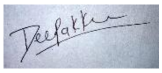
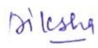
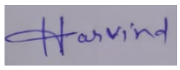
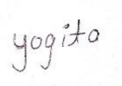
REPORT
ON
ANALYSIS OF
A WAY TO STUDY
CLIMATE
CHARACTERISTICS

<u>AUTHORS</u> :	<u>SIGNATURE :</u>
Deepak Kumar(21025762017)	
Diksha Sharma(21025762018)	
Harvind Kishor Mahto(21036762019)	
Yogita Kumari(21026762068)	

*Report of project work done between September and November 2022 as part of the M.Sc. course PHY-OT543 Complex Systems and Networks, Department of Physics and Astrophysics, University of Delhi

1. ABSTRACT :

In this report we are going to analyse a climate change characteristic i.e., temperature of regions like New Delhi and Mumbai in India & Xi'an in China over a time series using visibility graph theory, a tool in complex network theory. We established a global temperature network of each region for a decade (2011-2020) where nodes will be each day with corresponding daily highest temperature and edges between each node will be calculated by algorithm of visibility graph theory.

Then we divided the global network into local networks of 4 year window by moving each back 2 years or 1 year to generate a new one to analyse the temperature fluctuations. We studied the degree distribution for all local and global temperature networks for every region to find if it conforms to the law of power law or not which further implies if the climate networks of all these regions are scale free networks or not as found in the reference research paper.

Afterwards using the Louvain algorithm (a modular algorithm), community detection for these networks is done to get further insight into the function of networks and see any evolutionary trend in it .Community detection constitute the most important part of it as it helps us to explore how temperature fluctuation between years is related to each other and see how under the background of climate changes (particularly due to global warming),it is affected and can help us to find possibility of extreme events .

Furthermore, we are considering the data for Xi'an in China to see if it is in agreement to result as stated in the reference paper. We are also comparing the results of each region with each other to see if visibility graph is a convenient way to study the climate change characteristics or not indeed.

2. INTRODUCTION:

Natural as well as human activities have led to a shift in temperature and weather pattern due to which extreme climate events have been observed across the globe in past few decades. high temperature and dry spells leading to drought and water unavailability at some places while erratic monsoon, cyclones and floods in others are topics of great concern. In this project we intended to study law of variation of temperature. For a time, series of 10 years, we have considered each day as a node and corresponding graph is constructed via visibility graph theorem various aspects of which are studied using degree distribution and community detection.

2.1. <u>Complex network</u>:

A complex network or graph is collection of points called nodes or vertices connected together via edges or links. A graph is represented as G = (E, V) where V is the set of vertices and E is the set of links between the nodes. To get a more intuitive idea of the system we can associate each node to a strength or weight like here we have associated each node i.e., a day to its maximum temperature, such a graph is called a weighted graph/ network. In a graph if edges point in particular direction say between nodes i and j there is an edge either going from i to j or j to i, then such a network is directed whereas a network with bidirected edges is called undirected graph. In this project visibility graph method is applied to form complex network for a time series.

2.2. <u>Degree distribution</u>:

Degree of a node i is the total number of edges connected to it. The distribution of probability (P_k) of a node having degree k is degree distribution. There are networks in which a long tail is observed in degree distribution which implies that higher degrees are possessed by few nodes and most nodes have less degree. In such long tail regions, power law is followed. Truncated power law distribution is where $P_k = ck^{-\gamma}$ is followed for $k > k_{min}$, otherwise area under probability distribution curve will reach infinite. Power law attributes to scale free nature of the network i.e., form or shape of the probability distribution remains invariant if we scale up or down the argument k. say we scale up k by a factor of s,

$$P_k(sk) = c(sk)^{-\gamma} = (s)^{-\gamma} \times \; P_k(k)$$

$$P_k(sk) \; \propto \; k^{-\gamma}$$

to get a better picture of long tail region of high degree nodes, logarithmic scale is taken for p_k and k. power law index γ will be the solpe of straight line in log-log plot. This plot becomes messy for higher degrees since there are very few nodes corresponding to high k values to average out the noise in distribution.

$$\text{Log}\ (p_{k)} = \log(C) + \gamma \log (k),$$

2.3.**<u>Louvain Algorithm for community detection</u>**:

Community is a group of nodes densely connected amongst themselves and sparsely connected to the rest of the network. Nodes with similar characteristics tend to form a community with more edges within a community as compared to edges between them. Community detection aims to find naturally occurring communities in a network. One way to do so for global networks is by Louvain's algorithm based on modularity maximization which focuses on maximizing the edges within a community. Modularity(Q) is

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{K_iK_j}{2m}\right]\delta(C_iC_j)\ , \quad -1 \leq Q \leq 1$$

Where i and j are nodes. k_i is the degree of i^{th} node which belongs to community C_i, m is the weighted sum of all edges in the network therefore 2m gives degree of undirected graph. A_{ij} is the weight of the links between i^{th} and j^{th} node. $\delta(C_iC_j)$ is Kronecker delta function representing whether nodes i and j belongs to same community or not. $\delta(C_iC_j)$ =1 if i and j belong to same community otherwise for $\delta\left(C_iC_j\right) = 0$, i and j are in different communities.

For a community c: $\quad Q_c = \frac{\sum in}{2m} - \left(\frac{\sum tot}{2m}\right)^2$

Where $\sum in$ is the sum of all edge weights within a community c and $\sum tot$ is sum of weights of all edges with at least of end or node in the community c. initially modularity optimization over all nodes give small communities. Each node is then assigned to its own community and modularity gain ΔQ is calculated on merging two nodes say i and j, where j is neighbour of i. in other words node i is removed from its own community and inserted into the neighbouring community of j change in modularity Q in this process is modularity gain. For positive gain, communities will merge else each node remains in its own community.

$$\Delta Q = \frac{\sum in + 2K_{i,in}}{2m} - \left(\frac{\sum tot + K_i}{2m}\right)^2 - \frac{\sum in}{2m} + \left(\frac{\sum tot}{2m}\right)^2 - \left(\frac{K_i}{2m}\right)^2$$

$$\Delta Q = K_{i,in} - \frac{\sum tot \times K_i}{(m)}$$

$K_{i,in}$ is the sum of edge weight between i and all other nodes in community that i is moving into. Here j represents the number of communities i is connected to, iteratively all nodes in our network are taken as i and modularity gain is estimated for all j corresponding to i. each community is then considered as a node and above process is repeated till there is no increment in modularity.

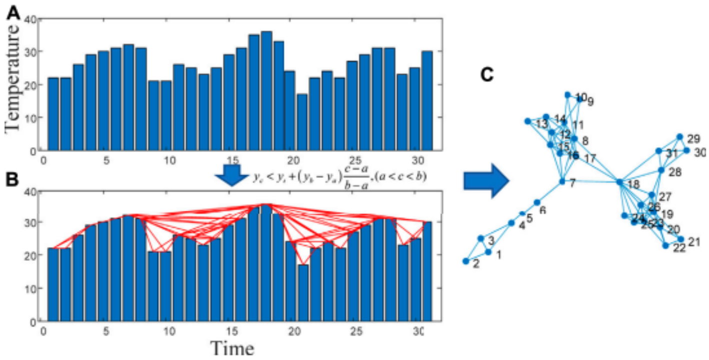
2.4.**<u>Visibility graph</u>**:If we have a time series or a data set for a given time period, we can construct a network based on visibility algorithm, which is a mathematical approach to analyse the nontrivial aspects of considered time series. we can plot a bar graph corresponding to a time series, and link all bars visible from top of the considered bar. Here a vertical bar represents a node and there is a link between two nodes when there is a straight line connecting them such that there is no obstacle or intermediate bar obstructing the straight line i.e., visibility exists between the nodes. Suppose there are two nodes A and B in x-y plane their coordinates are given by (x_a,y_a) and (x_b,y_b) respectively, equation of a straight line connecting these two nodes will be:

$$y = \frac{x-x_b}{x_a-x_b}(y_a - y_b) + y_b$$

If there is another point C (x_c,y_c) in between A and B then only possibility for a link to be formed between A and B is when hight of the of the vertical bar for C does not intersect the straight line y i.e.,

$$y_c < \frac{x_c-x_b}{x_a-x_b}(y_a - y_b) + y_b \text{ ....(1)}$$

Checking this condition for all nodes in our system we can get the set of all possible edges. There will always be a visibility between consecutive nodes each node must have at least a degree = 2. Edges given by this algorithm are undirected since if visibility exist between nodes i and j then edges (i,j) and (j,i) are generated.



This is an Illustration of the network construction via visibility graph theory. (A) represents histogram/ bar graph of original data or time series with time on x axis and temperature (in °C) on y axis. (B) network generated by VG theorem (C) network structure in the form of graph

Ref: Zhang P, Ning P, Cao R and Xu J (2021) Analysis of Climate Change Characteristics in Xi'an Based on the Visibility Graph.

An Algorithm can be developed to generate visibility graph based on the above-mentioned concept. All nodes in the time series (say N) are considered one by one. For a node i all preceding nodes j ranging from i+1 to N are checked for visibility to exist. For two consecutive nodes (j=i+1), edge is formed else for an edge to be formed condition (1) has to be satisfied for all nodes in between i and j. (Code of this algorithm is attached in appendix A)

3. RELATED WORK:

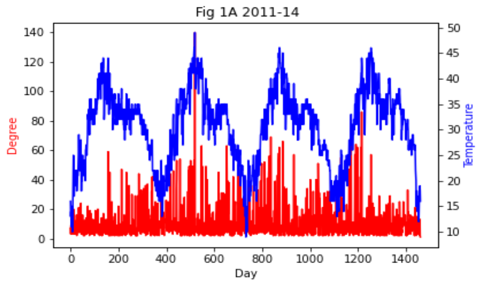
3.1.<u>Algorithm used in visibility graph</u>: To form visibility graph, there is a visibility_graph library available in python which can be imported to generate the network. Only difference in the algorithm behind it, is in the condition for an edge to be formed i.e., in equation (1). In this case an edge is formed if intervening node c between a and b, follows the condition:

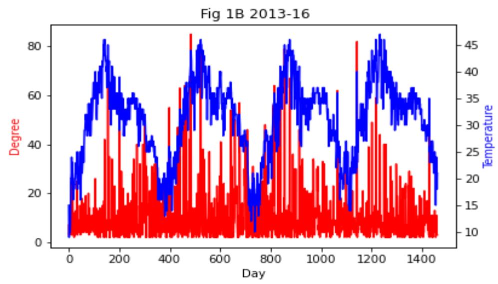
$$y_c \leq \frac{x_c-x_b}{x_a-x_b}(y_a - y_b) + y_b$$

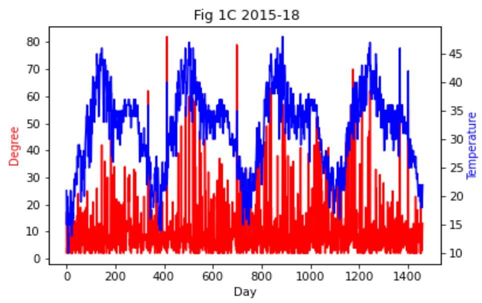
in this case we will get some extra edges, which will not affect the degree distribution and community structure to a much extent hence we will be sticking to this approach only.

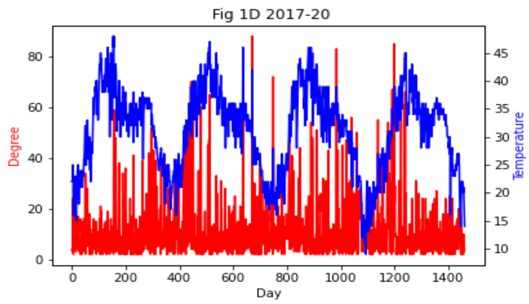
3.2.**<u>Degree characteristics of temperature network</u>:**To analyse the characteristics of temperature change in the three cities we constructed global network for each city from 01 January 2011 to 31 December 2020 and local networks by dividing into four 4 years' time series window and also plotted degree of each node. For example, time series from 01 January 2011 to 31 Dec 2014 constitute one window and the next window is taken from 01 January 2013 to 31 December 2016 and so on. Each local network has about 1461 nodes and around 730 nodes are common to every consecutive local network of time series windows whereas the global network has about 3653 nodes.

Our objective behind making such local networks is to analyse if temperature of last 2 years has any impact on the temperature of current two years and also if temperature of current two years is affecting temperature of next 2 years and so on. This helps us to find if we can see certain trend in the networks which can give us any evolutionary characteristics over time. Following plots are for **New Delhi.**









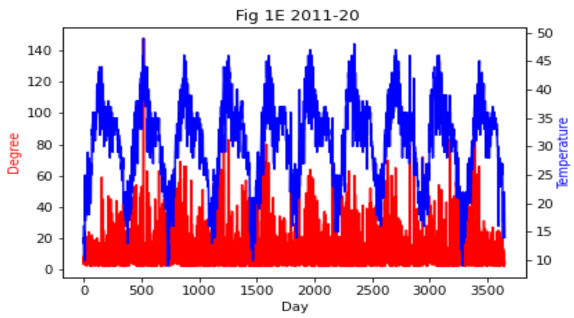


Fig 1A-1D are local temperature networks and Fig 1E is global network for city of New Delhi.

For the city of New Delhi, we can see that there is some periodic variation in the temperature with periodicity of 1 year and degree distribution also has a little periodicity as since degrees are rising for nodes in summer season. We can compare it with Xi'an that New Delhi is a landlocked, it has continental type of climate i.e. it will have extreme summers and extreme winters. This rise in degrees is obvious since with increase in temperature in summer months the visibility of nodes also increases. We also find out the nodes with maximum degree.

Table 1

Year	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Maximum Degree	59	140	69	86	79	82	63	88	83	85
Date	06/06	02/06	15/04	29/04	05/05	16/02	17/04	01/11	11/09	13/04

Here from the table, it can be understood that the maximum node in years 2011 and 2012 occurs in June whereas in case of remaining years it is occurring in April and May which can imply that temperature is fluctuating due to climate change with exception of some years. Please note that a day having maximum degree does not necessarily imply that it has the maximum temperature of the year as we can see that some years like 2018 and 2019 have maximum degree node in November and September.

Following plots are for **Mumbai.**

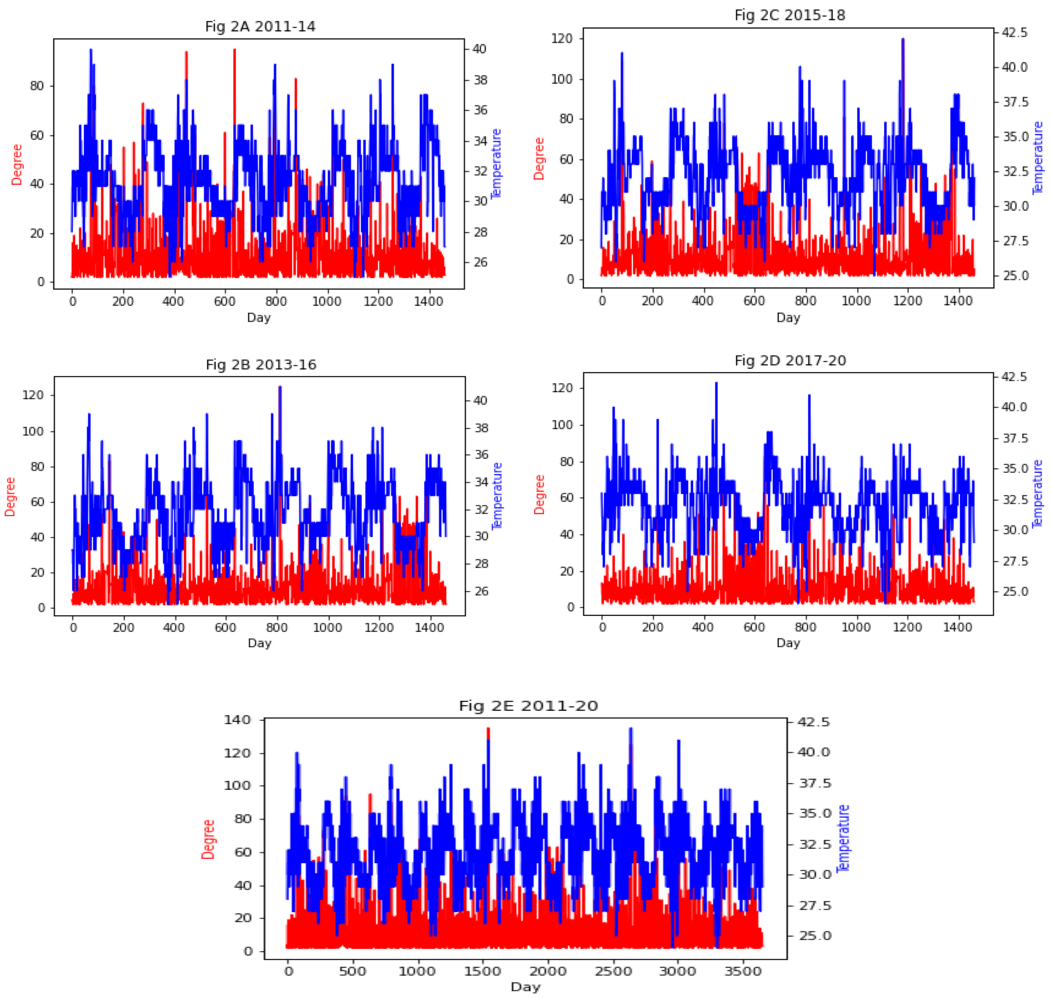


Fig 2A-2D are local temperature networks and Fig 2E is global network for city of Mumbai.

For the city of Mumbai, we can see that temperature variation is not very clear, there seems to be a very little periodicity in temperature and degrees also. This can be attributed to the fact that Mumbai is a coastal city and due to the moderating effect of sea it also does not have extreme temperatures. Summers and winters both are mild in Mumbai.

Table 2

Year	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Maximum Degree	80	95	88	79	125	78	81	123	106	51
Date	28/03	29/09	06/03	09/06	05/05	25/04	08/08	26/03	26/03	18/02

Here we can see that Table 2 is not showing any pattern. Nothing conclusive can be drawn from it.

Following plots are for the city of **Xi'an (China).**

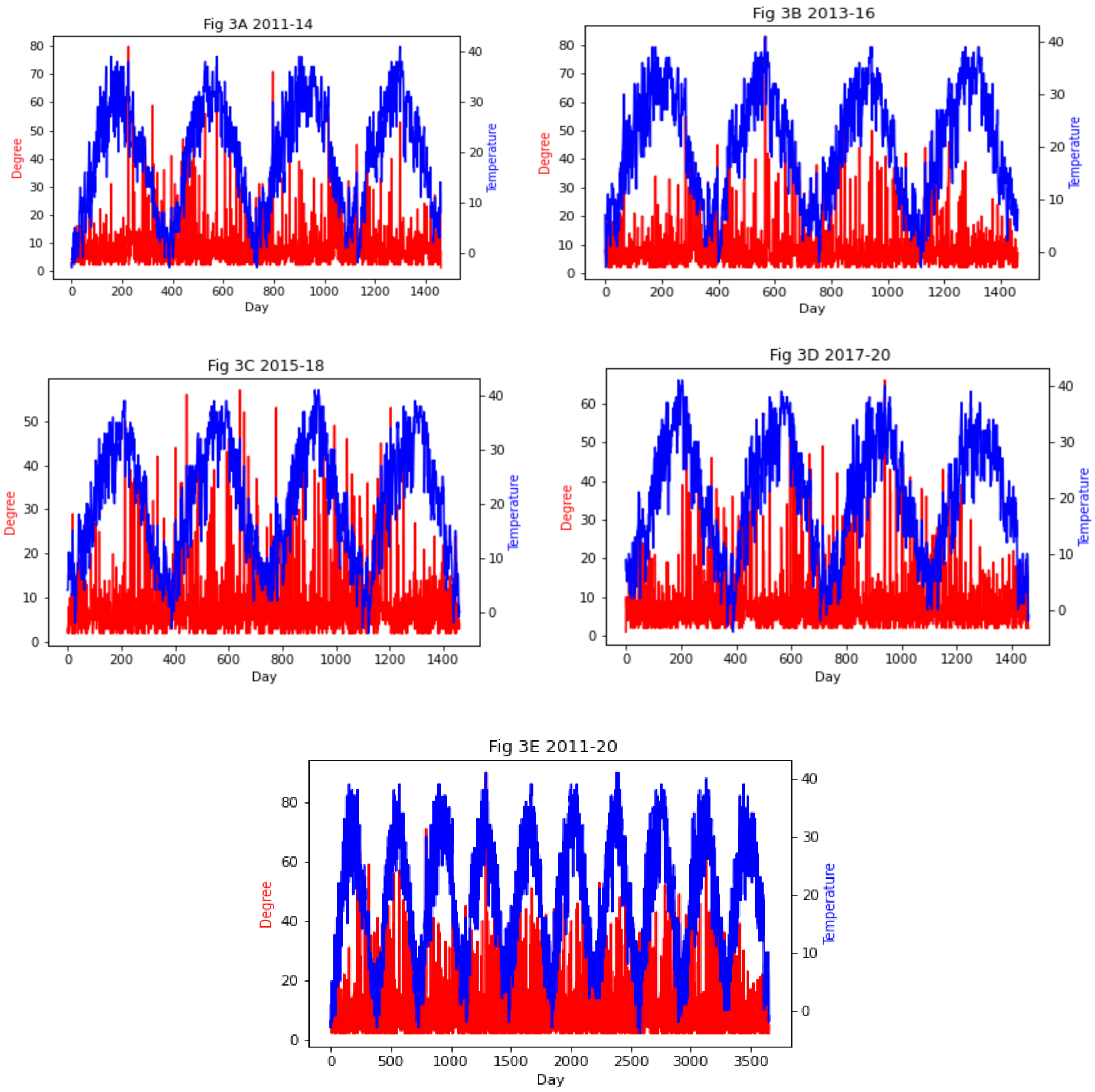


Fig 3A-3D are local temperature networks and Fig 3E is global network for city of Xi'an.

For the plots of Xi'an, the variation in temperature network is as per reference paper. It is not required to further elaborate on Xi'an as we are just using it to do a comparison.

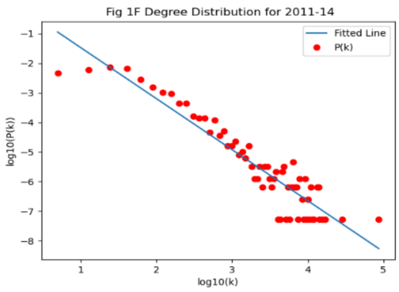
3.3. Degree Distribution

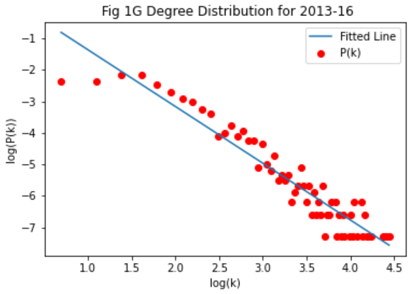
Degree distribution curves are plotted with degree on x axis and probability of a node having degree k on y axis in double logarithmic (base 10) scale. Fitted degree distribution curves are obtained by using power law function for the global network as well as for all 4-year windows. On log scale straight line with equation

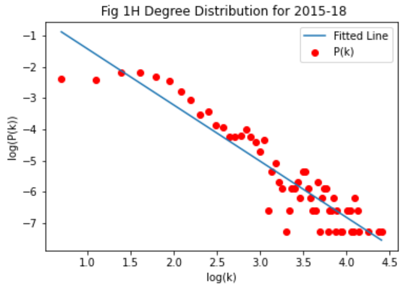
$$\log(p(k)) = \log(C) - \gamma \log(k)$$

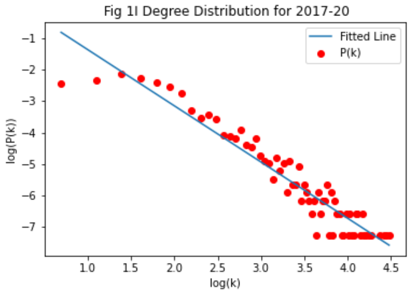
gives the fitted curve, slope of which gives the power law index. There are nodes of degree 2 which means they have edges on with their adjacent nodes and have no influence on global network so even if we combine them with node 3, our degree distribution shall not be affected.

Following plots are for **New Delhi**,









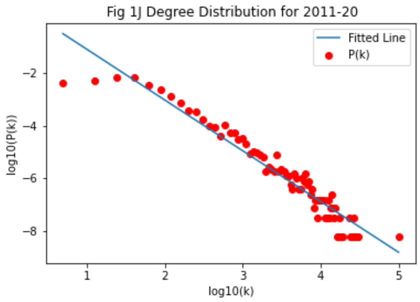


Fig 1F-1I are local degree distribution and Fig 1J is global network for city of New Delhi.

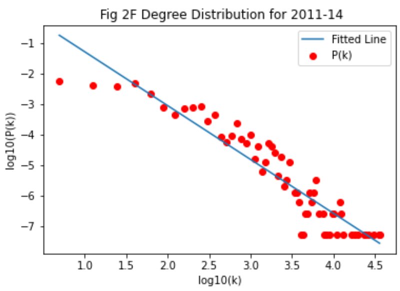
From fig 1.F to 1.J it can be seen that higher degrees are possessed by few nodes only, while most nodes have lower degree of visibility. High degree means temperature of that day/node has great influence on temperature subsequent node and is affected by its preceding node. such nodes represent extreme events.

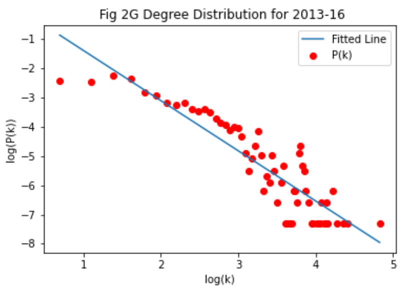
Table 4:

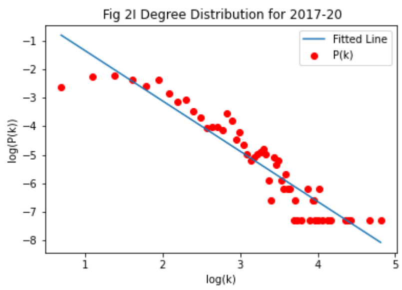
Year	2011-14	2013-16	2015-18	2017-20
Powe law index γ	1.7236	1.8030	1.7996	1.7939

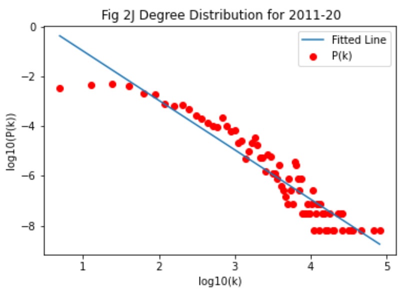
Power law index γ is 2.0396 for global network. Values γ for each 4 year time window are given in table 4, and they lie between 1.70 to 1.81. this corresponds to the scale free nature of our graph, time series for which is a fractal series. If you take a small part of the distribution and scale it, a pattern similar to the original distribution is obtained. This means temperature fluctuation in future are similar to that in past. γ is approximately similar for local and global network which implies temperature fluctuations in each time period follow similar trend in case of New Delhi.

Plots for **Mumbai**









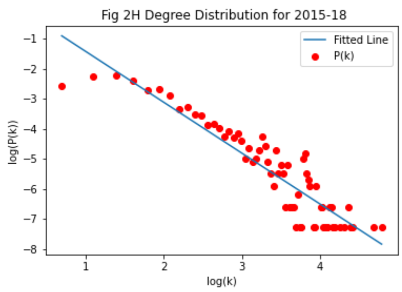


Fig 1F-1I are local degree distribution and Fig 2J is global network for city of Mumbai.

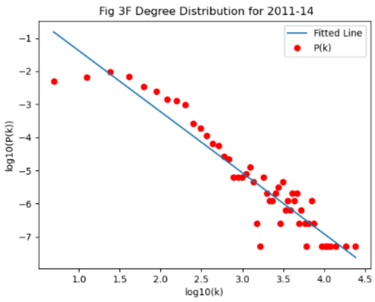
For Mumbai fig 2.F to 2.J it can be seen that higher degrees are possessed by very few nodes even for global network and most nodes have lower degree visibility.

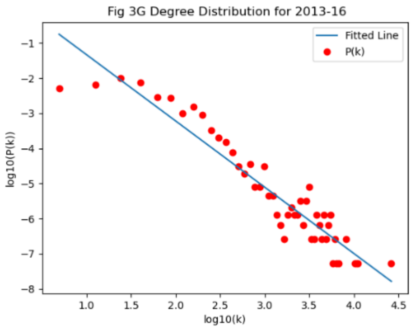
Table 5:

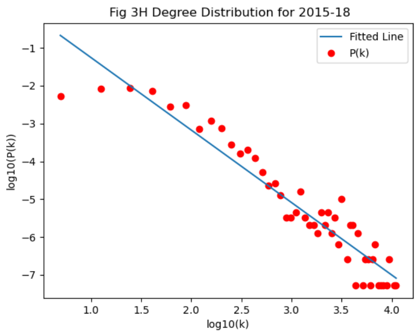
Year	2011-14	2013-16	2015-18	2017-20
Powe law index γ	1.7598	1.7156	1.6969	1.7602

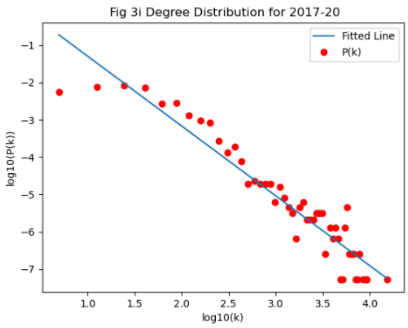
Power law index γ is 1.9904 for global network. γ for 4 year time windows for Mumbai are given in table 5, which ranges from 1.697 to 1.759, which accounts for the scale free characteristics of Mumbai temperature network.

Plots for **Xi'an (China)**









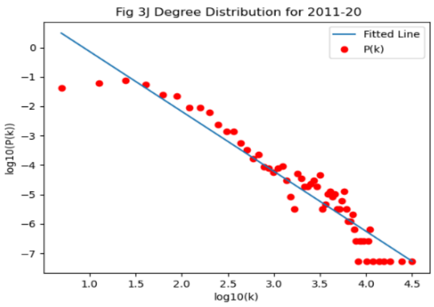


Fig 3F-3I are local degree distribution and Fig 3J is global network for city of Xi'an.

Table 6

Year	2011-14	2013-16	2015-18	2017-20
Powe law index γ	1.8440	1.8894	1.9162	1.8709

For global network of 10 years γ is 2.0396, and γ for local network ranges from 1.80 to 1.92. which shows that Xi'an network is also scale free and temperature fluctuations are similar over a time period. Our result is therefor in accordance with the refence paper.

3.4 COMMUNITY DETECTION :

To study the community structure of temperature network Louvain algorithm is used. To understand working of this algorithm different time slots with different total number of nodes are taken. Community structure of 4 years window, i.e. 2011-2014, 2012-2015 ,2013-2016 , 2014-2017 , 2015-2018 ,2016-2019 and 2017-2020, of Delhi's Temperature network is being compared with same 4 years in a decade temperature network. It is observed that some small communities which were there in network with only 4 years nodes is no longer there when we are using 10 years nodes (As shown in fig 3.4.1 for 2011-14).

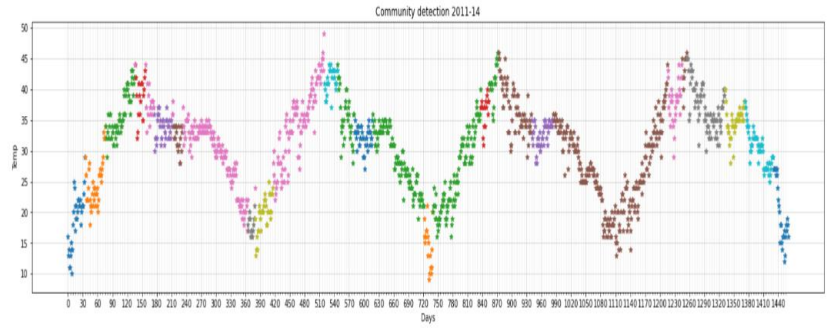


fig 3.4.1.1 : Temperature network communities for 2011 -14 from decade data for Delhi

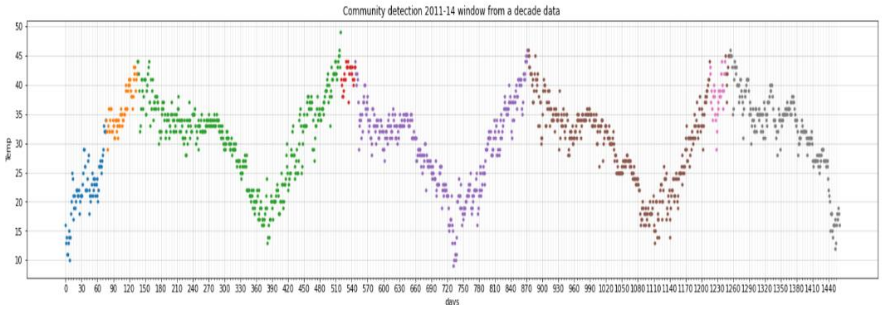


fig 3.4.1.2 : Temperature network communities for 2011-14 data Delhi

To explore about small communities quantitively bins of different time span , i.e. 3months and 4 months , are taken and number of small communities within these months are calculated . Communities detected are shown in Table:3.4.1 and Table: 3.4.2. For other time span tables are given in appendix B. It can be stated from all these table that number of small communities are going to decrease if there is increase in number of nodes . This can also be concluded from modularity which is given by fraction of edges that run between nodes of same type subtracted by fraction of such edges expected if edges were positioned at random without any regard for type of node. So increase in number of nodes implies that edges are going to increase but edges are going to increase in last year (for 2011-14 new edges with 2015 or onwards is going to start with 1389th node) so nodes in previous years are going to look more connect now compare to outside.

Table 3.4.1 : Small communities of temperature network for Delhi with time interval 3 months

MONTHS	TOTAL NUMBER OF SMALL COMMUNITIES							
	2011		2012		2013		2014	
	FROM 4 YEARS	FROM 10 YEARS	FROM 4 YEARS	FROM 10 YEARS	FROM 4 YEARS	FROM 10 YEARS	FROM 4 YEARS	FROM 10 YEARS
1JAN-31MARCH	2	1	2	1	0	0	0	0
1APRIL–30 JUNE	1	0	0	0	1	0	1	0
1JULY– 30 SEPT	1	0	1	0	1	0	0	0

1 OCT- 31 DEC	0	1	0	0	0	0	1	0

Table 3.4.2 Small communities of temperature network for Delhi with time interval 4 months

MONTHS	TOTAL NUMBER OF SMALL COMMUNITIES					
	1MAR2011-31FEB2012		1MAR2012-31FEB2013		1MAR2013-31FEB2014	
	FROM 4 YEARS	FROM 10 YEARS	FROM 4 YEARS	FROM 10 YEARS	FROM 4 YEARS	FROM 10 YEARS
1MAR-31JUNE	2	1	1	1	1	0
1JULY-31OCT	1	0	1	0	1	0
1NOV-31FEB	2	0	1	0	0	0

To detect whether or not there is some small communities which are repeating over the year analysis of small communities is done by moving a 1 year window over 4 year community structure as shown in Table: 3.3.3

Table 3.4.3.1 : Small communities in 3 month time interval for Delhi 2011-20 by moving window by 1 year in 4yrs community structure

Month	**Total number of communities**																										
	2011-14				***2012-15***				***2013-16***				***2014-17***				***2015-18***				***2016-19***				***2017-20***			
	11	***12***	***13***	***14***	***12***	***13***	***14***	***15***	***13***	***14***	***15***	***16***	***14***	***15***	***16***	***17***	***15***	***16***	***17***	***18***	***16***	***17***	***18***	***19***	***17***	***18***	***19***	***20***
Jan -Mar	2	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	0	1	0	0	1	0	0
April -June	1	0	1	1	1	1	1	0	1	1	0	0	1	0	1	2	0	1	1	1	1	1	1	1	1	2	2	0
July -Sept	1	1	1	0	0	2	1	0	2	0	0	0	1	0	0	0	0	0	2	0	0	2	0	0	2	0	1	0
Oct -Dec	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0

Table 3.4.3.2: : Small communities in 3 month time interval for Delhi 2011-20 by moving window by 1 year in 4yrs community structure

Month	***Total number of small communities***																				
	2011-14			***2012-15***			***2013-16***			***2014-17***			***2015-18***			***2016-19***			***2017-20***		
	11-12	***12-13***	***13-14***	***12-13***	***13-14***	***14-15***	***13-14***	***14-15***	***15-16***	***14-15***	***15-16***	***16-17***	***15-16***	***16-17***	***17-18***	***16-17***	***17-18***	***18-19***	***17-18***	***18-19***	***19-20***
Mar-may	1	0	1	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	0
Jun-Aug	2	1	0	1	2	1	3	0	0	1	0	0	0	0	1	0	1	0	1	0	1
Sept-Nov	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dec-Feb	2	1	0	1	0	0	0	0	0	0	0	1	0	1	1	0	2	0	1	0	1

From Table: 3.4.3.1 , we are getting at least one small community 8 times in April to June timeline over 10 years , 4 times in July to September timeline and 2 times in January to March . Since we are moving Window by 1 year so we are getting value at a specific year more than one times so we counted a community if it has probability more than ½ for that specific year

From Table: 3.4.3.2 ,we are getting one small community 5 times in June to August timeline , 2 small communities 2 times in June to August timeline , one small community 5 times in December-January-February and one small community 3 times in March-May timeline .

Different table are made via taking different time interval , i.e. 2 month-4months intervals which are included in Appendix B . At last It can be concluded that there can be small communities in high temperature zone from May to August and in low temp Region December-January-February .

For 10 years temperature network , different time interval are taken (as shown in Appendix B). Only one community can be detected which is April-May-June that also only 2 times in a decade.

For Delhi in 2011-20 temperature network , No significant pattern in small communities is detected but there can be seen a pattern in 4 years temperature network which is not in agreement with observation in original research on Analysis of climate characteristics in Xian based Visibility graph by Pengtao Zhang, Pengyi Ning, Runhua Cao and Jiwei Xu. So make better conclusion we have also tried to make community structure of Xian. We get a pattern as shown in fig:3.4.2. Small communities are detected and small community was observed for July to Sept 5 times in a decade (Appendix B). So algorithm is not making big difference.

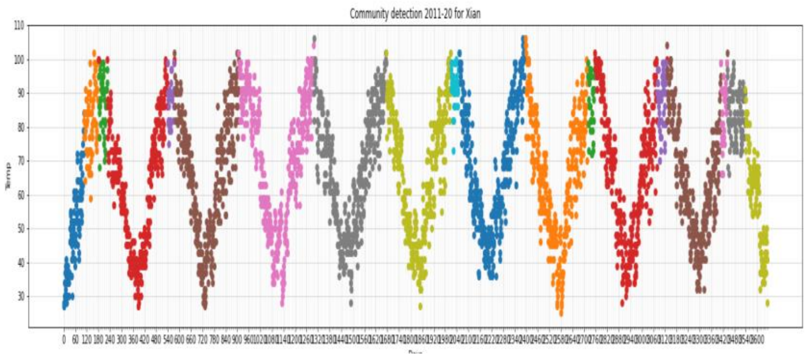


fig 3.4.2 : Community Structure 2011-20 for Xian

For Mumbai ,Community structure is for decade is a shown in fig: 3.3.3. By taking different time intervals it can be seen that we are getting a small community in March-May which is relatively high temperature.

Window of 4 years is also taken by moving window 1 year and small community in March-May and Sept-Nov is found but they are not that frequent.

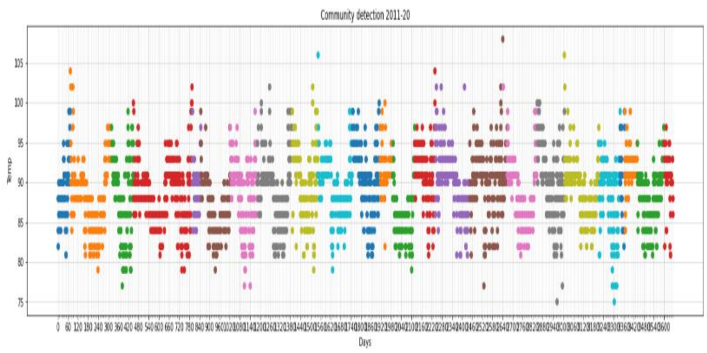


fig 3.4.3: Community Structure 2011-20 for Mumbai

An algorithm is developed by our group in which links which are at same line are not included and community structure is drawn using these links which is as shown in fig 3.4.4.

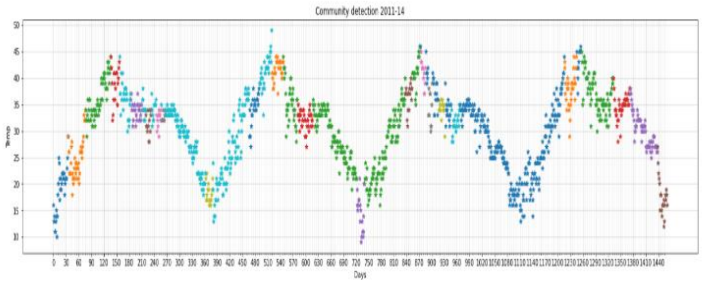


fig 3.3.4 : community detection via algorithm developed by our grp

Total number of communities are found to be 26 while from visibility graph library total number of communities were 21 Which is because total number of links are lesser in our algorithm. We are getting 6560 nodes while from visibility graph library we are getting 7555 links . To check whether it is affecting pattern of communities, small communities were detected in 2011-14 . It is found that most small communities via algorithm are in same region just their frequency increases so if we have done that for 10 year then also there can some change in frequency and we would have detected pattern but for Xian pattern is detected with Visibility Graph library so it should not make a major difference. Comparison tables is included in Appendix B.

4. CONCLUSION AND DISCUSSION :

Degree characteristics are being studied. No pattern is observed in maximum degree occurrence . For Delhi there is periodic variation in temperature with periodicity of 1 year but for Mumbai there is no definite periodic pattern. Xian pattern is also studied and found to be almost same as original paper

Degree Distribution is found to be scale free which is in as original paper. Reason behind it is just that extreme temp are rare event so most nodes are going to have less degree.

In Community Detection pattern is found for 4 year data there is no significant pattern observed in case of 10 years data for New Delhi. So small communities are detected for 4 year data. For New Delhi two significant small communities are found in high temperature zone from May to August and in low temp Region December-January-February .For Mumbai , a small community is found at relatively high temperature in March-May and one is found in Sept-Nov but they are less frequent (2 and 3 times in decade repectively). Xian's communities are also detected for 10 years data to verify and 5 small communities are detected.

It can be stated that original paper's claim regarding small communities in high temperature region and its relation with global warming is not global conclusion .Visibility graph does not seem a good method to study about climate change characteristics but it provides a method to study weather pattern if instead of 10 years if we would take longer time line may be some conclusion regarding global warming can be made.

5. AUTHOR CONTRIBUTIONS :

(1) Coding- Deepak , Diksha , Harvind , Yogita

(2) Report Writing :

(i) Abstract-Deepak

(ii) Introduction -Diksha , Harvind

(iii) Degree Characteristics and Degree Distribution- Deepak , Diksha

(iv)Community Detection- Harvind , Yogita

(v)Conclusion-Deepak, Diksha , Harvind , Yogita

6. ACKNOWLEDGEMENTS:

Foremost, we would like to express our sincere gratitude to our mentor Mr. Atiyab Zafar for amazing explanation of topic, advice, help and stimulating suggestions. we would also like to acknowledge with much appreciation crucial role of lectures by prof. Sanjay Jain (Department of physics & Astrophysics, University of Delhi), all that has led to the fulfilment of this project.

7. REFERENCES:

1. DATA AVAILABILITY FROM: *https://www.wunderground.com/history/monthly/cn/xi'an/ZLSN*
2. *Zhang P, Ning P, Cao R and Xu J (2021) Analysis of Climate Change Characteristics in Xi'an Based on the Visibility Graph.*
3. *M.E.J.Newman,Network An Introduction*
4. *https://towardsdatascience.com/louvains-algorithm-for-community-detection-in-python-95ff7f675306*
5. *Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nun (Septempber 29th, 2007) From time series to complex networks: The visibility graph*
6. *https://github.com/rgarcia-herrera/visibility_graph/blob/56ebee948e61a15bf096aaa405bab0fd5048d75c/visibility_graph/__init__.py#L23*

APPENDIX A

In this appendix we have included the python code for the figures under the heading degree characteristic of temperature networks and degree distribution networks. We have used a python library called visibility graph for creating visibility graphs although we have also created our own program for it which is also provided further

1. For figures 1A-1E, 2A-2E and 3A-3E (we have provided the code for say just 1A, for the rest of figures the same code can be used to generate plots by changing some variables and data files for the region and respective time.)

```
#New Delhi
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
from visibility_graph import visibility_graph
import math
import community
import partition


TT1114=pd.read_csv("temp1114.csv")
x1=TT1114["Time"]
y1=TT1114["Temp"]
G1=visibility_graph(y1)
d1=dict(nx.degree(G1))
fig,ax1=plt.subplots()
ax2=ax1.twinx()
ax1.plot(d1.keys(),d1.values(),'r')
ax2.plot(d1.keys(),y1,'b')
ax1.set_xlabel("Day")
ax1.set_ylabel("Degree",color='r')
ax2.set_ylabel("Temperature",color='b')
nx.write_edgelist(G1,'1set.txt')
plt.title(" Fig 1A 2011-14")
plt.savefig('1set.png')
```

2. For the figures from 1F-1J, 2F-2J and 3F-3J (we are providing code for 1F), in this case we have plotted the degree distribution along with a fitted line, to get a best fit we have removed some nodes having degree 1 and degree 2.For finding nodes with maximum degrees in each year we have also divided the dictionary created in previous program into 4 parts for every year.

```
import numpy as np
tn1=G1.number_of_nodes()
import math
d11 =dict(list(d1.items())[:len(d1)//4])
d21 =dict(list(d1.items())[len(d1)//4:len(d1)//4+365])
d31=dict(list(d1.items())[len(d1)//4+365:len(d1)//4+730])
d41=dict(list(d1.items())[len(d1)//4+730:])
```

```
#for 2011-14
u=max(d1.values())
v=max(d1,key=d1.get)
print("Maximum degree for 2011-14=",u,"Corresponding day",v)

#for 2011
u=max(d11.values())
v=max(d11,key=d11.get)
print("Maximum degree for 2011=",u,"Corresponding day",v)

#for 2012
u=max(d21.values())
v=max(d21,key=d21.get)
print("Maximum degree for 2012=",u,"Corresponding day",v)

#for 2013
u=max(d31.values())
v=max(d31,key=d31.get)
print("Maximum degree for 2013=",u,"Corresponding day",v)

#for 2014
u=max(d41.values())
v=max(d41,key=d41.get)
print("Maximum degree for 2014=",u,"Corresponding day",v)


degree=[k for n,k in G1.degree()]
for i in range (1,degree.count(1)+1):
    degree.remove(1)
    degree.remove(2)
    degree.append(3)
values=sorted(set(degree))
hist=[np.array((degree.count(x))/tn1) for x in values]
plt.xlabel('log10(k)')
plt.ylabel('log10(P(k))')
l11=np.log(values)
l21=np.log(hist)
a,b=np.polyfit(l11,l21,1)
plt.plot(l11,a*l11+b)
plt.scatter(l11,l21,color='r')
print("slope=",a,"Intercept=",b)
plt.title(" Fig 1F Degree Distribution for 2011-14")
plt.legend(["Fitted Line","P(k)"],loc="upper right")
plt.savefig("DD1.png")
```

3. Following code was created for visibility graph algorithm in python

```
import networkx as nx
import numpy as np
def vg(n,y=[]):
    g=nx.Graph()
    x=[]
    for p in range(n):
        x.append(p)
```

```
    A=np.empty((0,2),int)
for i in range(n):
    for j in range(i+1,n):
        if j==i+1:
            A=np.append(A,np.array([[x[i],x[j]]]),axis=0)
        else:
            f=0
            for k in range(i,j):
                if y[k]<y[i]+(y[j]-y[i])*((x[k]-x[i])/(x[j]-x[i])):
                    f=f+1
            if f==j-i-1:
                A=np.append(A,np.array([[x[i],x[j]]]),axis=0)
g.add_nodes_from(x)
g.add_edges_from(A)
return(g)
```

Please note that in the condition for a straight line between two nodes for visibility graph ,in the library of visibility graph we are using , a link between i and jth node will be created if value of kth node is either lying on the straight line connecting i and j or less than it. Although in original reference paper the condition is taken to be less than but we think that according to concept of visibility if value of kth node is lying on the line connecting i and j it is not obstructing the visibility of j from I since visibility theory is quite abstract in nature, we cannot take it in a way like physically imagining ourselves if we view from top of i, if we can see j and k or not. It is just that a straight line can be connected from I to j without a node obstructing the connection

4. Code for Community detection and its analysis is as follows:

```
#For Community detection

import networkx.algorithms.community as nx_comm

com1=nx_comm.louvain_communities(G1)

j1=np.array(com1)

#for New Delhi (2011-14)  Community Structure Visualization

t=0

plt.rcParams["figure.figsize"] = (25,5) #To change size of figure

fig = plt.figure()

ax = fig.add_subplot(1, 1, 1)

for k in j1:#k th community

    w1={}

    for j in k:#j th element of k th community

        for i in x1a:#i th element of time series

            if (i==j):

                w1[x1a[i]]=y1a[i]

    ax.plot(w1.keys(),w1.values(),'*')

    t=t+1

plt.title("Community detection 2011-14")
```

```
print("Total number of communities",t)
plt.xlabel("Days")
plt.ylabel("Temp")



# Major ticks every 30, minor ticks every 5
major_ticks = np.arange(0, 1460, 30)
minor_ticks = np.arange(0, 1460, 5)

ax.set_xticks(major_ticks)
ax.set_xticks(minor_ticks, minor=True)

#  different settings for the grids:
ax.grid(which='minor', alpha=0.2)
ax.grid(which='major', alpha=0.5)
```

for calculating small communities :

```
#JAN2011-DEC2020 WITH 6 DIVISION PER YEAR CYCLE
for l in range(0,1460,61):
    r1=[]
    b=0
    for i in range(l,l+61):
        r1.append(i)
    r1a=np.array(r1)
    for k in j1:
        w1={}
        s1=[]
        s2=[]
        for j in k:
            s1.append(j)
            for i in r1a:
                if (i==j):
                    w1[j]=y1a[i]
                    s2.append(i)
        if(s1==s2):
            b=b+1
```

print(b,l,l+61)

APPENDIX B

Month	Total number of communities																											
	2011-14				*2012-15*				*2013-16*				*2014-17*				*2015-18*				*2016-19*				*2017-20*			
	11	*12*	*13*	*14*	*12*	*13*	*14*	*15*	*13*	*14*	*15*	*16*	*14*	*15*	*16*	*17*	*15*	*16*	*17*	*18*	*16*	*17*	*18*	*19*	*17*	*18*	*19*	*20*
Jan-Feb	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0	1	0	0
Mar-April	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
May-Jun	1	1	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0
July-Aug	1	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
Sept-Oct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Nov-Dec	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0

<u>Table for small community detection</u>

1. For New Delhi :

(i) Jan - Dec Cycle with 6 division per year : No small community found

Month	*Total number of small communities*																				
	2011-14			*2012-15*			*2013-16*			*2014-17*			*2015-18*			*2016-19*			*2017-20*		
	11-12	*12-13*	*13-14*	*12-13*	*13-14*	*14-15*	*13-14*	*14-15*	*15-16*	*14-15*	*15-16*	*16-17*	*15-16*	*16-17*	*17-18*	*16-17*	*17-18*	*18-19*	*17-18*	*18-19*	*19-20*
Feb-Mar	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
April-May	0	0	1	0	0	1	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0
June-July	1	1	0	1	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Aug-Sept	0	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1
Oct-Nov	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
Dec-Jan	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	2	0	1	0	1

(ii) Feb2011-Jan2022 with 6 division per year : community in June-July appearing 3 times

(iii) Jan2011-Dec2020 with 3 division per year :

- Community in Jan to April appearing 6 times
- Community in March to August appearing 6 times
- Community in September to December appearing 3 times

• Month	Total number of communities																											
	2011-14				*2012-15*				*2013-16*				*2014-17*				*2015-18*				*2016-19*				*2017-20*			
	11	*12*	*13*	*14*	*12*	*13*	*14*	*15*	*13*	*14*	*15*	*16*	*14*	*15*	*16*	*17*	*15*	*16*	*17*	*18*	*16*	*17*	*18*	*19*	*17*	*18*	*19*	*20*
Jan-April	2	1	1	0	`1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1	0	0
May-Aug	3	1	0	1	1	2	1	0	3	0	0	0	1	0	1	1	0	1	1	0	0	1	0	0	1	1	1	0
Sept-Dec	0	0	0	2	1	0	0	1	0	0	0	1	0	0	1	1	0	1	1	1	0	1	0	1	0	0	0	2

(iv) Feb2011-Jan2020 with 4 division per year :

- Community in Feb-May appearing 6 times
- Community in June-Sept appearing 6 times with 2 communities 2 times and 3 communities 1 time
- Community in Oct-Nov-Dec-Jan appearing 5 times

Month	***Total number of small communities***																				
	2011-14			***2012-15***			***2013-16***			***2014-17***			***2015-18***			***2016-19***			***2017-20***		
	11-12	***12-13***	***13-14***	***12-13***	***13-14***	***14-15***	***13-14***	***14-15***	***15-16***	***14-15***	***15-16***	***16-17***	***15-16***	***16-17***	***17-18***	***16-17***	***17-18***	***18-19***	***17-18***	***18-19***	***19-20***
Feb-May	2	0	1	1	1	1	1	1	0	0	1	2	0	2	1	1	1	1	0	1	0
June-Sept	2	2	1	1	3	2	3	1	0	2	0	0	0	0	2	0	2	0	2	0	2
Oct-Jan	1	1	0	1	0	3	0	0	0	0	0	2	0	2	1	0	2	0	1	0	1

(v) Jan2011-Dec2020 with 4 division per year :

- Community in April-June appearing 8 times
- Community in July-Sept appearing 4 times
- Community in Jan-Mar appearing 2 times

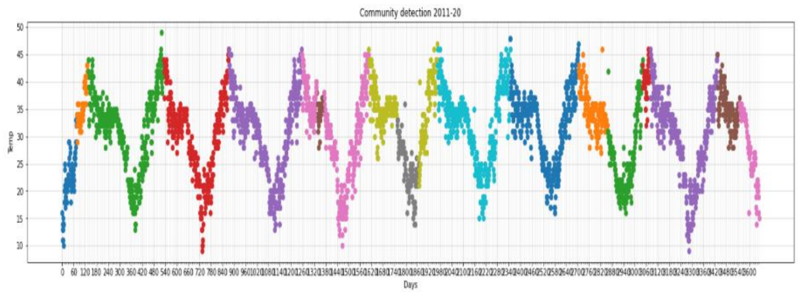
Month	**Total number of communities**																											
	2011-14				***2012-15***				***2013-16***				***2014-17***				***2015-18***				***2016-19***				***2017-20***			
	11	***12***	***13***	***14***	***12***	***13***	***14***	***15***	***13***	***14***	***15***	***16***	***14***	***15***	***16***	***17***	***15***	***16***	***17***	***18***	***16***	***17***	***18***	***19***	***17***	***18***	***19***	***20***
Jan-Mar	2	0	0	0	1	1	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	0	1	0	0	1	0	0
April-June	1	0	1	1	1	1	1	0	1	1	0	0	1	0	1	2	0	1	1	1	1	1	1	1	1	2	2	0
July-Sept	1	1	1	0	0	2	1	0	2	0	0	0	1	0	0	0	0	0	2	0	0	2	0	0	2	0	1	0
Oct-Dec	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0

(vi) Mar2011-Feb2020 with 4 division per year:

- Community in June- August appearing 5times with 2 communities 2 times
- Community in Dec-Jan-Feb appearing 5times

Month	***Total number of small communities***																				
	2011-14			***2012-15***			***2013-16***			***2014-17***			***2015-18***			***2016-19***			***2017-20***		
	11-12	***12-13***	***13-14***	***12-13***	***13-14***	***14-15***	***13-14***	***14-15***	***15-16***	***14-15***	***15-16***	***16-17***	***15-16***	***16-17***	***17-18***	***16-17***	***17-18***	***18-19***	***17-18***	***18-19***	***19-20***
Mar-may	1	0	1	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0	1	0
Jun-Aug	2	1	0	1	2	1	3	0	0	1	0	0	0	0	1	0	1	0	1	0	1
Sept-Nov	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Dec-Feb	2	1	0	1	0	0	0	0	0	0	0	1	0	1	1	0	2	0	1	0	1

(vii) 10 year data



Month	Total Number of Small Communities								
	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20
Mar-June	1	0	0	0	0	0	0	0	1
July-Oct	0	0	0	1	0	0	0	0	0
Oct-Jan	0	0	0	0	0	0	0	0	0

Month	Total Number of Small Communities									
	11	12	13	14	15	16	17	18	19	20
Jan-Mar	1	0	0	0	0	0	0	0	0	0
April-June	0	0	0	1	0	0	0	0	1	0
July-Sept	0	0	0	0	0	0	0	0	0	0
Oct-Dec	0	0	0	0	0	0	0	0	0	0

2. For Mumbai

Mar2011-Feb2020

- community appearing in March -May 2 times
- one community appearing in Sept-Nov 3 times

Month	*Total number of small communities*																				
	2011-14			*2012-15*			*2013-16*			*2014-17*			*2015-18*			*2016-19*			*2017-20*		
	11-12	*12-13*	*13-14*	*12-13*	*13-14*	*14-15*	*13-14*	*14-15*	*15-16*	*14-15*	*15-16*	*16-17*	*15-16*	*16-17*	*17-18*	*16-17*	*17-18*	*18-19*	*17-18*	*18-19*	*19-20*
Mar-may	0	0	1	0	1	0	1	0	0	0	0	0	1	1	0	1	0	0	0	2	0
Jun-Aug	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sept-Nov	0	1	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	0
Dec-Feb	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

For Xian :

Feb2011-Jan2022 (10 years data)

Month	Total Number of Small Communities								
	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20
Feb-may	0	0	0	0	0	0	0	0	0
June-Sept	1	1	0	0	0	1	0	1	1
Oct-Jan	0	0	0	0	0	0	0	0	0

<u>Comparison b/w algorithm developed by group and Visibility Graph Library</u>

Month	*Total number of small communities(2011-14)*					
	VISIBILITY GRAPH LIBRARY			*ALGORITHM DEVELOPED*		
	11-12	***12-13***	***13-14***	***11-12***	***12-13***	***13-14***
Mar-may	1	0	1	1	0	1
Jun-Aug	0	0	0	2	1	1
Sept-Nov	0	1	0	2	0	0
Dec-Feb	0	0	0	1	1	0