pushing boundaries for the betterment of humankind. In the process of this humans have compromised the effects it has on the environment. Until recently people of woken up to the severity of the threat that discarding the environment from the picture brings. Extensive greenhouse gas has so many adverse effects on the environment that not giving it proper attention will have catastrophic consequences. There are data available on the internet where how much GHG gas is emitted each year. That is a great place to start analysis about this topic.

One of the major problem we face nowadays is ice cap melting as a result of global warming. Sea level rising trend has become alarming in recent time as the temperature increased globally is melting the ice in the polar region. Data about the ice covered area in sea is available on the internet that goes back 30 years at least. Analyzing the relation of this change with respect to GHG emissions will give important insights.

According to (Encyclopaedia Britannica 2021) Sustainable development is an approach where economic growth is planned keeping the environment in mind and to preserve its qualities as much as possible. In other words, it is stated as when present development is made without compromising the ability of growth of the future. United Nations **website** states there are 17 goals for sustainable development which are explained in detail what each of those goals is trying to achieve.SDG 13 deals with limiting climate change and its impact. It has 5 targets by 2030 which are clearly given in the website(United Nations Sustainable Development Goals n.d.[a]). SDG 14 is about Life Below Water and is the highest goal of SDG (Lee, Noh, and Khim 2020). As sea levels are rising as the ice cap melts, the ecosystem is changing rapidly and there are a lot of species that are endangered by this. GHG emissions make it hard for animals to survive on land too which falls under the goals of SDG 15.(United Nations Sustainable Development Goals n.d.[b])

1.4 Aim and Research Objectives

The research aims to find relations between polar ice area and GHG emissions with following research objectives(RO).

1.4.1 Broad Research Objectives

Broad research objectives of the study are -

- Analyse Greenhouse gas emissions from across the world in the last 3 decades. We collected the raw data from (Ritchie, Roser, and Rosado 2020) and reshape the data to fit the research.
- Analyse the relation between greenhouse gas emission and polar ice cap melting. Polar ice-covered area data is taken from National Snow and Ice Data Center (Fetterer et al. 2017) and do manipulation as necessary.

1.4.2 Level 1 Research Objectives: Descriptive Statistical Analysis

Level 1 Research Objective is to analyze different GHG emissions of countries over the world in the last three decades around the world as well as the amount of ice-covered area using descriptive statistical analysis and view the data with different graphs.

1.4.2.1 RO1.1

-Find the top ten countries emitting the most carbon dioxide in the world and compare it with a threshold value.

1.4.2.2 RO1.2

- Find top ten countries emitting the most carbon dioxide per capita to get a view of the most dense carbon-emitting countries.

1.4.2.3 RO1.3

- Descriptive analysis(Mean, Standard Deviation, lower-quartile, upper-quartile etc.) of the 1st and 10th in the last 3 decades which emitted greenhouse gas the most and compare the results.

1.4.2.4 RO1.4

- Conduct a Descriptive Analysis of the total ice-covered area in the south and north hemisphere in the last 32 years and show trend using a graph.

1.4.3 Level 2 Research Objectives: Inferential Statistical Analysis

Level 2 research objective is to analyze the correlation between different gas components attributed to total greenhouse gas emissions on a continental basis. A sufficient T-Test is done with hypothesises about the chosen features.

1.4.3.1 RO2.1

- Conduct T-test to seek the correlation between production-based carbon dioxide and total greenhouse emissions in Asia, Europe, Africa, and Australia continent.

1.4.3.2 RO2.2

- Conduct T-test to seek the correlation between production-based carbon dioxide and coal carbon dioxide emissions in Europe, Africa continent.

1.4.3.3 RO2.3

- Conduct T-test to seek the correlation between land use change carbon dioxide and coal carbon dioxide emissions in Africa continent.

1.4.3.4 RO2.4

- Conduct T-test to seek the correlation among oil sector carbon dioxide and gas carbon dioxide emissions with coal carbon dioxide emissions in Europe.

1.4.3.5 RO2.5

- Conduct T-test to seek the correlation between oil sector carbon dioxide and gas carbon dioxide emissions in North America.

1.4.3.6 RO2.5

- Conduct T-test to seek the correlation of methane and land use change carbon dioxide emissions and oil sector carbon dioxide with production-based carbon dioxide with each pair taken separately in South America continent.

1.4.3.7 RO2.6

- Conduct T-test to seek the correlation of methane and oil sector carbon dioxide emissions with coal sector carbon dioxide each pair taken separately in Australia continent.

1.4.3.8 RO2.7

- Analyze the correlation between different pollutants of greenhouse gas to see if they are correlated to each other or not.

1.4.4 Level 3 Research Objectives: Machine Learning

This level is dedicated to use different machine learning regression model to predict total greenhouse gas emitted as well as ice covered area from the same parameters and analyse the accuracy of the model.

1.4.4.1 RO3.1

- Use of multi-linear regression to predict the values of total greenhouse gas emissions using year, country and total production based carbon dioxide column.

1.4.4.2 RO3.2

- Adding the column land use change carbon dioxide into the regresson model of RO3.1 and compare the accuracy of two scenario.

1.4.4.3 RO3.3

- Use of multi-linear regression to predict the values of total ice covered area using year, country and total production based carbon dioxide and land use change carbon dioxide column.

1.4.4.4 RO3.4

- Using support vector regressor to do the experiments of RO3.2 and RO3.3.

1.4.4.5 RO3.5

- Using random forest algorithm to do the experiments of RO3.2 and RO3.3.

1.4.4.6 RO3.6

- Using adaptive boosting algorithm to do the experiments of RO3.2 and RO3.3.

1.4.4.7 RO3.7

- Using MLP algorithm to do the experiments of RO3.2 and RO3.3.

1.4.5 Level 4 Research Objectives: Deep Learning

Use of deep learning algorithms(RNN,CNN) to predict the total greenhouse gas emission and ice covered area and analyze the model loss.In this study, LSTM and GRU Recurrent Neural Network (RNN) algorithm and 1D CNN is used from Convolutional Neural Network.

1.4.5.1 RO4.1

- By applying LSTM and GRU is it possible to minimize the error with respect to level 3 models.

1.4.5.2 RO4.2

- Compare the efficiency of RO4.1 algorithms and 1D CNN that will be applied on the same data.

1.5 Contribution of the Research

Greenhouse gas emission and consequently polar ice cap melting is one of the existential threats the world is facing nowadays. To combat a global threat analysis like this study will give proper insight needed to take decisions. Through this research, the trend of past three decades will be clearly visible. This type of rigorous research on linking two different research areas is quite rare. This can be a template for people who wish to expand on this topic because the analysis is done thoroughly in four levels. The blend of statistical and machine learning analysis on this important topic is illustrated in this study.

1.6 Organization of Report

The report is organized in six chapters. First chapter gives a proper introduction to the problem with the aim and goal of the research.Second chapter is about the reviews of different literature on the topics and giving a proper image of the current research in the area. On the third chapter, the macro and micro methodology of the study is discussed while chapter four is discussing the experiments done for this research. Before giving conclusions on chapter six, some recommendations align with the topics given on chapter five.

2 Literature Review

2.1 Effects of Greenhouse Gas Emission

Greenhouse gases act as a layer in the atmosphere that holds the temperature inside the atmosphere. (Keating 2007) states that earth's atmosphere would be close to 40k(Kelvin) less warm if there was no greenhouse effect and used blanket and bed as a metaphor to explain the effect. The earth's natural greenhouse effect is critical for survival and diversity of life but since the industrial revolution greenhouse effect is altered to such an extent that it is no longer in the balance and it is affecting the global biochemical system (Casper 2010). Consequently, studies found that to avoid the worst possible effects of climate change GHG emission must be half of what it is now by 2030 and by 2050 reach about net zero(Hamilton et al. 2021). (Ward et al. 2013) found interesting relation between GHG in peat-lands and vegetation composition and said that vegetation composition is a significant modulator of greenhouse gas. In USA, diary sector contributes to greenhouse gas less than 2% where worldwide it is 2.7% (Wattiaux et al. 2019). Various steps has been taken to mitigate the emission of greenhouse gas. Utilization of bio-char in environment has had great success in recent times(C. Zhang et al. 2019).

2.2 Situation of Polar Ice Caps

Due to the continuous global warming earth ice area is continuously decreasing as stated by global climate model and The projection shows that sea ice will decline during the 21st century (Slater et al. 2021). Another study reveals that the average annual pole position is drifting toward the east which is which is states as abrupt drift direction and measurement by Gravity Recovery and Climate Experiment show that about 90% of this change is because of melting of polar ice and related rising of sea level (J. Chen et al. 2013). study reviews the active role of ice-covered area in the global carbon cycle which was thought to be an inert component for a long period of time but has a great influence on the greenhouse gas effect (Wadham et al. 2019). (Garbe et al. 2020) raises warning that world is lacking in the stability analysis of the Antarctic ice sheet for different levels of global warming and shows that there are some threshold beyond which ice loss is irreversible. This type of analysis should be seen with outmost importance cause this affects the whole existence of not only land animals but also marine living creatures. The mixing of fresh water with salt water is dangerous for organisms living in different ecosystems. A complete release of ice into the ocean would raise the global mean sea level by around six meters which will cause catastrophic floods in coastal areas (Oppenheimer 1998).

2.3 Prediction analysis for Greenhouse gas and Ice Caps

There is extensive research done to predict greenhouse gas emissions using different machine learning models. (Z. Hu et al. 2022) uses rough set theory to predict aquaculture wetlands emitted greenhouse gas which improves

the prediction accuracy and shortens the prediction time.(Comyns and Figge 2015) has done a statistical analysis between 1998 and 2010 to find any significant change in overall GHG reporting quality and found that quality has not improved. Study done in Iran where the objective was to predict wheat production GHG emission on the basis of energy inputs and the ANN models were developed using quality parameters. In Canada a research was done where GHG emission from agricultural field of 5 year time range were explored using different regression model and deep learning model and found that LSTM model worked the best with the scenario ("Machine learning for predicting greenhouse gas emissions from agricultural soils" 2020). worked on the the development of machine learning algorithm in filtering CO_2 reduction electrocatalyst over the past few years and explored artificial neural network, k-Nearest neighbors, decision tree, Kernel methods to train the properly (N. Zhang et al. 2021).

(Lösing and Ebbing 2021) applied the Gradient Boosted Regression method to analyze geothermal heat flow in Antarctica and found that it aligns with the findings but concluded the model deals with some uncertainties too. (Thapa et al. 2020)developed a model based on long short-term memory(LSTM) for snowmelt-driven discharge in a Himalayan basin that flows through riveres near the area and compare it with the autoregressive exogenous model (NARX), Gaussian process regression (GPR), and support vector regression (SVR) models. (Simonsen et al. 2021) worked with Greenland ice caps and used machine learning to convert radar-derived volume changes into radar-derived mass changes from 1992 to 2020.

3 Methodology

3.1 Macro Methodology

Data mining is the process of patterns, correlations, and anomalies in large datasets and is a crucial step in the process of knowledge discovery(Fayyad, Piatetsky-Shapiro, and Smyth 1996). The techniques of data mining helps discover hidden patterns and help to make informed choices. These techniques help in discovering hidden patterns, correlations, and anomalies in large datasets and are used to make informed decisions in different aspects of daily life (Wright 1998). (Azevedo and Santos 2008) viewed data mining as one of the phases of KDD process and compared SEMMA and CRISP-DM because of their popularity. A brief description of KDD process is giving below.

3.1.1 KDD Process

Knowledge discovery in database is a process of extracting valuable information from datasets according to the required specification with pre-processing, sub-sampling and transformation of database if necessary. There are five stages of KDD process:

1. **Selection:** This is where datasets are chosen or created.
2. **Pre-processing:** Data cleaning and imputing missing data to get a consistent dataset.
3. **Transformation:** According to the specification dimension reduction is used or another transformation method
4. **Data Mining:** This is when the patterns and hidden information are searched in the database.
5. **Evaluation :** The found information is interpreted and evaluated in this stage.

3.1.2 Dataset Selection

For the study after examining a sample of different datasets related to greenhouse gas emissions the one from (Ritchie, Roser, and Rosado 2020) was selected. This dataset has different emissions of different gases which contribute to the greenhouse effect like carbon dioxide from different source, nitrous oxide, methane according to countries. The data collection ranges from 1850 to 2021 and the values are calculated total and also in per capita unit. The dataset has some extra area specific value in addition to country data which will be dealt with in dataset pre-processing stage.

Polar ice caps melting is a major issue in this modern world. A considerable amount of research has been conducted to collect data from polar regions for analysis. By examining different datasets, one is selected from National Snow & Ice Data Center for our research (Fetterer et al. 2017). This dataset has ice-covered area in millions of square kilometers units of the Southern and Northern hemispheres separately. Some Transformation is and pre-processing is needed in this dataset to be fit for research.

3.1.3 Data Pre-Processing

The greenhouse gas emission dataset used in the study has values from the year 1850 to 2021. But for our research, the years 1990 to 2021 is taken for every country. other data is cleaned from the dataset. Moreover, the data set had some specific areas like Upper-middle-income countries, International transport, High-income countries, European Union (27) etc which only added noise to the actual data. Those rows were also cleaned from the dataset.

After cleaning there were some missing data which was imputed using the mean values. The mean calculation had some constraints. The cell that was missing the value was filled by the mean of that column taking only that specific country. Otherwise, there would be some anomaly if the cell was filled by the mean of the column considering every country.

3.1.4 Data Transformation

Now taking a deep dive into the features of our datasets it was required to do some dimension reduction by removing some of the columns. In the greenhouse gas emission dataset, there are total of 74 columns. But for the research 22 columns were chosen which have the most significant importance to the objectives and goals of the study.

For the dataset of the ice-covered area, the values are divided into northern and southern hemisphere data. A merged dataset was made by adding the both values making it a total ice covered area according to year.

For level 3, Both the datasets were merged according to year to have both greenhouse gas values and ice-covered area in one dataset, and empty rows were discarded if found.

3.2 Micro Methodology

The final dataset is analyzed in four different levels following the research objectives and goals

3.2.1 Level 1

Level 1 research particularly deals with the descriptive analysis of the dataset. Mean, median, standard deviation,lower-quartile, and upper-quartile will be the main focus of this level along with some graphs for better understanding.

3.2.2 Level 2

Inferential statistics will be done in this level. Different pairwise T- Tests will be done to test the hypothesises that are made. The correlation of different emissions will also be shown by the correlation matrix.

3.2.3 Level 3

Machine learning regression models will be used to predict the total greenhouse gas emission of a year and ice-covered area. For regression multi-linear regression, support vector regression, random forest regression, adaptive boosting regression and Multi-layer Perceptron(MLP) a regression will be done and compare which works best in terms the dataset.

3.2.4 Level 4

In this level deep learning algorithm such as Long-Short Term Memory or LSTM along with Gated Recurrent Unit or GRU and 1-Dimensional Convolutional Neural Network or CNN will be applied to find the possibility of better prediction results.

4 Finding and Discussion

After Preparing the dataset it had 22 columns and over 7000 rows. The dataset was ready to be used for the study.

```
Number of rows: 7041
Number of columns: 22
```

Figure 1: Shape of the greenhouse gas Dataset

On the other hand ice-covered area had 33 row and 33 column.

```
Number of rows in Total Ice Area Dataset: 33
Number of columns in Total Ice Area Dataset: 4
```

Figure 2: Shape of the Ice-Covered Area Dataset

4.1 LEVEL 1: Descriptive Statistical Analysis

Top 10 countries which emits the production carbon dioxide most is found out.It was seen that China emits the most carbon dioxide followed by USA. In summary of the IPCC policymakers report (Stocker 2014) it is stated that the current emission of greenhouse gases in carbon dioxide equivalents. The report also said that to have 50 percent chance of keeping the temperature less than 2 degree mankind can emit 1,212 Giga ton co2 from 2011 to rest of the century. In this research, this amount was divided into all the countries to find out how much can a country on average can emit GHG gas per year. The figure (figure 3) shows the top ten GHG gas-emitting countries and the average amount they should be emitting.

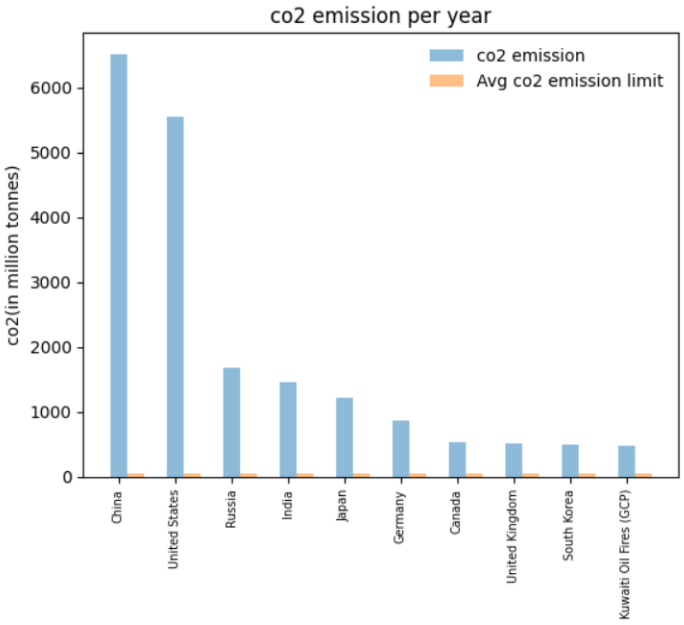


Figure 3: Avg CO_2 emission and The acceptable limit

Representative Concentration Pathways (RCPs) is a term used for scenarios that include time series of emissions and concentrations of greenhouse gases and aerosols and land use or land cover. It provides time-dependent estimation of greenhouse gas concentration taking different climate futures into consideration (Wayne 2014). According to the research current climate features is close to resembling RCP8.5.

Scenario	**Description**	**Cumulative Co2 can be emitted from 2012 to 2100(GtCO2)**
RCP 2.6	A scenrio that keeps the global temprature below 2 degree Celsius increase from pre industrial time	510 to 1505 giga ton
RCP 4.5	A scenario where various GHG mitigation policy is implemented	2180 to 3690 giga ton
RCP 6	A scenario where normal GHG mitigation scenario is implemented	3080 to 4585 giga ton
RCP 8.5	A scenario where GHG emission is very high and efforts are bein used to constrain emission	5185 to 7005 giga ton

Table 1: RCP Scenarios

Countries in the world has different areas in length. So total emission of co2 does not tell the full story. A country can be very big in area and consequently emits more GHG. Figure (figure 4) shows the top ten countries which emits the most ghg gas per capita.

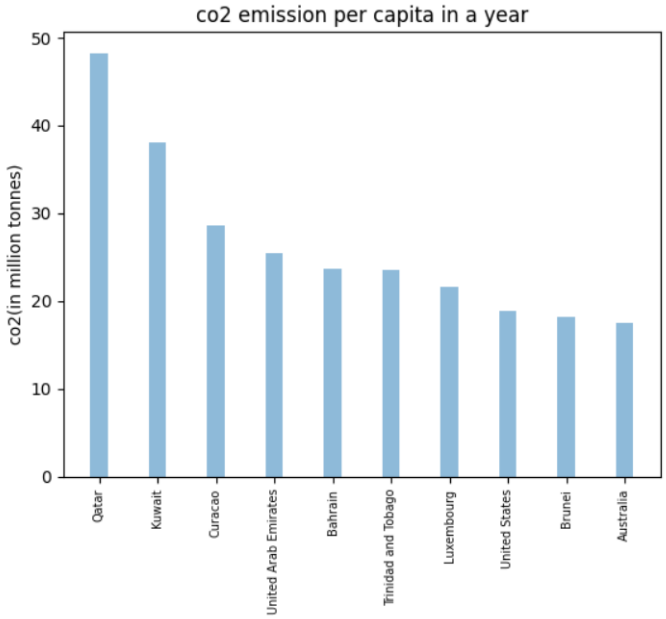


Figure 4: Avg CO_2 per capita

Figure 5 illustrates the countries that are leading the world terms of total greenhouse gas emission and compare if they are different from leading carbon dioxide emission countries.

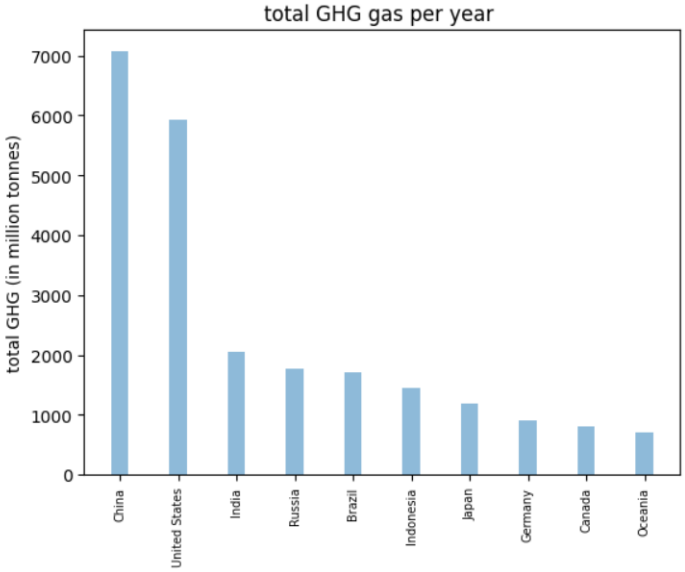


Figure 5: Leading Greenhouse Gas Emitting Countries

In our dataset, there are more than 200 countries. So for descriptive analysis, top country and tenth country in terms of greenhouse gas is taken which are China and Ocenia respectively to see the difference and how they compare in terms of mean, median, etc. After calculating it is seen that, the two countries have huge differences in means and standard deviation of greenhouse gas emissions.

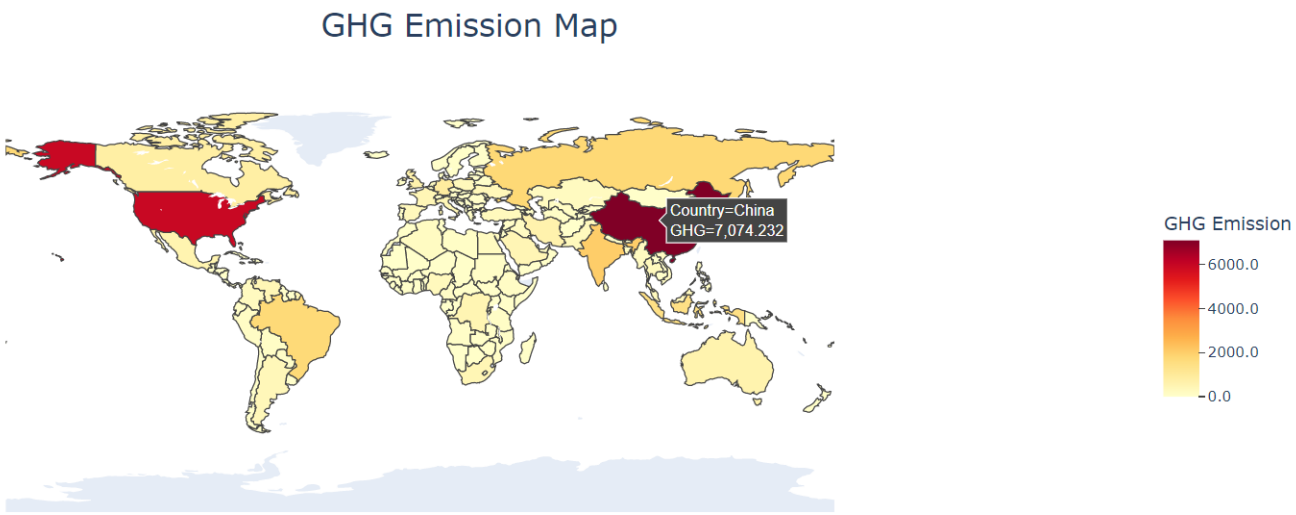


Figure 6: World Map of GHG Emission of the World

	north_hemisphere_ice	south_hemisphere_ice	total_ice
count	33.000000	33.000000	33.000000
mean	9.179424	11.657121	20.836545
std	0.475100	0.476877	0.726420
min	8.311000	10.647000	19.433000
25%	8.779000	11.410000	20.250000
50%	9.215000	11.687000	21.192000
75%	9.593000	11.961000	21.317000
max	10.086000	12.776000	21.861000

Figure 7: Descriptive Analysis of Ice-Covered Area

In the Ice-covered area dataset, Descriptive Analysis was done where it is seen that during the last 3 decades average ice-covered area is around 20 million sq. kilometers. Figure 8 shows that there was a sharp decline in the area just after 2015 which recovered somewhat in the Corona lockdown period but starts to decline again.

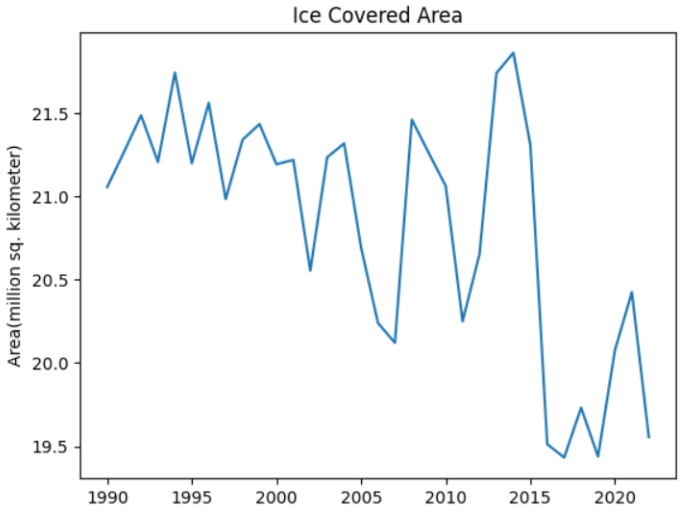


Figure 8: Ice Covered Area of The World

4.2 LEVEL 2: Inferential Statistical Analysis

For level 2, quite a few two sample T-test is done. Pre requirement of T-Test is that the two populations must have equal variance. According to (Zack April 2021), if the ratio of variance is less than 4 then it can

be said that the samples are approximately equal. Before doing T-test, the dataset was transformed to fit the research objectives. In level 2 data analysis was done by taking all the countries in a continent together. So, dataset countries and their values were sorted into separate continents.

4.2.1 T-Test

Every t-test of the study is done with a confidence level of 95% ie. the alpha value is 0.05 or 5%. The two hypotheses of every t-tests done here can be generalized by the following:

Null Hypothesis (H_0) : $\mu_1 = \mu_2$ (the mean of "SAMPLE_1" is equal to mean of "SAMPLE_2")

Alternate Hypothesis (H_a) : $\mu_1 \neq \mu_2$ (the mean of "SAMPLE_1" is not equal to mean of "SAMPLE_2")

4.2.1.1 T-Test between carbon dioxide emission of production-based sources and total greenhouse gas emission in Asia

The t-test shows that the null hypothesis is accepted and that the mean of these two samples is equal.

```
T test between co2 and total ghg gas in Asia
There are no zero values in the dataframe
The variance ratio is:  1.184023146329435
p value:  0.0542429047039973
t value:  -1.9256787930636554
Null Hypothesis Is Accepted
```

Figure 9: T-Test

4.2.1.2 T-Test between carbon dioxide emission of production-based sources and total greenhouse gas emission in Europe

The t-test shows that the null hypothesis is accepted and that the mean of these two samples is equal.

```
T test between co2 and total ghg gas in Europe
There are no zero values in the dataframe
The variance ratio is:  1.156002613505089
p value:  0.3119993284718027
t value:  -1.0112308557586818
Null Hypothesis Is Accepted
```

Figure 10: T-Test

4.2.1.3 T-Test between carbon dioxide emission of production-based sources and total greenhouse gas emission in Africa

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between co2 and total ghg gas in Africa
There are no zero values in the dataframe
The variance ratio is:  1.7629708549190608
p value:  1.6697930485416337e-45
t value:  -14.375387729923323
Null Hypothesis Is Rejected
```

Figure 11: T-Test

4.2.1.4 T-Test between carbon dioxide emission of production-based sources and total greenhouse gas emission in Australia

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between co2 and total ghg gas in Australia
There are no zero values in the dataframe
The variance ratio is:  2.607797402625915
p value:  0.009940113017316843
t value:  -2.58391322108322
Null Hypothesis Is Rejected
```

Figure 12: T-Test

4.2.1.5 T-Test between carbon dioxide emission of production-based sources and carbon dioxide emission from coal-based sources in Asia

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between co2 and coal co2 in Asia
There are no zero values in the dataframe
The variance ratio is:  1.9376405613743652
p value:  2.9549412087996e-05
t value:  4.185596235884583
Null Hypothesis Is Rejected
```

Figure 13: T-Test

4.2.1.6 T-Test between carbon dioxide emission of production-based sources and carbon dioxide emission from coal-based sources in Africa

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between co2 and coal co2 in Africa
There are no zero values in the dataframe
The variance ratio is:  1.5893029414468904
p value:  1.801881317431634e-07
t value:  5.245468482788616
Null Hypothesis Is Rejected
```

Figure 14: T-Test

4.2.1.7 T-Test between carbon dioxide emission land use change sources and carbon dioxide emission from coal-based sources in Africa

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between land use change co2 and coal co2 in Africa
There are no zero values in the dataframe
The variance ratio is:  2.7631014643943703
p value:  0.003405810293273801
t value:  2.9336373165531135
Null Hypothesis Is Rejected
```

Figure 15: T-Test

4.2.1.8 T-Test between carbon dioxide emission oil industry sources and carbon dioxide emission from coal-based sources in Europe

The t-test shows that the null hypothesis is accepted and that the mean of these two samples is equal.

```
T test between coal and oil co2 in Europe
There are no zero values in the dataframe
The variance ratio is:  1.102679155843684
p value:  0.05914077505760557
t value:  1.8880170148817825
Null Hypothesis Is Accepted
```

Figure 16: T-Test

4.2.1.9 T-Test between carbon dioxide emission due to gas sources and carbon dioxide emission from coal-based sources in Europe

The t-test shows that the null hypothesis is accepted and that the mean of these two samples is equal.

```
T test between coal and gas co2 in Europe
There are no zero values in the dataframe
The variance ratio is:  1.615047855577658
p value:  0.7887581596363937
t value:  -0.26795524134275717
Null Hypothesis Is Accepted
```

Figure 17: T-Test

4.2.1.10 T-Test between carbon dioxide emission due to gas sources and carbon dioxide emission from oil-based sources in North America

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between oil and gas co2 in North America
There are no zero values in the dataframe
The variance ratio is:  3.214865612692902
p value:  0.003203535906449318
t value:  2.9626328454316666
Null Hypothesis Is Rejected
```

Figure 18: T-Test

4.2.1.11 T-Test between carbon dioxide emission due to land use change sources and total greenhouse gas in South America

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between land use change co2 and total ghg in South America
There are no zero values in the dataframe
The variance ratio is:  1.2280258347804027
p value:  0.0004038182800784818
t value:  -3.553241164361917
Null Hypothesis Is Rejected
```

Figure 19: T-Test

4.2.1.12 T-Test between carbon dioxide emission due to production-based sources and total methane gas in South America

The t-test shows that the null hypothesis is accepted and that the mean of these two samples is equal.

```
T test between co2 and methane in South America
There are no zero values in the dataframe
The variance ratio is:  1.1100408772293138
p value:  0.5803296179313125
t value:  0.5531390120529461
Null Hypothesis Is Accepted
```

Figure 20: T-Test

4.2.1.13 T-Test between carbon dioxide emission due to production-based sources and oil bases sources in South America

The t-test shows that the null hypothesis is rejected and that the mean of these two samples is not equal.

```
T test between co2 and oil in South America
There are no zero values in the dataframe
The variance ratio is:  2.475164584414191
p value:  6.233272834756613e-06
t value:  4.550092657102079
Null Hypothesis Is Rejected
```

Figure 21: T-Test

4.2.1.14 T-Test between carbon dioxide emission due to coal-based sources and oil bases sources in Australia

The t-test shows that the null hypothesis is accepted and that the mean of these two samples is equal.

```
T test between coal and oil co2 in Australia
There are no zero values in the dataframe
The variance ratio is:  2.490234494403896
p value:  0.13173907026577356
t value:  1.5125219670792243
Null Hypothesis Is Accepted
```

Figure 22: T-Test

4.2.1.15 T-Test between carbon dioxide emission due to coal-based sources and methane in Australia

The t-test shows that the null hypothesis is accepted and that the mean of these two samples is equal.

```
T test between coal based co2 and methane in Australia
There are no zero values in the dataframe
The variance ratio is:  2.0763009470830958
p value:  0.7883420085713649
t value:  0.268882951595082
Null Hypothesis Is Accepted
```

Figure 23: T-Test

4.2.2 Co-relation Matrix

A correlation matrix (Figure 24) was made using the most important data columns for the research. It is seen that total greenhouse gas really correlates with production-based carbon dioxide, nitrous oxide, and coal. Methane emission very closely correlates with production-based carbon emission. Gas carbon emission is tightly connected to the oil-based carbon dioxide emission although In total greenhouse emission gas based co2 is relatively less important.

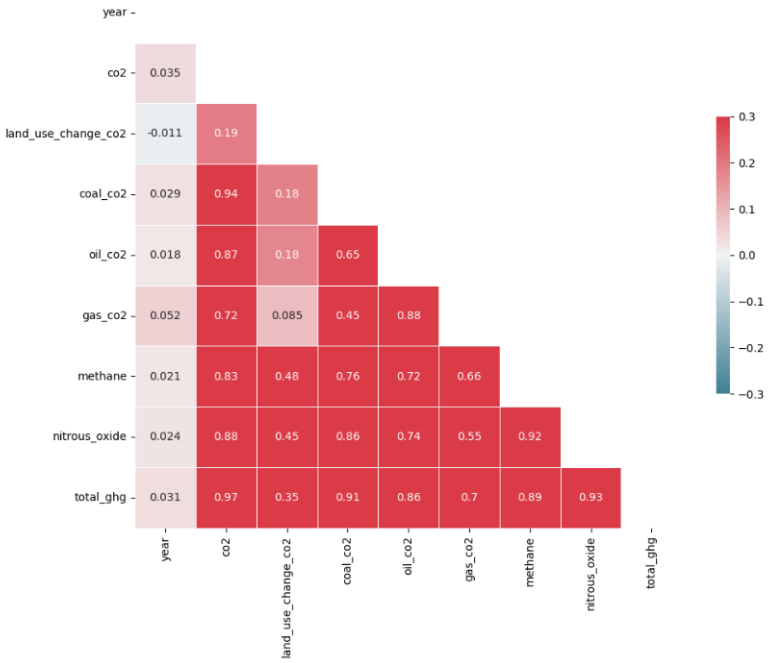


Figure 24: Correlation Matrix

4.3 LEVEL 3: Machine Learning

At this level, different machine learning algorithms to build models which will predict total greenhouse gas and ice-covered area. At first, the dataset is cleaned of any null values present there and sorted according to year. The country values are transformed using label encoding.

```
There are no zero values in the dataframe
Number of rows:  5618
```

Figure 25: No Null Values in Dataset in Level 3

Following the research objective of 3.1, using multi-linear regression using only 3 columns out of possible 22 columns, the model was trained to predict total greenhouse gas emissions. The model was around 94 percent accurate in this case(Figure 26).

```
------------------------------------------------------------
prediction of Total Greenhouse gas with multi linear regression
------------------------------------------------------------
                            OLS Regression Results
==============================================================================
Dep. Variable:              total_ghg   R-squared:                       0.947
Model:                            OLS   Adj. R-squared:                  0.947
Method:                 Least Squares   F-statistic:                 3.374e+04
Date:                Sun, 07 May 2023   Prob (F-statistic):               0.00
Time:                        00:42:22   Log-Likelihood:                -37135.
No. Observations:                5618   AIC:                         7.428e+04
Df Residuals:                    5614   BIC:                         7.431e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        596.4152    521.613      1.143      0.253    -426.148    1618.978
country       -0.3106      0.046     -6.784      0.000      -0.400      -0.221
year          -0.2594      0.260     -0.997      0.319      -0.769       0.251
co2            1.0690      0.003    317.938      0.000       1.062       1.076
==============================================================================
Omnibus:                     5844.959   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         13085055.448
Skew:                          -3.972   Prob(JB):                         0.00
Kurtosis:                     239.296   Cond. No.                     4.38e+05
==============================================================================
```

Figure 26: Linear Regression of Greenhouse gas With 3 Columns

In RO3.2, using the same model but adding one more column in this case the accuracy was found to be 97 percent(Figure 27).

```
------------------------------------------------------------
prediction of Total Greenhouse gas with multi linear regression
------------------------------------------------------------
                            OLS Regression Results
==============================================================================
Dep. Variable:              total_ghg   R-squared:                       0.974
Model:                            OLS   Adj. R-squared:                  0.974
Method:                 Least Squares   F-statistic:                 5.179e+04
Date:                Sun, 07 May 2023   Prob (F-statistic):               0.00
Time:                        00:43:17   Log-Likelihood:                -35200.
No. Observations:                5618   AIC:                         7.041e+04
Df Residuals:                    5613   BIC:                         7.044e+04
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -26.0066    369.727     -0.070      0.944    -750.815     698.802
country       -0.0247      0.033     -0.757      0.449      -0.089       0.039
year           0.0259      0.184      0.140      0.888      -0.336       0.387
co2            1.0350      0.002    426.629      0.000       1.030       1.040
land_use_change_co2     0.8290      0.011     74.610      0.000       0.807       0.851
==============================================================================
Omnibus:                    12063.809   Durbin-Watson:                   1.980
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        131642995.173
Skew:                         -18.493   Prob(JB):                         0.00
Kurtosis:                     752.005   Cond. No.                     4.39e+05
==============================================================================
```

Figure 27: Linear Regression of Greenhouse gas With 4 Columns

In RO3.3, Total ice-covered area was predicted using multi-linear regression model and found that the model is not that much accurate(Figure 28).

```
------------------------------------------------------------
prediction of iceArea with multi linear regression
------------------------------------------------------------
                            OLS Regression Results
==============================================================================
Dep. Variable:                 iceArea   R-squared:                       0.342
Model:                            OLS   Adj. R-squared:                  0.342
Method:                 Least Squares   F-statistic:                     729.5
Date:                Sun, 07 May 2023   Prob (F-statistic):               0.00
Time:                        00:43:58   Log-Likelihood:                -4702.9
No. Observations:                5618   AIC:                             9416.
Df Residuals:                    5613   BIC:                             9449.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        108.4442      1.623     66.814      0.000     105.262     111.626
country     2.712e-05      0.000      0.189      0.850      -0.000       0.000
year          -0.0437      0.001    -53.956      0.000      -0.045      -0.042
co2         -4.95e-07   1.06e-05     -0.046      0.963   -2.14e-05    2.04e-05
land_use_change_co2  1.731e-05   4.88e-05      0.355      0.723   -7.83e-05       0.000
==============================================================================
Omnibus:                      139.700   Durbin-Watson:                   0.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              148.722
Skew:                           0.392   Prob(JB):                     5.07e-33
Kurtosis:                       2.861   Cond. No.                     4.39e+05
==============================================================================
```

Figure 28: Linear Regression of Ice-Covered Area With 4 Columns

using support vector regressor the same predictions was made and found that the model accuracy is not good for both cases(figure 29-30).

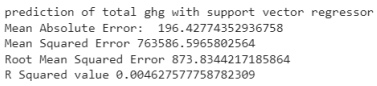


Figure 29: Support Vector Regreesor for GHG

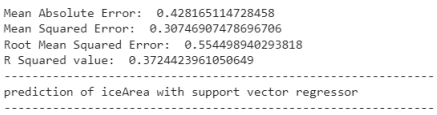


Figure 30: Support Vector Regreesor for Ice-Covered Area

With random forest algorithm, the prediction of greenhouse gas was around 97% while ice covered area model was around 95 percent which is RO3.5 (figure 31-32).

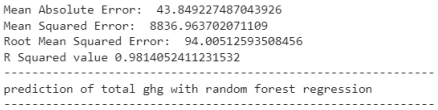


Figure 31: Random Forest Regreesor for GHG

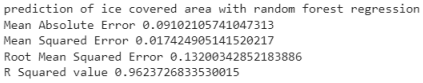


Figure 32: Random Forest Regreesor for Ice-Covered Area

After applying the Adaptive boosting algorithm, the prediction models for the ice-covered area and greenhouse gas are over 80 percent accurate but for greenhouse gas model running it a few time gives fluctuating results cause of different weight distributions at different levels (figure 33-34).

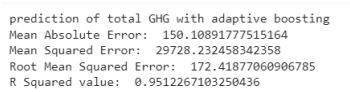


Figure 33: Adaptive Boosting Regreesor for GHG

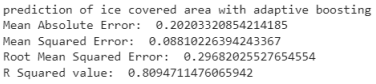


Figure 34: Adaptive Boosting Regreesor for Ice-Covered Area

Using MLP algorithm the greenhouse gas model was 98 percent accurate while ice-covered area model was very bad (figure 35-36).

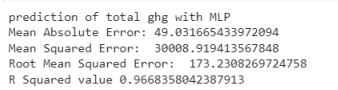


Figure 35: MLP Regreesor for GHG

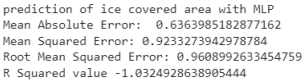


Figure 36: MLP Regreesor for Ice-Covered Area

Finally, it is seen that making predictions of greenhouse gas model with MLP gives the highest accuracy(98%) among all the algorithms while predicting ice-covered area random forest algorithm works the best(95%).

4.4 LEVEL 4: Deep Learning

In this level Deep learning was used to predict the same results as level 3 and compared if deep learning techniques improve the accuracy of the model. According to RO4.1, Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) algorithm is implemented to predict total greenhouse emissions and ice-covered areas. They are part of the Recurrent Neural Network (RNN) Techniques. Following RO4.2, 1D CNN was applied and compared with other results. In every case, some parameter(units, epochs, filters, etc.) was tuned multiple times to find a good combination for the model.

Summary of LSTM Algorithm

Summary of greenhouse gas emissions applying LSTM algorithm is given below:

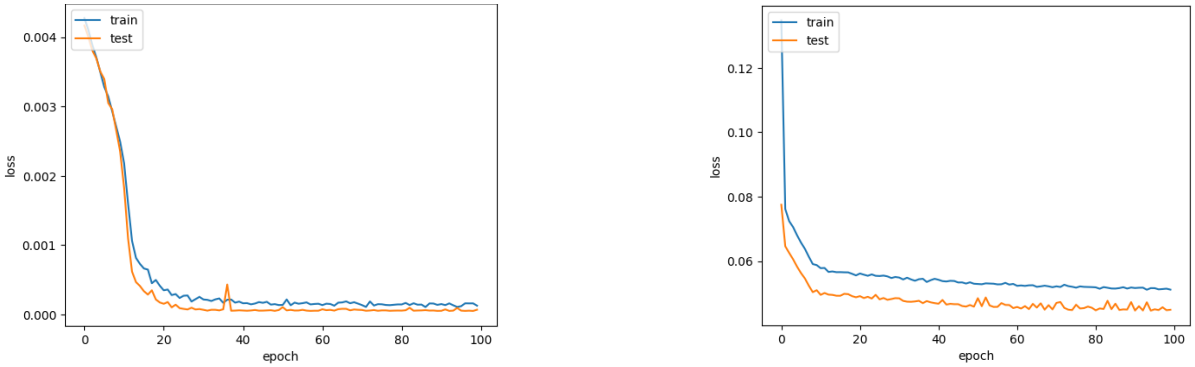
Layers	Units	Epoches	Batch Size	Training RMSE	Testing RMSE
1	50	25	4	420.59	222.27
1	50	50	4	261.31	182.04
1	100	25	4	161.12	182.70
1	100	50	4	234.59	136.19
1	100	100	4	193.59	131.50

Table 2: LSTM for GHG emission

Summary of ice-covered area applying LSTM algorithm is giving below:

Layers	Units	Epoches	Batch Size	Training RMSE	Testing RMSE
1	50	25	4	0.56	0.57
1	100	25	4	0.57	0.56
1	100	100	4	0.57	0.54

Table 3: LSTM for Ice-Covered Area



(a) GHG Emission

(b) Ice-Covered Area

Figure 37: LSTM Model Loss For 100 Unit and 100 Epochs

Summary of GRU Algorithm

Summary of greenhouse gas emissions applying GRU algorithm is given below:

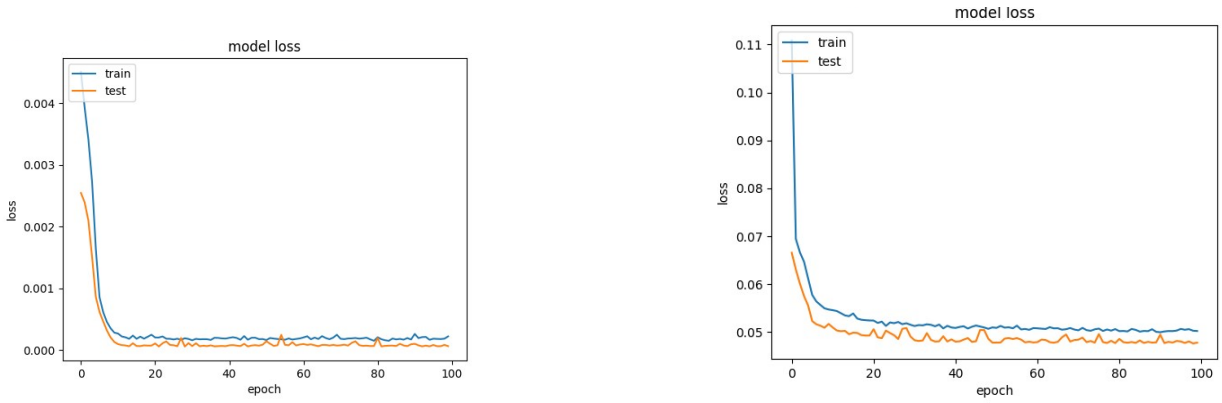
Layers	Units	Epoches	Batch Size	Training RMSE	Testing RMSE
1	50	25	4	104.14	138.95
1	100	25	4	181.29	110.84
1	100	50	4	226.16	125.03
1	100	100	4	120.41	129.22

Table 4: GRU for GHG emission

Summary of ice-covered area applying GRU algorithm is given below:

Layers	Units	Epoches	Batch Size	Training RMSE	Testing RMSE
1	50	25	4	0.52	0.56
1	100	50	4	0.54	0.54
1	100	100	4	0.55	0.54

Table 5: GRU for Ice-Covered Area



(a) GHG Emission

(b) Ice-Covered Area

Figure 38: GRU Model Loss For 100 Unit and 100 Epochs

Summary of 1D CNN Algorithm

Summary of greenhouse gas emissions applying 1D CNN algorithm is given below

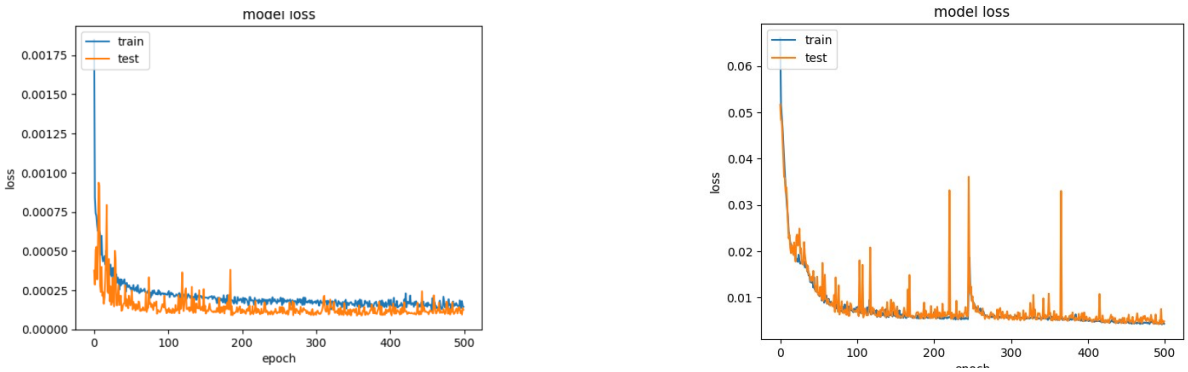
Layers	Units	Epoches	Filter	Training RMSE	Testing RMSE
1	100	25	50	195.11	197.86
1	100	50	75	201.88	154.12
1	100	200	75	175.91	161.25
1	100	500	75	137.93	147.68

Table 6: 1D-CNN for GHG emission

Summary of ice-covered area applying 1D CNN algorithm is given below:

Layers	Units	Epoches	Filter	Training RMSE	Testing RMSE
1	100	50	75	0.25	0.25
1	100	200	75	0.14	0.12
1	100	500	75	0.18	0.16

Table 7: 1D-CNN for Ice-Covered Area



(a) GHG Emission

(b) Ice-Covered Area

Figure 39: CNN Model Loss For 75 Filters and 500 Epochs

As seen from the tables, as the epoch are increased the difference between error gets decreased. That means the model gets well-trained. For predicting ice-covered area all the algorithms used gives quite similar results. On the other hand, for greenhouse gas emission predicting GRU algorithm with 100 epochs provides better results than LSTM trained for 100 epochs or CNN trained for 200 epochs. But if CNN is trained with 500 epochs it reduces the difference of error similar to GRU.

5 Recommendation

Researchers have done extensive studies on the greenhouse gas effect and polar ice-melting separately because of differences in research area but not enough literature has been found that links both of them together and analyze. A lot of data can be found on the internet about GHG gas emissions and polar ice. It is high time, these two topics should be looked at together. By doing so, SDG goal 13(Climate Change) and SDG goal 14(Life Below Water) can both be achieved. Polar ice melting is releasing paleoatmospheric carbon dioxide back into the atmosphere and also reducing the carbon absorption capabilities of the sea to make the situation even worse. Life on land will also be affected by this rising sea level (SDG 15). So the importance of this kind of research which takes a deep dive into these matters is immense.

When using machine learning algorithms to predict, it is very important to make the dataset as clean as possible. Some parts of the world like Antarctica has no GHG gas emission. This type of null value can add extra noise to the model. So the recommendation would be to look at and analyze the dataset first before approaching algorithms. Moreover, when using two or more separate datasets, it is crucial that both dataset units are common. Otherwise, it will be really difficult to extract any meaningfull results from the experiments.

6 Conclusion

In conclusion, the study deals with GHG gas emissions from the past 3 decades all over the world and compares the polar ice-covered area to find the relation between them. The research is done in 4 levels. In the first level, descriptive statistical analysis is used to get better view about the trend of the datasets. Comparing the means of current emissions with standard values and limitations gives a clear idea about the present scenarios. The world seeing more GHG gas than the advised limit on a per-country basis and at the current rate the limit of GHG gas emission target will be surpassed within the next 10 to 15 years. On level 2, the correlations between different contributors of greenhouse gas are analyzed on a continental scale. A good amount of t-tests have been done to identify which component is more responsible for GHG effect within a continent. Generally, production-based carbon dioxide emission causes the most greenhouse gas on every continent. On level 3, five regression algorithms were used to predict greenhouse gas emissions and ice covered are. After the analysis it is found that MLP algorithm works best in finding total amount of greenhouse gas in a year and random forest regression is most suitable for finding ice-covered area in a year. Level 4 is using deep learning techniques in the datasets. Several reading are taken using different combinations as parameter tuning and the learning rate value was used 0.0001. It was found that GRU works better than LSTM if the same combination is used and CNN accuracy depends on a higher number of epochs.

Finding a good combination for all the deep learning techniques is part of future work. Applying other deep learning algorithms like Radial Basis Function Networks (RBFNs) and comparing them with the current results can be a good way to move forward. Trail and Error approach is the only way to find good combinations for parameter tuning. So, some other combinations can be used to get the reading in order to find the best combinations. Finding the time frame where greenhouse gas emits the most to give a conclusion if the emission

is seasonal or overall same throughout the year will be beneficial to take proper decisions about reducing the emission. For this, daily emission data per country is required which will be a huge dataset and will probably require a considerable time to analyze.

References

Azevedo, Ana and Manuel Filipe Santos (2008). "KDD, SEMMA and CRISP-DM: a parallel overview". In: *IADS-DM*.

Baqi, Abdul, Ali Abbas, and Imran Imran (Apr. 2021). "Temporal Variations in Ice Cap of Antarctica and Greenland". In: *International Journal of Innovations in Science and Technology* 3, pp. 52–58. DOI: 10.33411/IJIST/2021030201.

Casper, Julie Kerr (2010). *Greenhouse gases: worldwide impacts*. Infobase Publishing.

Chen, JL et al. (2013). "Rapid ice melting drives Earth's pole to the east". In: *Geophysical Research Letters* 40.11, pp. 2625–2630.

Comyns, Breeda and Frank Figge (2015). "Greenhouse gas reporting quality in the oil and gas industry: A longitudinal study using the typology of "search","experience" and "credence" information". In: *Accounting, Auditing & Accountability Journal* 28.3, pp. 403–433.

Encyclopaedia Britannica, The Editors of (2021). "Environmental Law". In: *Encyclopaedia Britannica*. Accessed May 3, 2023. URL: https://www.britannica.com/topic/environmental-law/Sustainable-development#ref750231.

Fayyad, Usama, Gregory Piatetsky-Shapiro, and Padhraic Smyth (1996). "From data mining to knowledge discovery in databases". In: *AI magazine* 17.3, pp. 37–37.

Fetterer, F. et al. (2017). *Sea Ice Index, Version 3*. DOI: 10.7265/N5K072F8. URL: https://nsidc.org/data/G02135/versions/3.

Garbe, Julius et al. (2020). "The hysteresis of the Antarctic ice sheet". In: *Nature* 585.7826, pp. 538–544.

Hamilton, Ian et al. (2021). "The public health implications of the Paris Agreement: a modelling study". In: *The Lancet Planetary Health* 5.2, e74–e83. ISSN: 2542-5196. DOI: https://doi.org/10.1016/S2542-5196(20)30249-7. URL: https://www.sciencedirect.com/science/article/pii/S2542519620302497.

Hu, Zhiqiang et al. (2022). "CONSTRUCTION OF A MODEL FOR PREDICTING GREENHOUSE GAS EMISSION FOR AQUACULTURE WETLANDS BASED ON ROUGH SET." In: *Environmental Engineering & Management Journal (EEMJ)* 21.2.

Jorgenson, Andrew K (2007). "Does foreign investment harm the air we breathe and the water we drink? A cross-national study of carbon dioxide emissions and organic water pollution in less-developed countries, 1975 to 2000". In: *Organization & Environment* 20.2, pp. 137–156.

Keating, CF (2007). "A simple experiment to demonstrate the effects of greenhouse gases". In: *The Physics Teacher* 45.6, pp. 376–378.

Lee, Ki-Hoon, Junsung Noh, and Jong Seong Khim (2020). "The Blue Economy and the United Nations' sustainable development goals: Challenges and opportunities". In: *Environment International* 137, p. 105528. ISSN: 0160-4120. DOI: https://doi.org/10.1016/j.envint.2020.105528. URL: https://www.sciencedirect.com/science/article/pii/S0160412019338255.

Lösing, Mareen and J Ebbing (2021). "Predicting geothermal heat flow in Antarctica with a machine learning approach". In: *Journal of Geophysical Research: Solid Earth* 126.6, e2020JB021499.

"Machine learning for predicting greenhouse gas emissions from agricultural soils" (2020). In: *Science of The Total Environment* 741, p. 140338. ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2020.140338. URL: https://www.sciencedirect.com/science/article/pii/S0048969720338602.

Manabe, Syukuro (2019). "Role of greenhouse gas in climate change**". In: *Tellus A: Dynamic Meteorology and Oceanography* 71.1, p. 1620078. DOI: 10.1080/16000870.2019.1620078. eprint: https://doi.org/10.1080/16000870.2019.1620078. URL: https://doi.org/10.1080/16000870.2019.1620078.

Oppenheimer, Michael (1998). "Global warming and the stability of the West Antarctic Ice Sheet". In: *Nature* 393.6683, pp. 325–332.

Prakash, Sadguru (Jan. 2021). "IMPACT OF CLIMATE CHANGE ON AQUATIC ECOSYSTEM AND ITS BIODIVERSITY: AN OVERVIEW". In: *International Journal Biological Innovations* 03. DOI: 10.46505/IJBI.2021.3210.

Ritchie, Hannah, Max Roser, and Pablo Rosado (2020). "CO and Greenhouse Gas Emissions". In: *Our World in Data*. https://ourworldindata.org/co2-and-greenhouse-gas-emissions.

Simonsen, Sebastian B et al. (2021). "Greenland Ice Sheet mass balance (1992–2020) from calibrated radar altimetry". In: *Geophysical Research Letters* 48.3, e2020GL091216.

Slater, Thomas et al. (2021). "Earth's ice imbalance". In: *The Cryosphere* 15.1, pp. 233–246.

Stocker, Thomas (2014). *Climate change 2013: the physical science basis: Working Group I contribution to the Fifth assessment report of the Intergovernmental Panel on Climate Change*. Cambridge university press.

Thapa, Samit et al. (2020). “Snowmelt-driven streamflow prediction using machine learning techniques (LSTM, NARX, GPR, and SVR)”. In: *Water* 12.6, p. 1734.
Titus, James G et al. (1991a). “Greenhouse effect and sea level rise: the cost of holding back the sea”. In: *Coastal Management* 19.2, pp. 171–204.
— (1991b). “Greenhouse effect and sea level rise: The cost of holding back the sea”. In: *Coastal Management* 19.2, pp. 171–204. DOI: 10.1080/08920759109362138. eprint: https://doi.org/10.1080/08920759109362138. URL: https://doi.org/10.1080/08920759109362138.
Tuckett, Richard (2019). “Greenhouse gases”. In: *Encyclopedia of Analytical Science*. Elsevier, pp. 362–372.
United Nations Sustainable Development Goals (n.d.[a]). *Goal 13: Climate Action*. https://sdgs.un.org/goals/goal13. Accessed May 3, 2023.
— (n.d.[b]). *Goal 15: Life on Land*. https://sdgs.un.org/goals/goal15. Accessed May 3, 2023.
Wadham, Jemma L et al. (2019). “Ice sheets matter for the global carbon cycle”. In: *Nature communications* 10.1, p. 3567.
Ward, Susan E et al. (2013). “Warming effects on greenhouse gas fluxes in peatlands are modulated by vegetation composition”. In: *Ecology letters* 16.10, pp. 1285–1293.
Wattiaux, Michel et al. (Mar. 2019). “INVITED REVIEW: Emission and mitigation of greenhouse gases from dairy farms: The cow, the manure, and the field”. In: DOI: 10.15232/aas.2018-01803.
Wayne, GP (2014). “Representative concentration pathways”. In: *Skeptical science* 24.
Wright, Peggy (1998). “Knowledge discovery in databases: tools and techniques”. In: *XRDS: Crossroads, The ACM Magazine for Students* 5.2, pp. 23–26.
Zack (April 2021). *How to Determine Equal or Unequal Variance in t-tests*. https://www.statology.org/determine-equal-or-unequal-variance/.
Zhang, Chen et al. (2019). “Biochar for environmental management: Mitigating greenhouse gas emissions, contaminant treatment, and potential negative impacts”. In: *Chemical Engineering Journal* 373, pp. 902–922. ISSN: 1385-8947. DOI: https://doi.org/10.1016/j.cej.2019.05.139. URL: https://www.sciencedirect.com/science/article/pii/S1385894719311635.
Zhang, Ning et al. (2021). “Machine learning in screening high performance electrocatalysts for CO2 reduction”. In: *Small Methods* 5.11, p. 2100987.

A Appendices

A.1 Github Links

All the Codes along with screenshots will be found in the GitHub link

- **Github Link: https://github.com/NMLami/DAV-ACADEMIC-REPORT-LBU**
- **Dataset Github Link: Dataset GitHub Link**
- **Screenshot Github Link: Screenshot GitHub Link**
- **Data Analysis Code Github Link: Data Analysis GitHub Link**

A.2 Code For Level 1 Including Preprocessing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
# load GHG Emission data
ghgEmission = "drive/My Drive/Study/DAV/co2_data.csv"
northHemIce = "drive/My Drive/Study/DAV/north_hemisphere_ice.xlsx"
southHemIce = "drive/My Drive/Study/DAV/south_hemishpere_ice.xlsx"
df_ghg = pd.read_csv(ghgEmission)
df_northHemIce = pd.read_excel(northHemIce)
df_southHemIce = pd.read_excel(southHemIce)
# display(df_ghg)

# Set max_rows and max_columns options
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
```

```
# Select a range of years (2010 to 2015)
start_year = 1990
end_year = 2021
selected_rows = df_ghg.loc[(df_ghg['year'] >= start_year) & (df_ghg['year'] <= end_year)]
```

```
#value to be removed as country
values_to_remove = ['Africa (GCP)','Aland Islands','American Samoa','Asia (GCP)','Asia (excl. China and India)', 'Central America (GCP)','Europe (GCP)', 'European Union (27) (GCP)', 'Falkland Islands', 'Gu
'Middle East (GCP)', 'Netherlands Antilles', 'Non-OECD (GCP)', 'OECD (GCP)', 'Oceania (GCP)','Puerto Rico','Saint Martin (French part)', 'South America (GCP)','Svalbard and Jan Mayen','United States Virgin
 'Europe (excl. EU-28)',
 'European Union (27)',
 'European Union (28)',
 'North America',
 'North America (GCP)',
 'North America (excl. USA)',
 'South America',
 'Upper-middle-income countries',
 'International transport',
 'High-income countries',
 'Asia',
 'Low-income countries',
 'Lower-middle-income countries',
 'Africa',
 'Europe',
 'Europe (excl. EU-27)',]

# Filter the DataFrame to remove rows that match the values in the array
df_filteredCountry = selected_rows[~selected_rows['country'].isin(values_to_remove)]
```

```
# Define columns to keep
columns_to_keep = ['country', 'year','population','cement_co2','cement_co2_per_capita','co2','co2_per_capita','coal_co2','coal_co2_per_capita','flaring_co2','flaring_co2_per_capita','gas_co2','gas_co2_per_

# Keep only specified columns in the DataFrame
df_final = df_filteredCountry[columns_to_keep]
```

```
missing_percentages = df_final.isnull().sum() / len(df_final) * 100
display(missing_percentages)
```

```
country                          0.000000
year                             0.000000
population                       0.269848
cement_co2                       2.698480
cement_co2_per_capita            2.968328
co2                              0.255646
co2_per_capita                   0.525494
coal_co2                         0.298253
coal_co2_per_capita              0.568101
flaring_co2                      0.298253
flaring_co2_per_capita           0.568101
gas_co2                          0.298253
gas_co2_per_capita               0.568101
land_use_change_co2              8.649340
land_use_change_co2_per_capita   8.649340
methane                         17.341287
methane_per_capita              17.341287
nitrous_oxide                   17.341287
nitrous_oxide_per_capita        17.341287
oil_co2                          0.298253
oil_co2_per_capita               0.568101
total_ghg                       17.355489
dtype: float64
```

```
df_greeenHouseGas = df_final.copy()

# #Total use of GHG in a year
# df_greeenHouseGas['total_ghg'] = df_greeenHouseGas['co2'] + df_greeenHouseGas['methane'] + df_greeenHouseGas['nitrous_oxide']

column_name = 'country'
num_unique_country = df_greeenHouseGas[column_name].unique()
num_unique_values_count = df_greeenHouseGas[column_name].nunique()

unique_values_country_list = num_unique_country.tolist()
```

```
#Mean according to country for imputation of data
mean_accor_country = {}

for country in unique_values_country_list:
  countryName = country
  selecte_country_rows = df_greeenHouseGas.loc[(df_greeenHouseGas['country'] == countryName)]
  # display(selecte_country_rows)
  mean_accor_year = {}
  for column_name in selecte_country_rows.columns:
    if selecte_country_rows[column_name].dtype != object:
        mean = selecte_country_rows[column_name].mean()
        mean_accor_year[column_name] = mean

  mean_accor_country[country] = mean_accor_year
```

```
# Loop through the rows and columns and replace null values with Average value per country
for i, row in df_greeenHouseGas.iterrows():
    country = row['country']
    for column_name in df_greeenHouseGas.columns:
        if pd.isnull(row[column_name]):
            replacement_value = mean_accor_country[country][column_name];
            # print(replacement_value)
            if np.isnan(replacement_value) :
              df_greeenHouseGas.at[i, column_name] = 0.0
            else:
              df_greeenHouseGas.at[i, column_name] = mean_accor_country[country][column_name]

missing_percentages = df_greeenHouseGas.isna().sum()
# display(missing_percentages)
```

```
#After Imputation
missing_percentages = df_greeenHouseGas.isnull().sum() / len(df_final) * 100
display(missing_percentages)
```

```
country                          0.0
year                             0.0
population                       0.0
cement_co2                       0.0
cement_co2_per_capita            0.0
co2                              0.0
co2_per_capita                   0.0
coal_co2                         0.0
coal_co2_per_capita              0.0
flaring_co2                      0.0
flaring_co2_per_capita           0.0
gas_co2                          0.0
gas_co2_per_capita               0.0
land_use_change_co2              0.0
land_use_change_co2_per_capita   0.0
methane                          0.0
methane_per_capita               0.0
nitrous_oxide                    0.0
nitrous_oxide_per_capita         0.0
oil_co2                          0.0
oil_co2_per_capita               0.0
total_ghg                        0.0
```

```
#Mean according to country in the last 3 decade

mean_accor_country_Array = []

for country in unique_values_country_list:
  countryName = country
  selecte_country_rows = df_greeenHouseGas.loc[(df_greeenHouseGas['country'] == countryName)]
  # display(selecte_country_rows)
  mean_accor_year_Obj = {}
  for column_name in selecte_country_rows.columns:
    if selecte_country_rows[column_name].dtype != object:
        mean = selecte_country_rows[column_name].mean()
        mean_accor_year_Obj[column_name] = mean
    mean_accor_year_Obj['country'] = countryName

  mean_accor_country_Array.append(mean_accor_year_Obj)
```

```
#co2_graph for top 10 country

# sorted_list_by_co2 according to mean of that country
sorted_list_by_co2 = sorted(mean_accor_country_Array, key=lambda x: x['co2'], reverse=True)[:10]

#count country
country_count = len(mean_accor_country_Array)
print("Total Number of countries in the dataset: ",country_count)

#The IPCC report suggests that to have at least a 50 per cent chance of keeping to less than 2°C of warming, mankind must emit no more than 1212 giga ton
# from 2012 to rest of the century

#convert 1212 giga ton to million ton to match the data set and find average for every country
co2_limit_per_country = 1212000/(country_count*(2100-2012))
print("Average emission of CO2 per country should be: "+ str(round(co2_limit_per_country, 2))+" Million ton")

x_axis_countries_name = []
y_axis_mean_value = []
y_axis_average_co2_value = []

for meanValues in sorted_list_by_co2:
    x_axis_countries_name.append(meanValues['country'])
    y_axis_average_co2_value.append(co2_limit_per_country)
    y_axis_mean_value.append(meanValues["co2"])

bar_width=.3
y_pos = np.arange(len(x_axis_countries_name))


plt.bar(y_pos,y_axis_mean_value, align='center', alpha=0.5, width =bar_width ,label = 'co2 emission')
plt.bar(y_pos+bar_width,y_axis_average_co2_value, align='center', alpha=0.5, width =bar_width,label='Avg co2 emission limit')

plt.legend(loc='upper right', frameon=False)

plt.xticks(fontsize=7)
plt.xticks(y_pos,x_axis_countries_name, rotation='vertical')
plt.ylabel('co2(in million tonnes)')
plt.title("co2 emission per year")
```

```
# Graph total use of Co2 in a year
#Dataset are cleaned before and now only taking the variable

sorted_list_by_co2_per_capita = sorted(mean_accor_country_Array, key=lambda x: x['co2_per_capita'], reverse=True)[:10]
# print(sorted_list_by_co2_per_capita)

x_axis_countries_name = []
y_axis_mean_value = []

for meanValues in sorted_list_by_co2_per_capita:
    x_axis_countries_name.append(meanValues['country'])
    y_axis_mean_value.append(meanValues["co2_per_capita"])


y_pos = np.arange(len(x_axis_countries_name))

plt.bar(y_pos,y_axis_mean_value, align='center', alpha=0.5, width =.3)
plt.xticks(fontsize=7)
plt.xticks(y_pos,x_axis_countries_name, rotation='vertical')
plt.ylabel('co2(in million tonnes)')
plt.title("co2 emission per capita in a year")
```

```
# Graph total use of Co2 in a year

sorted_list_by_total_ghg = sorted(mean_accor_country_Array, key=lambda x: x['total_ghg'], reverse=True)[:10]
# print(sorted_list_by_total_ghg)

x_axis_countries_name = []
y_axis_mean_value = []

for meanValues in sorted_list_by_total_ghg:
    x_axis_countries_name.append(meanValues['country'])
    y_axis_mean_value.append(meanValues["total_ghg"])


y_pos = np.arange(len(x_axis_countries_name))

plt.bar(y_pos,y_axis_mean_value, align='center', alpha=0.5, width =.3)
plt.xticks(fontsize=7)
plt.xticks(y_pos,x_axis_countries_name, rotation='vertical')
plt.ylabel('total GHG (in million tonnes)')
plt.title("total GHG gas per year ")
```

```
#take china to show the descriptive analysis
df_china = df_greeenHouseGas[df_greeenHouseGas['country'] == 'China']

descriptiveAnalysisofChina = df_china.describe()

display(descriptiveAnalysisofChina)

descriptiveAnalysisofChina.to_csv('chinaDescriptiveAnalysis.csv', index=False)
```

	year	population	cement_co2	cement_co2_per_capita	co2	co2_per_capita	coal_co2	coal_co2_per_capita	flaring_co2	flaring_co2_per_capita	gas_co2	gas_co2_per_capita	land_use_change_co2	land_use_chan
count	32.000000	3.200000e+01	32.000000	32.000000	32.000000	32.000000	32.000000	32.00000	32.000000	32.000000	32.000000	32.000000	32.000000	
mean	2005.500000	1.307463e+09	463.018156	0.343250	6513.532969	4.858562	4808.551812	3.59100	1.183625	0.000875	209.944187	0.152500	560.976750	
std	9.380832	8.227590e+07	265.755567	0.182297	3142.035494	2.100468	2250.707277	1.50599	1.820694	0.001362	218.980660	0.152611	386.538195	
min	1990.000000	1.153704e+09	84.513000	0.073000	2484.855000	2.154000	1976.684000	1.71300	0.000000	0.000000	29.301000	0.025000	8.427000	
25%	1997.750000	1.244577e+09	209.245250	0.168500	3508.435250	2.834500	2596.550750	2.06850	0.000000	0.000000	48.993250	0.039750	287.322000	
50%	2005.500000	1.308987e+09	440.868000	0.336500	6182.679500	4.722500	4664.882000	3.56350	0.000000	0.000000	94.355000	0.072000	417.916000	
75%	2013.250000	1.378373e+09	727.257250	0.524750	9801.186750	7.084750	7277.680750	5.17575	3.118500	0.002000	332.650750	0.241500	1011.731500	
max	2021.000000	1.425894e+09	858.233000	0.602000	11472.368000	8.046000	7955.985000	5.58000	5.119000	0.004000	773.866000	0.543000	1322.667000	

```
#take Oceania  to show the descriptive analysis
df_oceania = df_greeenHouseGas[df_greeenHouseGas['country'] == 'Oceania']

descriptiveAnalysisofOcenia = df_oceania.describe()

display(descriptiveAnalysisofOcenia)
descriptiveAnalysisofOcenia.to_csv('oceaniaDescriptiveAnalysis.csv', index=False)
```

	year	population	cement_co2	cement_co2_per_capita	co2	co2_per_capita	coal_co2	coal_co2_per_capita	flaring_co2	flaring_co2_per_capita	gas_co2	gas_co2_per_capita	land_use_change_co2	land_use_change_co
count	32.000000	3.200000e+01	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	
mean	2005.500000	3.479677e+07	3.929344	0.116906	409.812344	11.926031	183.944625	5.421906	10.225812	0.289563	65.269312	1.864031	148.737750	
std	9.380832	5.459170e+06	0.379527	0.022337	53.572216	0.770652	20.870028	0.820367	3.873713	0.065078	16.032811	0.183007	113.270702	
min	1990.000000	2.674421e+07	3.229000	0.073000	309.475000	10.017000	147.035000	3.624000	7.401000	0.233000	40.887000	1.515000	-18.650000	
25%	1997.750000	3.018633e+07	3.658250	0.099750	367.875750	11.439500	165.393500	4.690000	7.729250	0.252250	51.455500	1.719500	64.376500	
50%	2005.500000	3.410462e+07	3.987500	0.128000	436.926500	11.926500	183.953500	5.617500	8.570000	0.263500	62.988500	1.905500	119.739500	
75%	2013.250000	3.922567e+07	4.140000	0.131500	452.223500	12.605000	200.104000	6.099250	10.289000	0.287500	78.958750	2.005000	209.260000	
max	2021.000000	4.449206e+07	4.706000	0.150000	470.359000	12.998000	213.906000	6.391000	21.176000	0.492000	93.151000	2.132000	551.652000	

```
#Ice area in 2 hemispheres

df_totalIceArea = pd.concat([df_northHemIce['year'],df_northHemIce['Annual'].rename("north_hemisphere_ice")  , df_southHemIce['Annual'].rename("south_hemisphere_ice")], axis=1)
df_totalIceArea['total_ice'] = df_northHemIce['Annual']+df_southHemIce['Annual']

# display(df_totalIceArea)
#Statistical Analysis of ice covered area
discriptiveAnalysisOfIceArea = df_totalIceArea[['north_hemisphere_ice','south_hemisphere_ice','total_ice']].describe()

display(discriptiveAnalysisOfIceArea)
# discriptiveAnalysisOfIceArea.to_csv('iceAreaDescriptiveAnalysis.csv', index=False)
```

	north_hemisphere_ice	south_hemisphere_ice	total_ice
count	33.000000	33.000000	33.000000
mean	9.179424	11.657121	20.836545
std	0.475100	0.476877	0.726420
min	8.311000	10.647000	19.433000
25%	8.779000	11.410000	20.250000
50%	9.215000	11.687000	21.192000
75%	9.593000	11.961000	21.317000
max	10.086000	12.776000	21.861000

```
#Graph of Annual data
#Take the vlues from dataset
years = np.array(df_totalIceArea["year"])
iceValues = np.array(df_totalIceArea['total_ice'])

fig, ax = plt.subplots()

ax.plot(years,iceValues )

plt.xlabel('Year')
plt.ylabel('Area(million sq. kilometer)')
plt.title('Ice Covered Area')
plt.show()
```

```python
#Make World Grpah of GHG Emission
import pandas as pd
import plotly.express as px
#Make a list from mean values
country_list = []
ghg_emission = []
for country, info in mean_accor_country.items():
    country_list.append(country)
    ghg_emission.append(info['total_ghg'])

# Create DataFrame
df = pd.DataFrame({'Country': country_list, 'GHG': ghg_emission})
# Create a choropleth map using plotly
fig = px.choropleth(df, locations='Country', locationmode='country names',
                    color='GHG', color_continuous_scale='YlOrRd',
                    title='GHG Emission Map')
# Customize the map layout
fig.update_layout(
    title_font=dict(size=26),
    title_x=0.5,
    geo=dict(
        showframe=False,
        showcoastlines=False,
        projection_type='equirectangular'
    ),
    coloraxis_colorbar=dict(
        title='GHG Emission',
        title_font=dict(size=15),
        len=0.5,
        yanchor='middle',
        y=0.5,
        tickfont=dict(size=12),
        ticks='outside',
        tickformat='.1f'
    )
)
# Display the graph
fig.show()
```

A.3 Code for Level 2 Without Preprocesing

A.3.1 Code for one T-Test is given and Correlation matrix. Because only column name were changed for different experiments

```python
#Countries by continent

asia_countries = ['Afghanistan', 'Bahrain', 'Bangladesh', 'Bhutan', 'Brunei', 'Cambodia', 'China', 'Cyprus', 'Georgia', 'India', 'Indonesia', 'Iran', 'Iraq', 'Israel', 'Japan', 'Jordan', 'Kazakhstan', 'Kuwait', 'Kyrgyzstan', 'Laos',

europe_countries = ['Albania', 'Andorra', 'Austria', 'Belarus', 'Belgium', 'Bosnia and Herzegovina', 'Bulgaria', 'Croatia', 'Cyprus', 'Czech Republic', 'Denmark', 'Estonia', 'Finland', 'France', 'Germany', 'Greece', 'Hungary', 'Icel

african_countries = ['Algeria', 'Angola', 'Benin', 'Botswana', 'Burkina Faso', 'Burundi', 'Cameroon', 'Cape Verde', 'Central African Republic', 'Chad', 'Comoros', 'Democratic Republic of the Congo', 'Republic of the Congo', 'Cote d\

north_america = ['Canada', 'United States', 'Mexico', 'Greenland', 'Cuba', 'Haiti', 'Dominican Republic', 'Jamaica', 'Trinidad and Tobago', 'Costa Rica', 'Panama', 'El Salvador', 'Honduras', 'Guatemala', 'Belize', 'Nicaragua', 'Sain

south_america = ['Argentina', 'Bolivia', 'Brazil', 'Chile', 'Colombia', 'Ecuador', 'Guyana', 'Paraguay', 'Peru', 'Suriname', 'Uruguay', 'Venezuela']

australia = ['Australia', 'Papua New Guinea', 'New Zealand', 'Fiji', 'Solomon Islands', 'Micronesia', 'Vanuatu', 'New Caledonia', 'French Polynesia','Kiribati','Tonga','Marshall Islands','Northern Mariana Islands','American Samoa','
```

```python
# T test between co2 and total_ghg in Asia
from scipy import stats
print("T test between co2 and total ghg gas in Asia")

df_filtered_by_country = df_greeenHouseGas[df_greeenHouseGas['country'].isin(asia_countries)]
column1 = 'co2'
column2 = 'total_ghg'
df_filtered_by_country = df_filtered_by_country[[column1,column2]]

# Drop columns where the value is zero

df_filtered_by_country = df_filtered_by_country[(df_filtered_by_country != 0).all(1)]

if (df_filtered_by_country == 0).any().any():
    print("There is at least one zero value in the dataframe")
else:
    print("There are no zero values in the dataframe")

num_rows = df_filtered_by_country.shape[0]
# print(num_rows)

dataset_column_1 = df_filtered_by_country[column1]
dataset_column_2 = df_filtered_by_country[column2]

ratio= max(np.var(dataset_column_1), np.var(dataset_column_2))/min(np.var(dataset_column_1), np.var(dataset_column_2))

print("The variance ratio is: ", ratio)

alpha = 0.05

#t-test
t_value, p_value = stats.ttest_ind(dataset_column_1, dataset_column_2 )

print("p value: ", p_value)
print("t value: ", t_value)

if p_value < alpha:
  print("Null Hypothesis Is Rejected")
else:
  print("Null Hypothesis Is Accepted")
```

```
# Create a correlation matrix
columns_choose = df_greeenHouseGas[['country','year','co2','land_use_change_co2','coal_co2','oil_co2','gas_co2','methane','nitrous_oxide','total_ghg']]
corr= columns_choose.corr()
display(corr)

#Potting the matrix
mask = np.zeros_like(corr, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True

# matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))

# Generate a custom colormap
cmap = sns.diverging_palette(220, 10, as_cmap=True)

# Draw the heatmap with styling

sns.heatmap(
    corr,          # The data to plot
    mask=mask,
    cmap=cmap,
    annot=True,
    vmax=.3,
    vmin=-.3,
    center=0,
    square=True,
    linewidths=.5,
    cbar_kws={"shrink": .5}
)
```

A.4 Code for Level 3: Machine Learning

A.4.1 Pre-Processing before Machine Learning

```
#Merging of Two Dataset
# create a new column with default value of 0
df_ghg_with_ice['iceArea'] = 0

# df_with_ice.reset_index(drop=True)
valueOfIceArea = df_totalIceArea.loc[df_totalIceArea['year'] == 1994, 'total_ice']
n=list(valueOfIceArea.keys())[0]
print(n)
# display(df_with_ice.loc[(df_with_ice['year'] == 2020) & (df_with_ice['country'] == 'Afghanistan')])
#loop through each row in the DataFrame

for index, row in df_ghg_with_ice.iterrows():
    year = row['year']
    # print(year)
    valueOfIceArea = df_totalIceArea.loc[df_totalIceArea['year'] == year, 'total_ice']
    key = list(valueOfIceArea.keys())[0]
    value = valueOfIceArea[key]


    df_ghg_with_ice.at[index, 'iceArea'] = value
```

```
df_ghg_with_ice=df_ghg_with_ice[['country','year','co2','land_use_change_co2','total_ghg','iceArea','methane','nitrous_oxide']]
```

```
df_ghg_with_ice = df_ghg_with_ice[(df_ghg_with_ice != 0).all(1)]
if (df_ghg_with_ice == 0).any().any():
    print("There is at least one zero value in the dataframe")
else:
    print("There are no zero values in the dataframe")

num_rows = df_ghg_with_ice.shape[0]
print("Number of rows: ",num_rows)
```

```
There are no zero values in the dataframe
Number of rows:  5618
```

```
# LEBEL ENCODING THE CONTRY
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
entity = ['country']
# for var in entity:
#     data[var] = le.fit_transform(data[var])
df_ghg_with_ice['country'] = le.fit_transform(df_ghg_with_ice['country'])
```

```
sorted_list = df_ghg_with_ice.sort_values('year')
```

A.4.2 Multi-linear Regression

```
#predict total_ghg with multi-linear regression
import statsmodels.api as sm

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

#'country','year','co2','land_use_change_co2','methane','nitrous_oxide'
x= sorted_list[['country','year','co2']];
y = sorted_list['total_ghg']

# print(y)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)


lr = LinearRegression()
lr.fit(x_train, y_train)

y_pred = lr.predict(x_test)

print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test,y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,y_pred))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(y_test,y_pred)))

print('R squared value: ', metrics.r2_score(y_test,y_pred))

#Use statsmodel for model summary
x = sm.add_constant(x)
model = sm.OLS(y, x).fit()
predictions = model.predict(x)
print_model = model.summary()
print("----------------------------------------------------------------")
print("prediction of Total Greenhouse gas with multi linear regression ")
print("----------------------------------------------------------------")

print(print_model)
```

A.4.3 SVM Regression

```
#predict iceArea with support vector regressor
import statsmodels.api as sm

from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn import metrics


x= sorted_list[['country','year','co2','land_use_change_co2']];
y = sorted_list['iceArea']

# print(y)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)


lr = make_pipeline(StandardScaler(),SVR(C=1.0,epsilon=.1))
lr.fit(x_train, y_train)

y_pred = lr.predict(x_test)

print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test,y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,y_pred))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(y_test,y_pred)))

print('R Squared value: ', metrics.r2_score(y_test,y_pred))

#Use statsmodel for model summary
x = sm.add_constant(x)
model = sm.OLS(y, x).fit()
predictions = model.predict(x)
print_model = model.summary()
print("----------------------------------------------------------------")
print("prediction of iceArea with support vector regressor")
print("----------------------------------------------------------------")
print(print_model)
```

A.4.4 Random Forest Regression

```
#predict total_ghg with random forest regression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

x= sorted_list[['country','year','co2','land_use_change_co2']];
y = sorted_list['total_ghg']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)
rf_model = RandomForestRegressor(n_estimators=100, max_depth=5)
# Train the model on the training data
rf_model.fit(x_train, y_train)

# Use the model to predict the output values for the test set
y_pred = rf_model.predict(x_test)
# Calculate the mean squared error and R^2 score for the predictions
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test,y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,y_pred))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(y_test,y_pred)))

print('R Squared value', metrics.r2_score(y_test,y_pred))

# Use statsmodel for model summary
x = sm.add_constant(x)
model = sm.OLS(y, x).fit()
predictions = model.predict(x)
print_model = model.summary()
print("----------------------------------------------------------------")
print("prediction of total ghg with random forest regression")
print("----------------------------------------------------------------")
print(print_model)
```

A.4.5 Adaptive Boosting Regression

```
#predict total_ghg with adaptive boosting
from sklearn.ensemble import AdaBoostRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score


x= sorted_list[['country','year','co2','land_use_change_co2']];
y = sorted_list['total_ghg']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)

ada_boost_model = AdaBoostRegressor(n_estimators=100, random_state=42)

# Train the model
ada_boost_model.fit(x_train, y_train)

# Make predictions on the test data
y_pred = ada_boost_model.predict(x_test)


# Calculate the mean squared error and R^2 score for the predictions
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("prediction of total GHG with adaptive boosting")
print("Mean Absolute Error: ", metrics.mean_absolute_error(y_test,y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,y_pred))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(y_test,y_pred)))

print('R Squared value: ', metrics.r2_score(y_test,y_pred))
```

A.4.6 MLP Regression

```
#predict total_ghg with MLP
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score


x= sorted_list[['country','year','co2','land_use_change_co2']];
y = sorted_list['total_ghg']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)

# Create MLP regressor model
mlp_model = MLPRegressor(hidden_layer_sizes=(10,10), max_iter=1000, alpha=0.01, solver='adam', random_state=42)

# Train the model
mlp_model.fit(x_train, y_train)

y_pred = mlp_model.predict(x_test)

# Calculate the mean squared error and R^2 score for the predictions
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("prediction of total ghg with MLP")
print("Mean Absolute Error:", metrics.mean_absolute_error(y_test,y_pred))
print("Mean Squared Error: ", metrics.mean_squared_error(y_test,y_pred))
print("Root Mean Squared Error: ", np.sqrt(metrics.mean_squared_error(y_test,y_pred)))

print('R Squared value', metrics.r2_score(y_test,y_pred))
```

A.5 Code for Level 4: Deep Learning

A.5.1 LSTM

```
#use lstm on total_ghg prediction model
import pandas as pd
from sklearn.model_selection import train_test_split
import keras
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import MinMaxScaler
from keras.layers import Dense, LSTM, Dropout,Flatten

#'country','year','co2','land_use_change_co2','methane','nitrous_oxide'
x= sorted_list[['country','year','co2','land_use_change_co2']].values
y = sorted_list[['total_ghg']].values

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)

x_scale =MinMaxScaler()
y_scale =MinMaxScaler()

x_train_scaled = x_scale.fit_transform(x_train)
y_train_scaled = y_scale.fit_transform(y_train)
x_test_scaled = x_scale.fit_transform(x_test)

x_train_scaled,y_train_scaled =np.array(x_train_scaled),np.array(y_train_scaled)

# reshape input to be [samples, time steps, features]
trainX = np.reshape(x_train_scaled, (x_train_scaled.shape[0],  x_train_scaled.shape[1], 1))
trainY = np.reshape(y_train_scaled, (y_train_scaled.shape[0], y_train_scaled.shape[1], 1))


# Create the LSTM model
model = Sequential()
model.add(LSTM(units=100, return_sequences=False, input_shape=(trainX.shape[1], 1)))
# model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1))
```

```
optimizer = keras.optimizers.Adam(lr=0.0001)
# Compile the model
model.compile(optimizer=optimizer, loss='mean_squared_error',metrics =['accuracy'])

# Train the model
lstm_history = model.fit(trainX, trainY, epochs=25, batch_size=4, shuffle=True, validation_split=0.20)

testX = np.array(x_test_scaled)
testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))

predictions = model.predict(testX)

x_pred = model.predict(trainX)

predictions= y_scale.inverse_transform(predictions)
x_pred = y_scale.inverse_transform(x_pred)


# calculate root mean squared error
trainScore = np.sqrt(np.mean((( predictions-y_test)**2)))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = np.sqrt(np.mean(((x_pred - y_train)**2)))
print('Test Score: %.2f RMSE' % (testScore))

print("\n lstm_history ", lstm_history.history.keys())
```

```
#plot the accuracy
plt.plot(lstm_history.history['accuracy'])
plt.plot(lstm_history.history['val_accuracy'])
plt.title('Accuracy of the model')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train','test'], loc = 'upper left')
print("Accuracy for 25 epoch with 100 Unit")
plt.show
```

```
#plot the losses
plt.plot(lstm_history.history['loss'])
plt.plot(lstm_history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train','test'], loc = 'upper left')
print("Model loss for 25 epoch with 100 Unit")
plt.show
```

A.5.2 GRU

```
#GRU
import pandas as pd
from sklearn.model_selection import train_test_split
import keras
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import MinMaxScaler
from keras.layers import Dense, LSTM, Dropout,Flatten, GRU

#'country','year','co2','land_use_change_co2','methane','nitrous_oxide'
x= sorted_list[['country','year','co2','land_use_change_co2']].values
y = sorted_list[['total_ghg']].values


x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)

x_scale =MinMaxScaler()
y_scale =MinMaxScaler()

x_train_scaled = x_scale.fit_transform(x_train)
y_train_scaled = y_scale.fit_transform(y_train)
x_test_scaled = x_scale.fit_transform(x_test)

x_train_scaled,y_train_scaled =np.array(x_train_scaled),np.array(y_train_scaled)

# reshape input to be [samples, time steps, features]
trainX = np.reshape(x_train_scaled, (x_train_scaled.shape[0],  x_train_scaled.shape[1], 1))
trainY = np.reshape(y_train_scaled, (y_train_scaled.shape[0], y_train_scaled.shape[1], 1))

gru_model = Sequential()
gru_model.add(GRU(units=100, return_sequences=False, input_shape=(trainX.shape[1], 1)))
gru_model.add(Dropout(0.2))
gru_model.add(Dense(units=1))
```

```
optimizer = keras.optimizers.Adam(lr=0.0001)
# Compile the model
gru_model.compile(optimizer=optimizer, loss='mean_squared_error',metrics =['accuracy'])

# Train the model
gru_history = gru_model.fit(trainX, trainY, epochs=50, batch_size=4, shuffle=True, validation_split=0.20)

testX = np.array(x_test_scaled)
testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))

predictions = gru_model.predict(testX)

x_pred = gru_model.predict(trainX)

predictions= y_scale.inverse_transform(predictions)
x_pred = y_scale.inverse_transform(x_pred)


# calculate root mean squared error
trainScore = np.sqrt(np.mean((( predictions-y_test)**2)))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = np.sqrt(np.mean(((x_pred - y_train)**2)))
print('Test Score: %.2f RMSE' % (testScore))

print("gru_history", gru_history.history.keys())
```

A.5.3 1D CNN

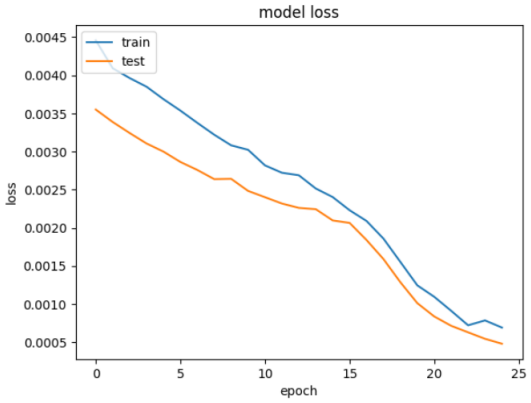
```
#1D CNN
import pandas as pd
from sklearn.model_selection import train_test_split
import keras
from keras.models import Sequential
from keras.layers import Dense
from sklearn.preprocessing import MinMaxScaler
from keras.layers import Dense, LSTM, Dropout,Flatten, GRU
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D

#'country','year','co2','land_use_change_co2','methane','nitrous_oxide'
x= sorted_list[['country','year','co2','land_use_change_co2']].values
y = sorted_list[['total_ghg']].values


x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .2)

x_scale =MinMaxScaler()
y_scale =MinMaxScaler()

x_train_scaled = x_scale.fit_transform(x_train)
y_train_scaled = y_scale.fit_transform(y_train)
x_test_scaled = x_scale.fit_transform(x_test)

x_train_scaled,y_train_scaled = np.array(x_train_scaled),np.array(y_train_scaled)

# reshape input to be [samples, time steps, features]
trainX = np.reshape(x_train_scaled, (x_train_scaled.shape[0],  x_train_scaled.shape[1], 1))
trainY = np.reshape(y_train_scaled, (y_train_scaled.shape[0], y_train_scaled.shape[1], 1))
```

```
# Create the 1D CNN model
cnn_model = Sequential()
cnn_model.add(Conv1D(filters = 75 , kernel_size =2, activation='relu' ,input_shape=(trainX.shape[1], 1)))
# model.add(LSTM(units=50))
cnn_model.add(MaxPooling1D(pool_size =2))
cnn_model.add(Flatten())
cnn_model.add(Dense(100,activation='relu'))
cnn_model.add(Dense(units=1))

# optimizer = keras.optimizers.Adam(lr=0.0001)
# Compile the model
cnn_model.compile(optimizer='adam', loss='mean_squared_error',metrics =['accuracy'])

# Train the model
cnn_model_hisotry = cnn_model.fit(trainX, trainY, epochs=500, batch_size=4, shuffle=True, validation_split=0.20)

testX = np.array(x_test_scaled)
testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))

predictions = cnn_model.predict(testX)

x_pred = cnn_model.predict(trainX)

predictions= y_scale.inverse_transform(predictions)
x_pred = y_scale.inverse_transform(x_pred)


# calculate root mean squared error
trainScore = np.sqrt(np.mean((( predictions-y_test)**2)))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = np.sqrt(np.mean(((x_pred - y_train)**2)))
print('Test Score: %.2f RMSE' % (testScore))

print("cnn_history", cnn_model_hisotry.history.keys())
```

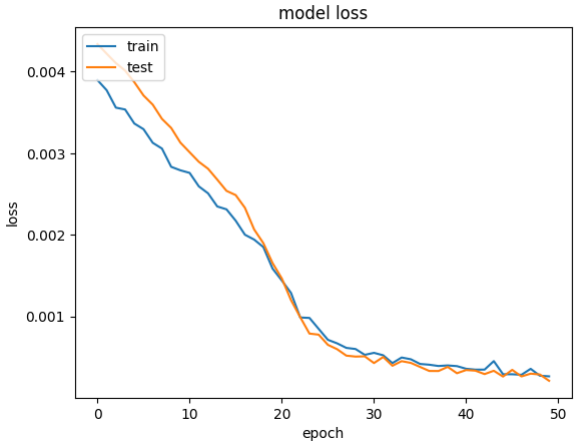
A.6 Loss Function Graphs

A.6.1 LSTM For GHG Emissions

```
Model loss for 25 epoch with 50 Unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

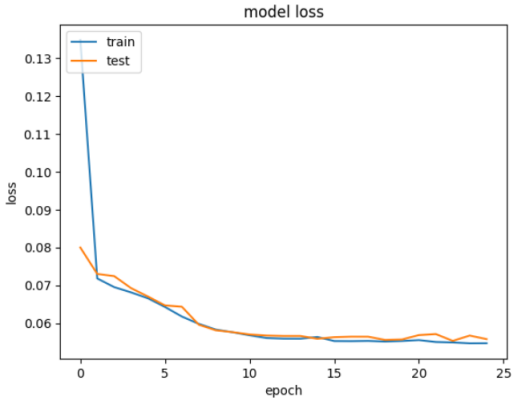



```
Model Loss for 50 epoch 50 unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

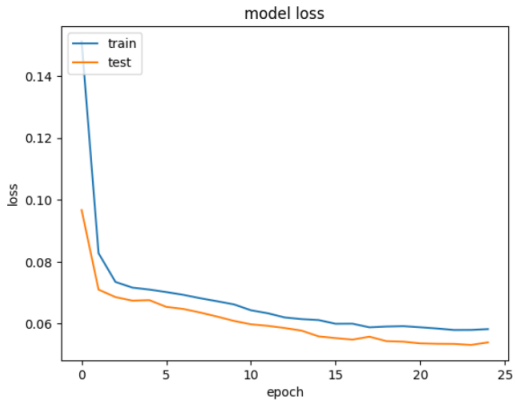


A.6.2 LSTM For Ice Covered Area

```
Ice cover area Model loss for 25 epoch with 100 Unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

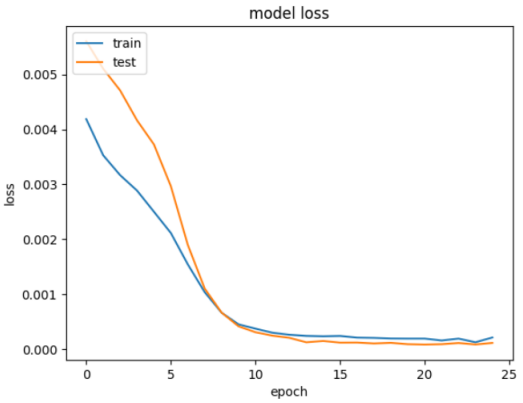


```
Ice cover area Model loss for 25 epoch with 50 Unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

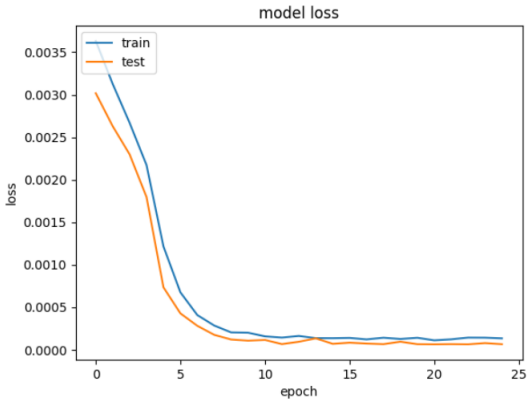


A.6.3 GRU For GHG Emissions

```
Total GHG Model loss for 25 epoch with 50 Unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

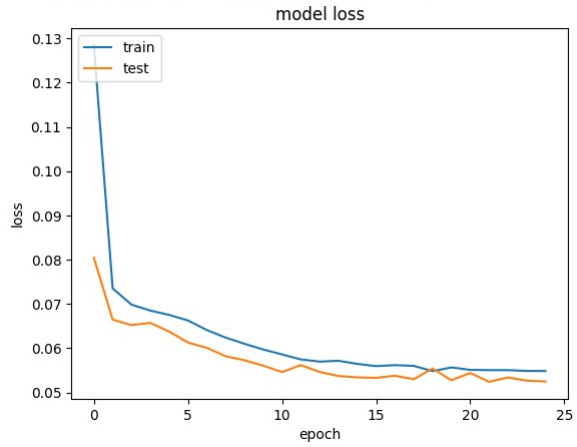



```
Total GHG Model loss for 25 epoch with 100 Unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

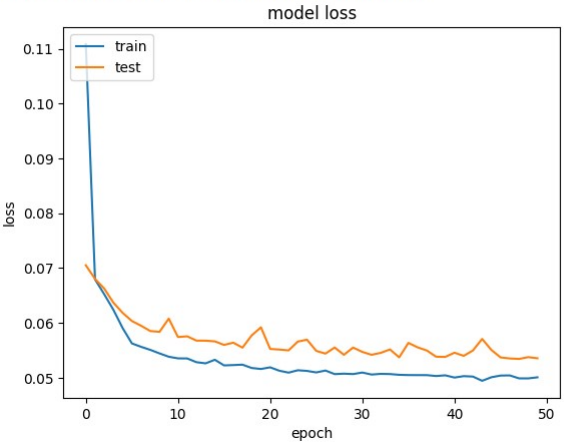


A.6.4 GRU For Ice-Covered Area

```
loss for 25 epoch 50 unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

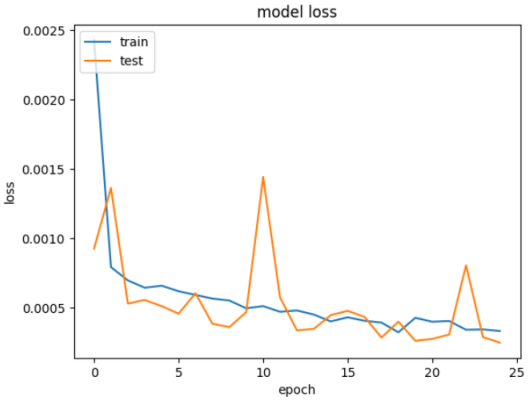



```
loss for 50 epoch 100 unit
<function matplotlib.pyplot.show(close=None, block=None)>
```

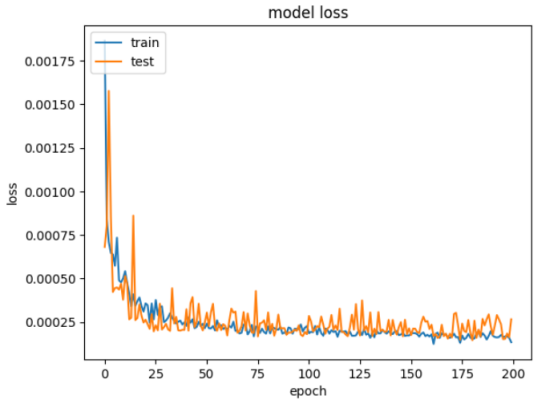


A.6.5 1D-CNN For GHG Emissions

```
Total GHG Model loss for 25 epoch with 50 filter
<function matplotlib.pyplot.show(close=None, block=None)>
```

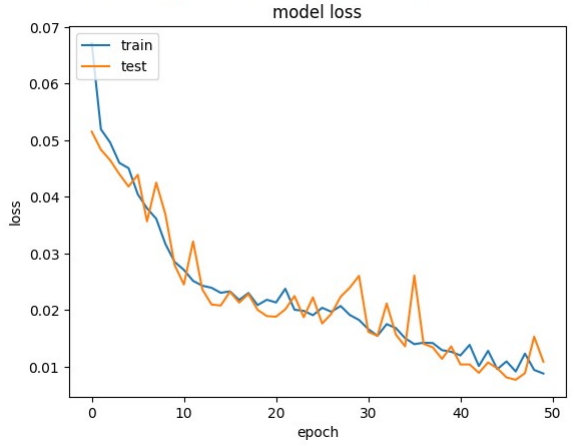


```
Total GHG Model loss for 200 epoch with 75 filter
<function matplotlib.pyplot.show(close=None, block=None)>
```



A.6.6 1D-CNN For Ice-Covered Area

```
losss for 50 epoch 75 filter
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
losss for 200 epoch 75 filter
<function matplotlib.pyplot.show(close=None, block=None)>
```