DETECTION OF DISASTER RELEVANT EVENTS BY INVESTIGATING MICROBLOG TWEETS

RESEARCH INTERNSHIP FOR STUDENTS OF OTHER INSTITUTIONS

DEPARTMENT OF INFORMATION TECHNOLOGY
SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING
KALAVAKKAM, CHENNAI 603 110

Submitted by

ROSHAAN S (2127200801072)

Information Technology
Sri Venkateswara College of Engineering(SVCE)
Post Bag No.1, Pennalur Village, Chennai-Bengaluru Highways,
Sriperumbudur(off Chennai) Tk – 602117, Tamil Nadu.
III Year

JULY 2022

SSN COLLEGE OF ENGINEERING : CHENNAI 603110

BONAFIDE CERTIFICATE

Certified that project report titled DETECTION OF DISASTER RELEVANT EVENTS BY INVESTIGATING MICROBLOG TWEETS is the bonafide work of ROSHAAN S (2127200801072 – III YEAR), Sri Venkateswara College of Engineering(SVCE) who carried out the project work under our supervision as a research internship project in the Department of Information Technology, SSN College of engineering during July – August 2022.

SIGNATURE
Dr. C. Aravindan

HEAD OF THE DEPARTMENT
Professor
Department of Information Technology
SSN College of Engineering
Kalavakkam - 603 110.

SIGNATURE
Dr. S. SASIREKHA

SUPERVISOR
Associate Professor
Department of Information Technology
SSN College of Engineering
Kalavakkam - 603 110.

Place: Kalavakkam

Date: 04-09-2022

ACKNOWLEDGEMENT

I would like to express my sincere thanks to Sri Sivasubramaniya Nadar College of Engineering, for giving me this opportunity to take part in this Research Internship for Students of Other Institutions.

I also wish to express my sincere gratitude to **Dr. C. ARAVINDAN**, Professor and Head of the Department of Information Technology, SSN College of Engineering.

With utmost respect, I would like to express my sincere gratitude to my mentor **Dr. S. SASIREKHA**, Associate Professor, Department of Information Technology, for helping and guiding me at every step of this project which made this research successful. In the course of this intern project, I got an insight into Disaster event detection and analyzing real-time data.

I wish to express my sincere thanks to all the panel members for their suggestions and encouragement during the review sessions.

ROSHAAN S

ABSTRACT

Microblogging is a brief blog post for quick and direct audience involvement. These short blogs are distributed via social media sites such as Twitter or Instagram and can comprise a variety of content formats such as audio, text, photographs, or video. Disaster events occur unexpectedly, and their effects are very destructive. These Disaster events in various locations come to public notice through different microblogging platforms. Twitter is one platform where people share details of various events it gets instantly received by a colossal section of the population. This project focuses on extracting details of the disaster events, particularly the event's location, and storing it as historical data. Various analyses and studies can be performed on historical data to forecast future disaster events in a particular area by forming specific patterns by studying the data.

The system is built by collecting live Twitter data. This live streaming data is analyzed to get insights about the event's location and marked on a map to create a GIS database. This database can be investigated and studied to forecast future events.

Keywords: Disaster Events, Twitter, Location, Historical data, Microblogging.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Microblogging is a method of sharing content that uses text, images, and videos, and they are more engaging, relatable, and personal. Twitter, Instagram, and LinkedIn are some common Microblogging sites. However, the focus is mainly on microblog tweets since data fetching is more straightforward, and various plugins are available. There are 238 million Twitter users worldwide, and 500 million tweets are posted daily on average. A massive number of Tweets are posted regarding disaster events every second. These tweets describe those events like where the disaster occurred, its impact and effect, and people's emotions towards it, and also contain images representing the disaster. The data extracted is analyzed with various methodologies to extract geo information about the disaster event. For better understanding, this project focuses on only floods, and flood events are detected by investigating microblog tweets.

During disaster events, rescue operations play a significant role. These response operations are inefficient when information regarding the event is not received correctly. Delay in responses may significantly impact saving lives and response actions. This project aims to reduce the delay, increase the speed of response, and make it more effective.

Fig 1.1 shows some examples of tweets that represent floods. Specific insights, particularly the location of the flood that is Rajasthan and Kangra, can be obtained from the pictures.

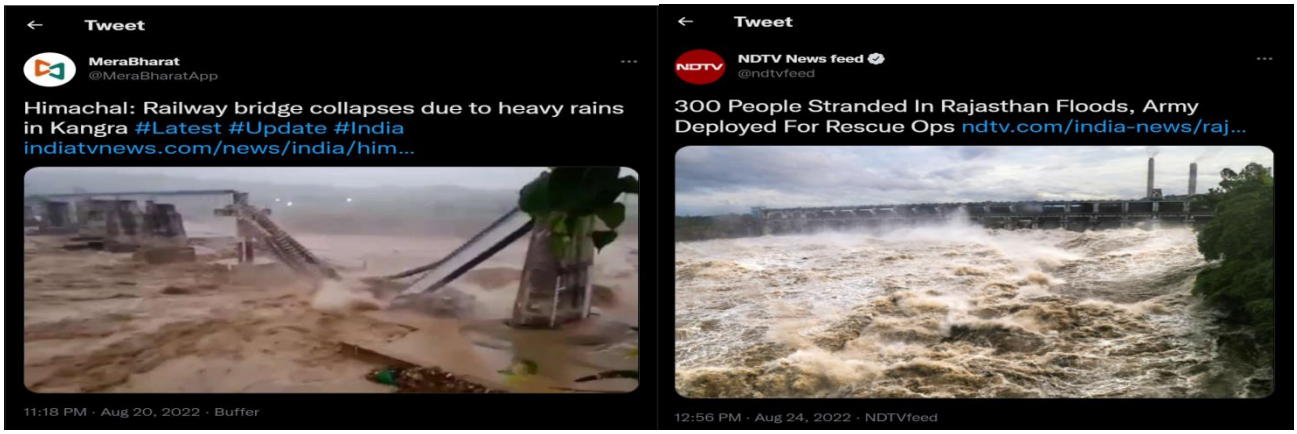


Fig 1.1 Example of Flood tweets

2. LITERATURE SURVEY

S. M. DedarAlam discussed creating a model which is built to monitor particular hashtag-related tweets, identify the actual disaster tweet, and extract essential data from disaster tweets, such as location and keywords. An API is developed with this data which detects the location and displays a disaster statistics message to the user.[1]

Alvaro Cuesta et al discuss that Tweets are available online in static and streaming formats, accessible via Twitter's public API. One can search for recent tweets using specific search criteria in their static form. The outcome of these searches is always a fixed view at a specific point in time. The streaming form displays an infinite stream of tweets that must be filtered by at least one criterion (free API access) accessible via a REST endpoint. A model is created that predicts the sentiment of the collected data handled by the user. Various sentiment analysis algorithms are used to implement the model.[2]

Meng WangJining et al discuss that the NLTK library, based on the Python programming language, can give more flexible and rich research methodologies and use uniform data standards to reduce the hassle of data type translation. Simultaneously, with the support of Python's various third-party libraries, it may compensate for other tools' inadequacies in syntax analysis, graphic rendering, regular expression retrieval, and other areas.[3]

Anthony Ivan discusses the Folium Library's ability to visualize the location of the data points which allows us to quickly relate the data points we have to the real world and regarding geographical data, using maps instead of other charting forms allows us to emphasize trends, identify patterns, and disclose previously hidden truths.[4]

Nayan Ranjan Paul et al designed two hybrid deep neural network models CNN-LSTM and CNN-GRU by merging both CNN, LSTM, and CNN, GRU networks for event identification on Twitter data during a crisis Both algorithms can properly detect the presence of disaster-related events in Twitter data.[5]

Anna Kruspe et al discuss that keywords and hashtags are the most commonly utilized for research, and they frequently act as valuable pre-filter for data collecting. The Twitter API allows researchers to search directly for keywords and hashtags and record the live stream of tweets containing those terms, so this approach is frequently a helpful starting point for them. Crowdsourcing approaches were followed for the analysis of data.[6]

Jyoti Prakash Singh et al defined a model that fetches tweets from Twitter and classifies them as low-priority and high-priority. High-priority tweets are analyzed to predict the location of the event. Three approaches are used for indicating the location: finding it from the tweet, geolocation, and Markov model to predict location based on historical data.[7]

In this project, a model is defined which collects live streaming data from Twitter. Various data cleaning approach is used with the NLTK library. Tweet data is analyzed to identify the event's location, which is precisely marked on a map to derive better inference. Fig 2.1 below depicts the system model of the project.

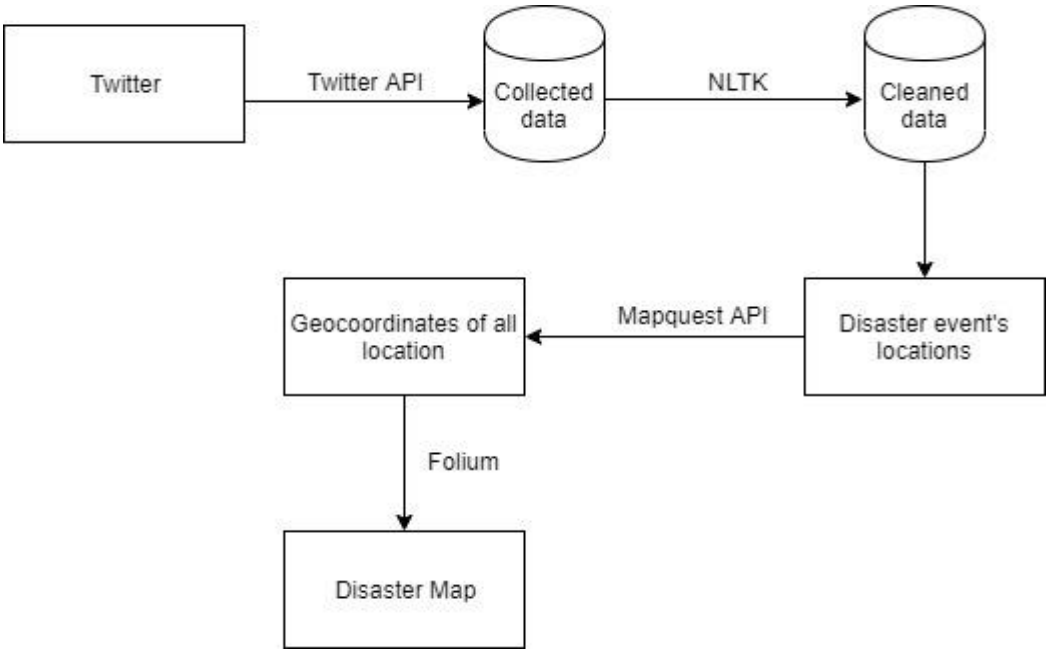


Fig 2.1 System model diagram

3. DATA COLLECTION

This section gives a detailed view of how data (tweets related to floods)is collected from Twitter.

Twitter API

The Twitter API enables the collection and analysis of Twitter data effectively. Resources like Tweets, Users, Spaces, Trends, Media, Places, etc. can be fetched with this API. Twitter API is accessed by signing up for a developer account and creating a Token Key and Secret. The figure below is the Twitter development portal screen which displays API keys.

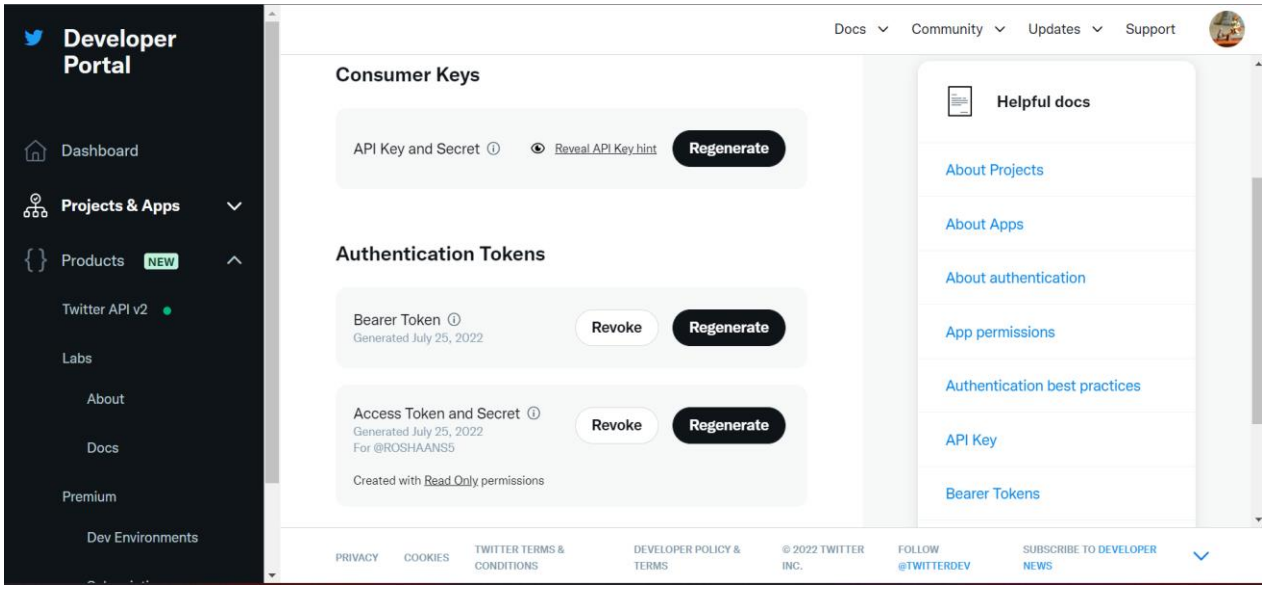


Fig 3.1 Access Token and Key in Twitter Developer Portal

Tweepy

Tweepy is an open-source Python module for accessing the Twitter API. Authentication is achieved by the OAuthHandler function. Tweets are extracted using the Cursor function based on specific keywords floods, disasters, and India.

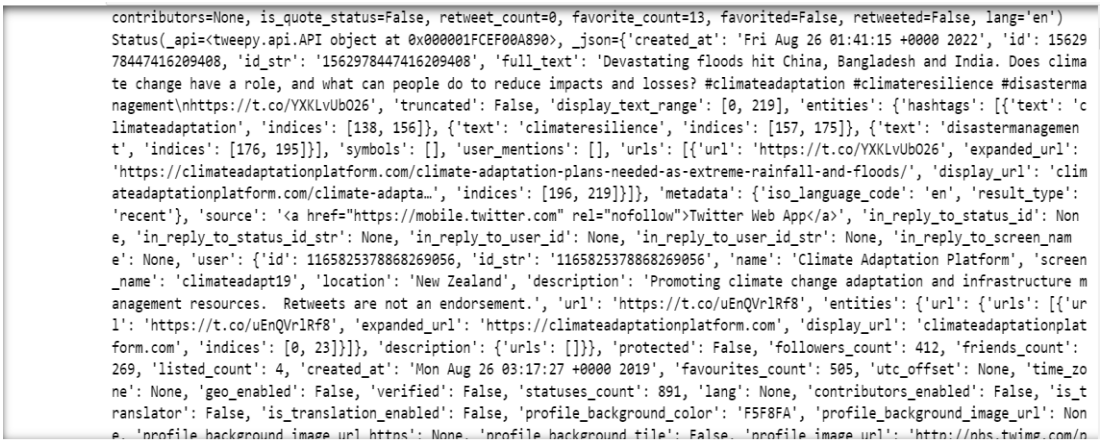
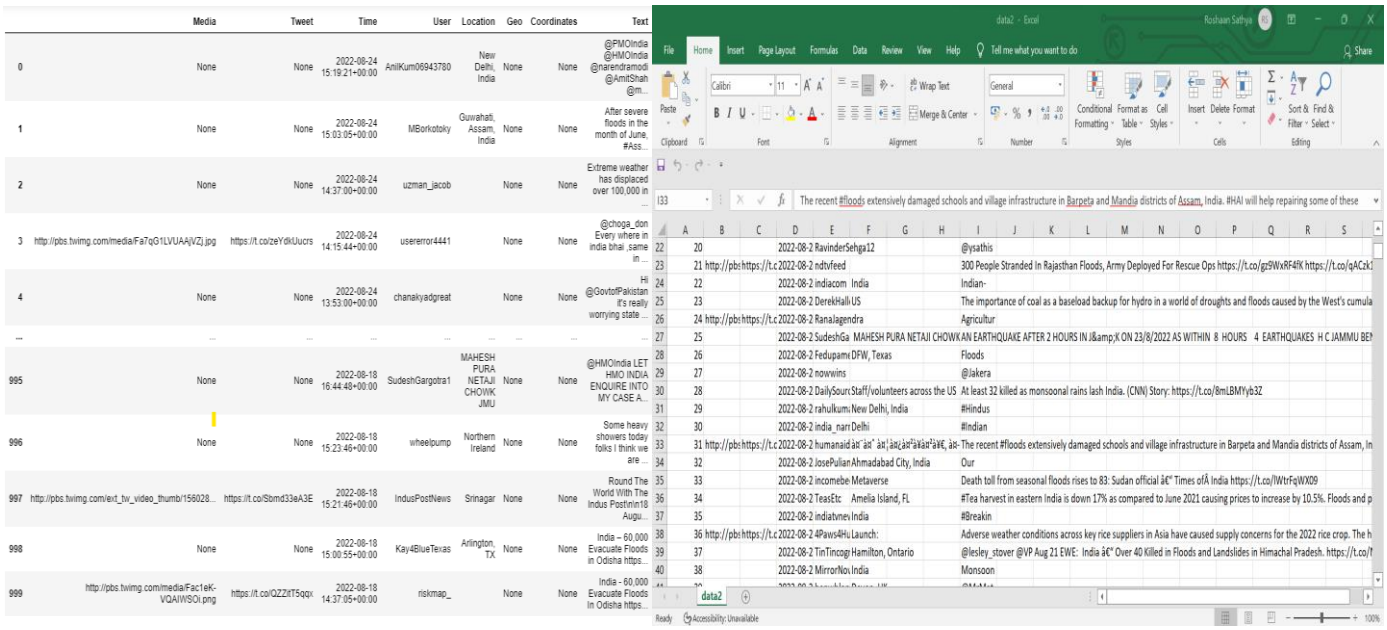


Fig 3.2 Real-time extracted Twitter data

Fig 3.2 shows the live streaming Twitter data which has to be processed for further procedure. The extracted Twitter data is stored in the form of a data frame with attributes time, user, location, text, etc. Later, the data frame is stored as a CSV file for better analysis. The figures below show how data is stored in a data frame and CSV file.



	Media	Tweet	Time	User	Location	Geo	Coordinates	Text
0	None	None	2022-08-24 15:19:21+00:00	AnilKum06943780	New Delhi, India	None	None	@PMOIndia @HMOIndia @narendramodi @AmitShah @m…
1	None	None	2022-08-24 15:03:05+00:00	MBorkotoky	Guwahati, Assam, India	None	None	After severe floods in the month of June, #Ass…
2	None	None	2022-08-24 14:37:00+00:00	uzman_jacob		None	None	Extreme weather has displaced over 100,000 in …
3	http://pbs.twimg.com/media/Fa7qG1LVUAAjVZj.jpg	https://t.co/zeYdkUucrs	2022-08-24 14:15:44+00:00	usererror4441		None	None	@choga_don Every where in india bhai ,same in …
4	None	None	2022-08-24 13:53:00+00:00	chanakyadgreat		None	None	Hi @GovtofPakistan It's really worrying state …
…	…	…	…	…	…	…	…	…
995	None	None	2022-08-18 16:44:48+00:00	SudeshGargotra1	MAHESH PURA NETAJI CHOWK JMU	None	None	@HMOIndia LET HMO INDIA ENQUIRE INTO MY CASE A…
996	None	None	2022-08-18 15:23:46+00:00	wheelpump	Northern Ireland	None	None	Some heavy showers today folks I think we are …
997	http://pbs.twimg.com/ext_tw_video_thumb/156028…	https://t.co/Sbmd33eA3E	2022-08-18 15:21:46+00:00	IndusPostNews	Srinagar	None	None	Round The World With The Indus PostInIn18 Augu…
998	None	None	2022-08-18 15:00:55+00:00	Kay4BlueTexas	Arlington, TX	None	None	India – 60,000 Evacuate Floods in Odisha https…
999	http://pbs.twimg.com/media/Fac1eK-VQAIWSOi.png	https://t.co/QZZIeT5qqx	2022-08-18 14:37:05+00:00	riskmap_		None	None	India - 60,000 Evacuate Floods In Odisha https…

Fig 3.3 Twitter data stored in the data frame and CSV

4. DATA CLEANING AND VISUALIZATION

NLTK is a Python tool for creating programs that work with human language data and are used for natural language processing (NLP). It contains libraries for text-cleaning purposes. re is a library in python which works with regular expressions. itertools is a python library used to iterate over a data structure. Collections in python are used for storing data. Pandas is used to handle CSV files and data frames.

The natural language toolkit(NLTK) and re module in python are used for data cleaning. Using these modules, stop words from tweet text are removed, and then the text is converted into a lower case for uniformity and better analysis.

A bag of words is created from tweet text. Using the itertools and collections module, this list of bag of words is iterated, and words that represent the event's location are stored in a data frame with its corresponding frequency. Matplotlib is a library in python which is used for data visualization. Using matplotlib, the words representing location are plotted against their frequency.

The figure below represents the location of disaster events and their frequency in tweets. From the figure, it can be derived that Odisha has more frequency and understood that there are severe floods in Odisha followed by that Uttarkhand and Assam.

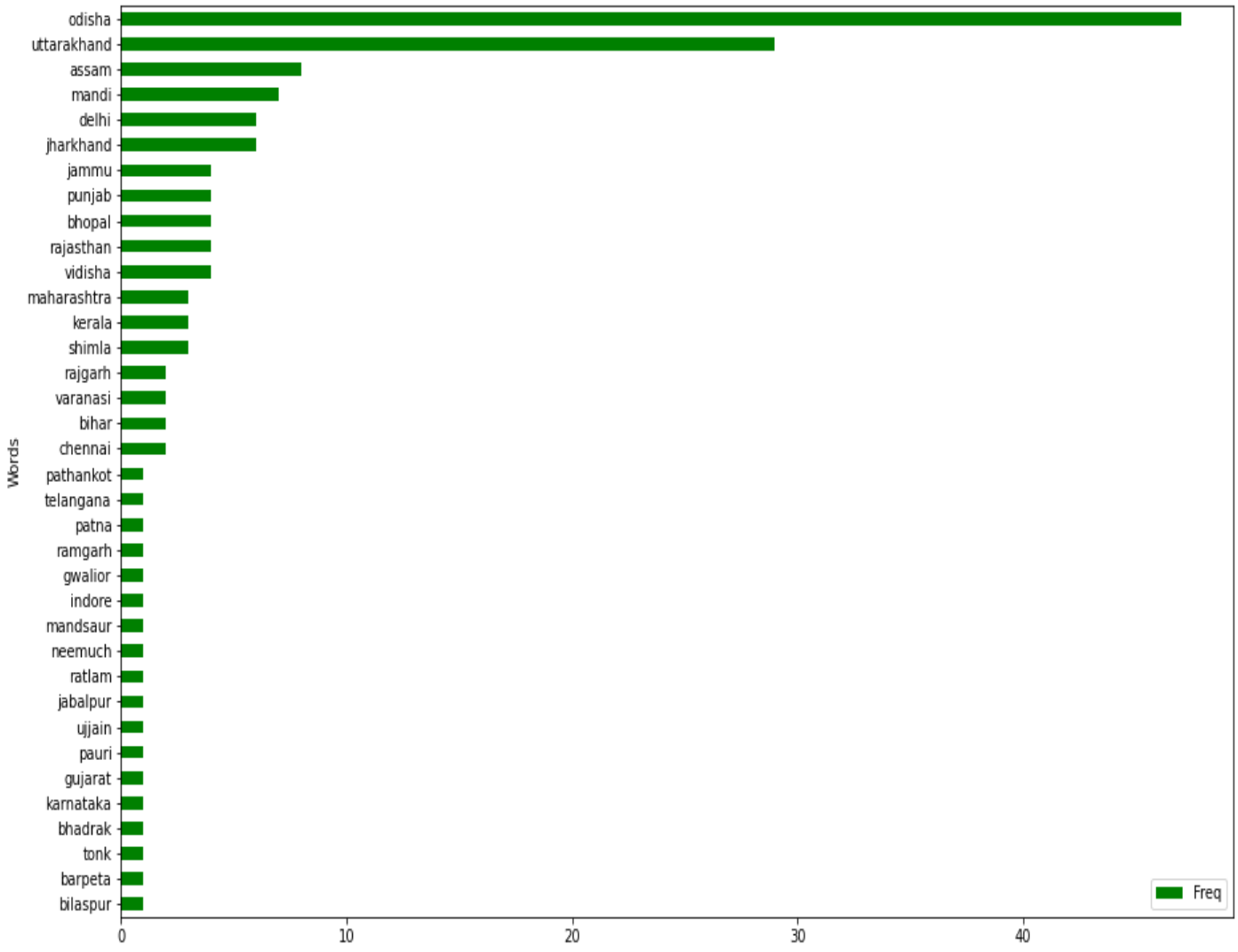


Fig 4.1 Frequency of Tweets for each location

Clustering is performed to visualize the location points as clusters. For this purpose, two clustering algorithms K-means and DBSCAN algorithm is used. One text vectorizer that converts text into a useful vector is the term frequency-inverse document frequency. This TFIDF vectorization is used to convert tweet text into vectors, and then a clustering algorithm is applied.

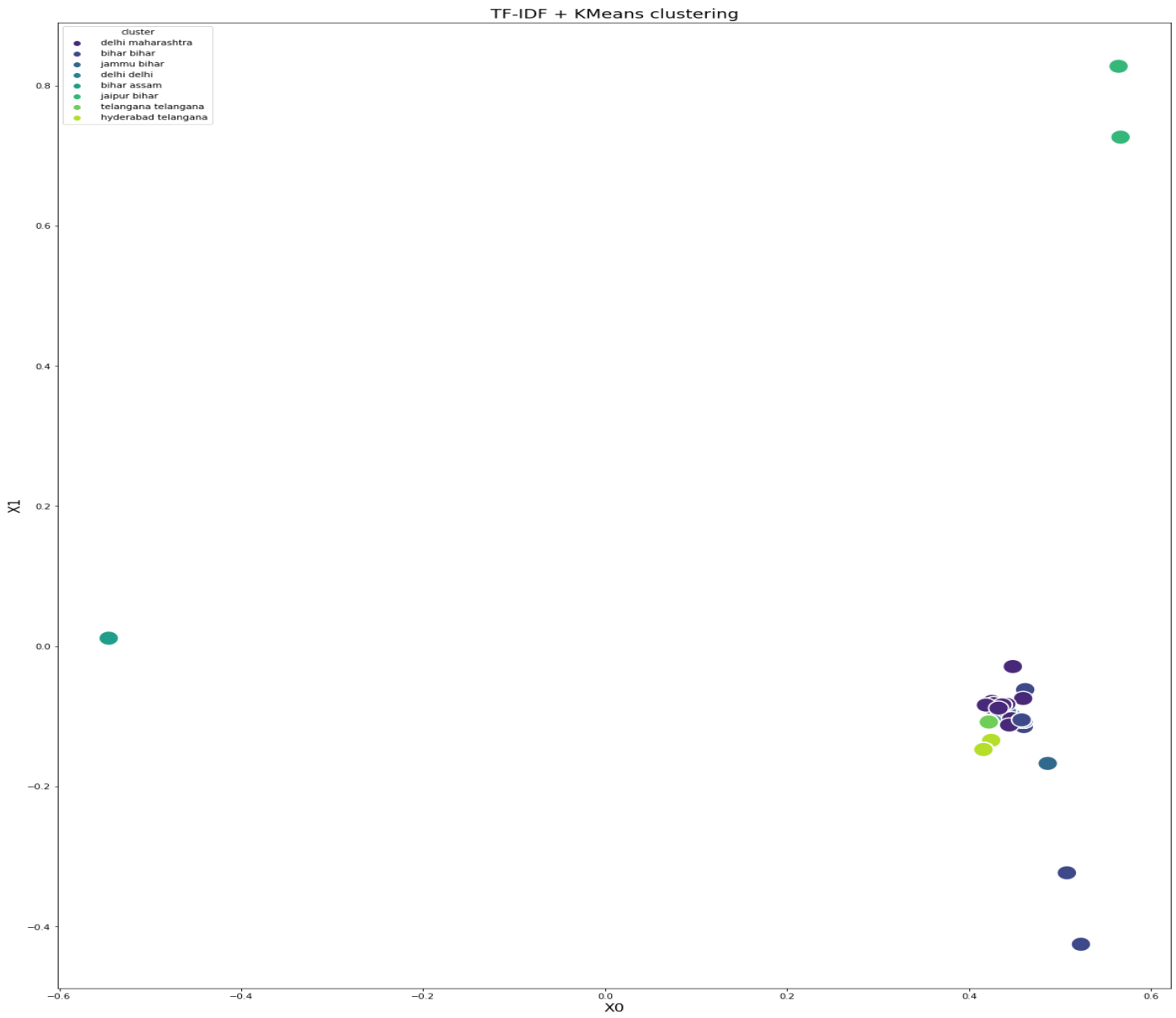


Fig 4.2 K-means Clustering

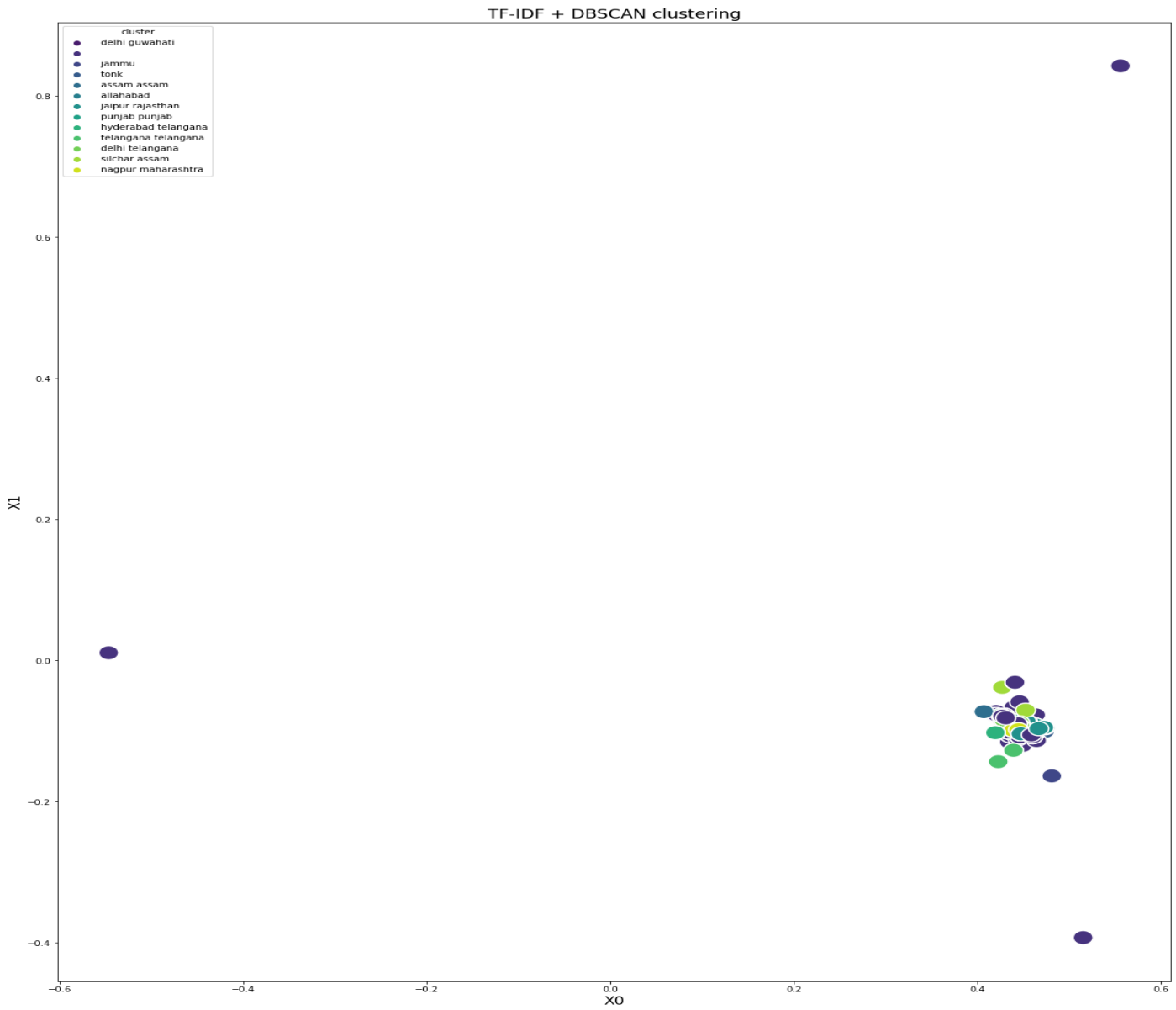


Fig 4.3 DBSCAN Clustering

Fig 4.2 and 4.3 are outputs of K-means and DBSCAN clustering algorithms. The location clusters are represented in a scatter plot.

Image collected from the tweet data is also processed. An open-source "Fine-tuned" flood detection model is used for this purpose. Using this model, it is tested whether the image collected from tweet data is related to the flood or not. Fig 4.4 depicts how the image is tested with a flood detection model.

```
In [101]: from IPython.display import Image
          Image(filename='evaluate/f8.jpg', width=300,height=200)
```

Out[101]:

```
In [102]: preprocessed_image = preprocess_image('f8.jpg')
          predictions = model.predict(preprocessed_image)

In [103]: predictions

Out[103]: array([[0.99754554, 0.00245439]], dtype=float32)

In [104]: result = np.argmax(predictions)

In [105]: if(result == 0):
              print("Flood Image")
          else:
              print("False")

          Flood Image
```

Fig 4.4 Flood detection model

5. IMPLEMENTATION AND RESULTS

A label “Category” is created against each tweet text to indicate the event's location described in each tweet. It is stored in a data frame and then transformed into a CSV file. Fig 5.1 show how these data are stored in a CSV file.

11	11			2022-08-2	kalyugend	Delhi, India			contrary to the popu	[]
12	12			2022-08-2	indiacom	India			#rajastha	[]
13	13			2022-08-2	latestly	Mumbai, India			madhya	['vidisha', 'vidisha']
14	14	http://pbs	https://t.c	2022-08-2	rahat_up	Lucknow			follow	[]
15	15	http://pbs	https://t.c	2022-08-2	GlobalWatchCGTN				floods and landslides	[]
16	16			2022-08-2	darpanani	New Delhi, India			from	['assam', 'odisha']
17	17	http://pbs	https://t.c	2022-08-2	firefighter	UK			northern india: 40 kil	[]
18	18	http://pbs	https://t.c	2022-08-2	hiindia	Chicago, IL			hindu	[]
19	19			2022-08-2	republic	Mumbai, India			maharashtra govt to	['maharashtra']
20	20			2022-08-2	RavinderSehga12				@ysathis	['telangana']
21	21	http://pbs	https://t.c	2022-08-2	ndtvfeed				300 people stranded	['rajasthan']
22	22			2022-08-2	indiacom	India			indian-	[]
23	23			2022-08-2	DerekHall	US			the importance of co	[]
24	24	http://pbs	https://t.c	2022-08-2	RanaJagendra				agricultur	[]
25	25			2022-08-2	SudeshGa	MAHESH PURA NETAJI CHOWK			an earthquake after	['jammu', 'jammu']
26	26			2022-08-2	Fedupame	DFW, Texas			floods	[]
27	27			2022-08-2	nowwins				@jakerac	[]

Fig 5.1 CSV with the label of event’s location

This label attribute of the CSV file is converted to a list using the tolist() function. This list contains the location of all flood events. A data frame is created with the list elements, and a CSV file is made from this data frame using the to_csv function. All these processes are implemented with the pandas library in python. Fig 5.2 shows the list of event locations.

```
In [6]: l

Out[6]: ['assam',
         'vidisha',
         'vidisha',
         'assam',
         'odisha',
         'maharashtra',
         'telangana',
         'rajasthan',
         'jammu',
         'jammu',
         'barpeta',
         'assam',
         'odisha',
         'odisha',
         'rajasthan',
         'tonk',
         'assam',
         'varanasi',
         'odisha',
```

Fig 5.2 List of event locations

The CSV file containing the location is used to obtain geocoordinates to mark it on the map. Fig 5.3 shows the CSV file containing all locations.

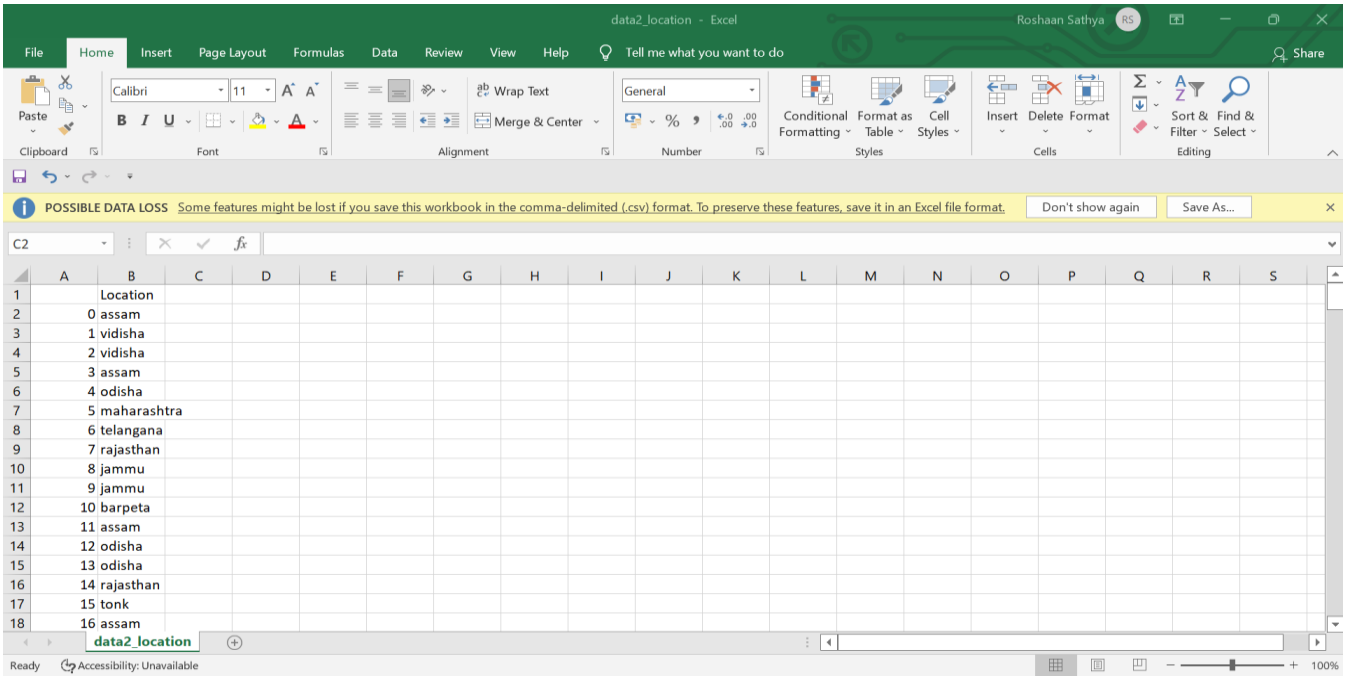


Fig 5.3 Event locations CSV file

Map API is used to retrieve location data. This API returns data which includes latitude and longitude of input location. This geoinformation is extracted using Map API.

MapquestAPI is an open-source API that can be accessed by API key. Using Mapquest API, each location's latitude and longitude details are gathered and stored in a data frame and saved as a CSV file. Fig 5.4 and 5.5 depicts how data with coordinates are stored in the data frame and CSV file respectively.

0	lat	lng
assam	26.140000	91.770000
vidisha	23.522200	77.803600
vidisha	23.522200	77.803600
assam	26.140000	91.770000
odisha	54.250000	-4.500000
maharashtra	19.531932	76.055457
telangana	17.990000	79.590000
rajasthan	26.810578	73.768455
jammu	32.718561	74.858092
jammu	32.718561	74.858092
barpeta	26.325782	91.007571
assam	26.140000	91.770000

Fig 5.4 Location data frame with coordinates

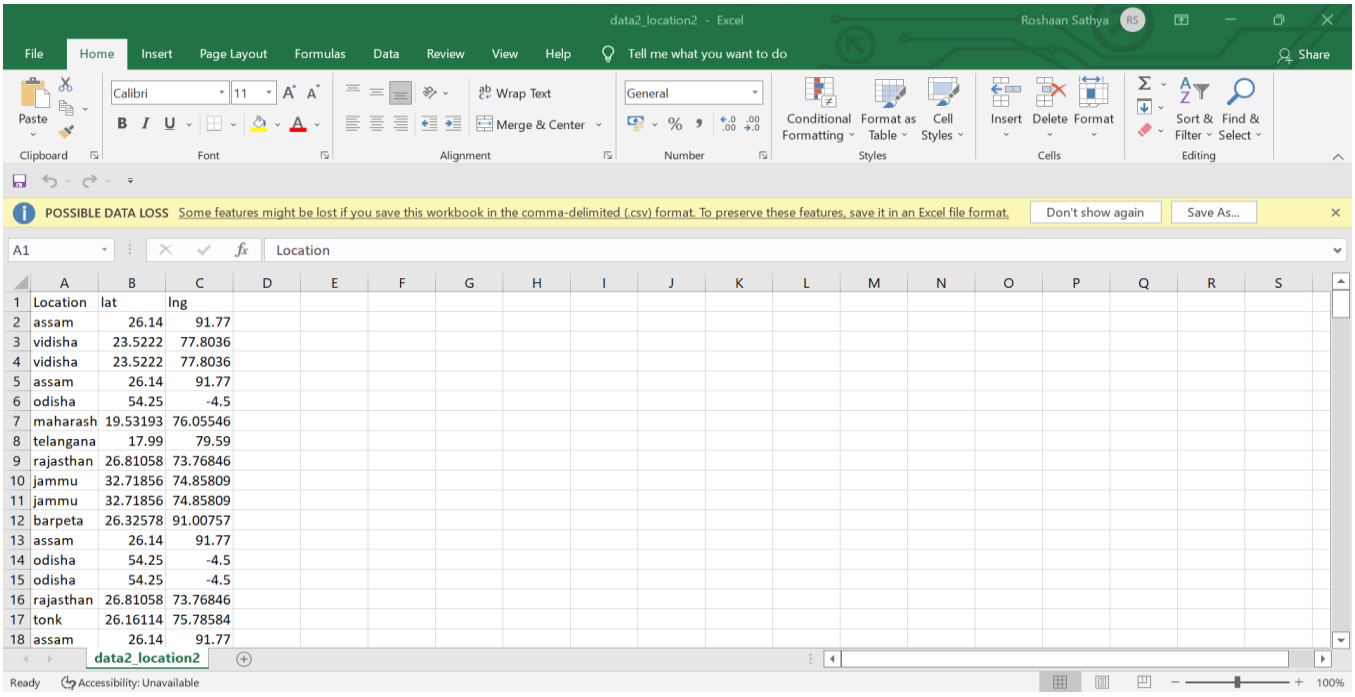


Fig 5.5 Location CSV file with coordinate

Web scraping is used to obtain additional inferences like temperature, precipitation, wind speed, etc., at the time of the event. Web scraping is a method of getting data from websites. requests_html is a python library used for web scraping. HTML content is parsed, and required data is collected.

Google weather data is parsed to obtain necessary details. Data is extracted by parsing respective CSS selectors. Based on the input location, the corresponding information is given. Fig 5.6 shows the collected data stored in a data frame against each location.

```
In [28]: df.head()
```

Out[28]:

	Location	lat	lng	Tmperature in C	Precipitation in %	Humidity in %	Wind Speed	Description
0	assam	26.1400	91.7700	31.0	7%	84%	3 km/h	Fog
1	vidisha	23.5222	77.8036	27.0	2%	89%	5 km/h	Clear with periodic clouds
2	vidisha	23.5222	77.8036	27.0	2%	89%	5 km/h	Clear with periodic clouds
3	assam	26.1400	91.7700	31.0	7%	84%	3 km/h	Fog
4	odisha	54.2500	-4.5000	29.0	5%	94%	5 km/h	Haze

Fig 5.6 Location data frame with all attributes

Folium is an easy-to-use module in python used to create maps and visualize geospatial data. It uses leaflets for creating the map. Maps created using Folium are interactive and customizable.

The Data frame containing location and attributes is converted into a nested list in which each list has a place, its temperature, humidity, and wind speed, at the time of the event. Fig 5.7 depicts the nested list.

```
In [30]: a_list

Out[30]: [['assam', 26.14, 91.77, 31.0, '7%', '84%', '3 km/h', 'Fog'],
          ['vidisha',
           23.5222,
           77.8036,
           27.0,
           '2%',
           '89%',
           '5 km/h',
           'Clear with periodic clouds'],
          ['vidisha',
           23.5222,
           77.8036,
           27.0,
           '2%',
           '89%',
           '5 km/h',
           'Clear with periodic clouds'],
          ['assam', 26.14, 91.77, 31.0, '7%', '84%', '3 km/h', 'Fog'],
          ['odisha', 54.25, -4.5, 29.0, '5%', '94%', '5 km/h', 'Haze'],
```

Fig 5.7 Nested list containing flood locations and related attributes

A function is traversing each list and marking it on a map. In folium, there is a function that enables a popup. The inferences, particularly temperature, humidity, and wind speed, are represented as a popup in the marker. Fig 5.8 show the flood map with all the locations marked.

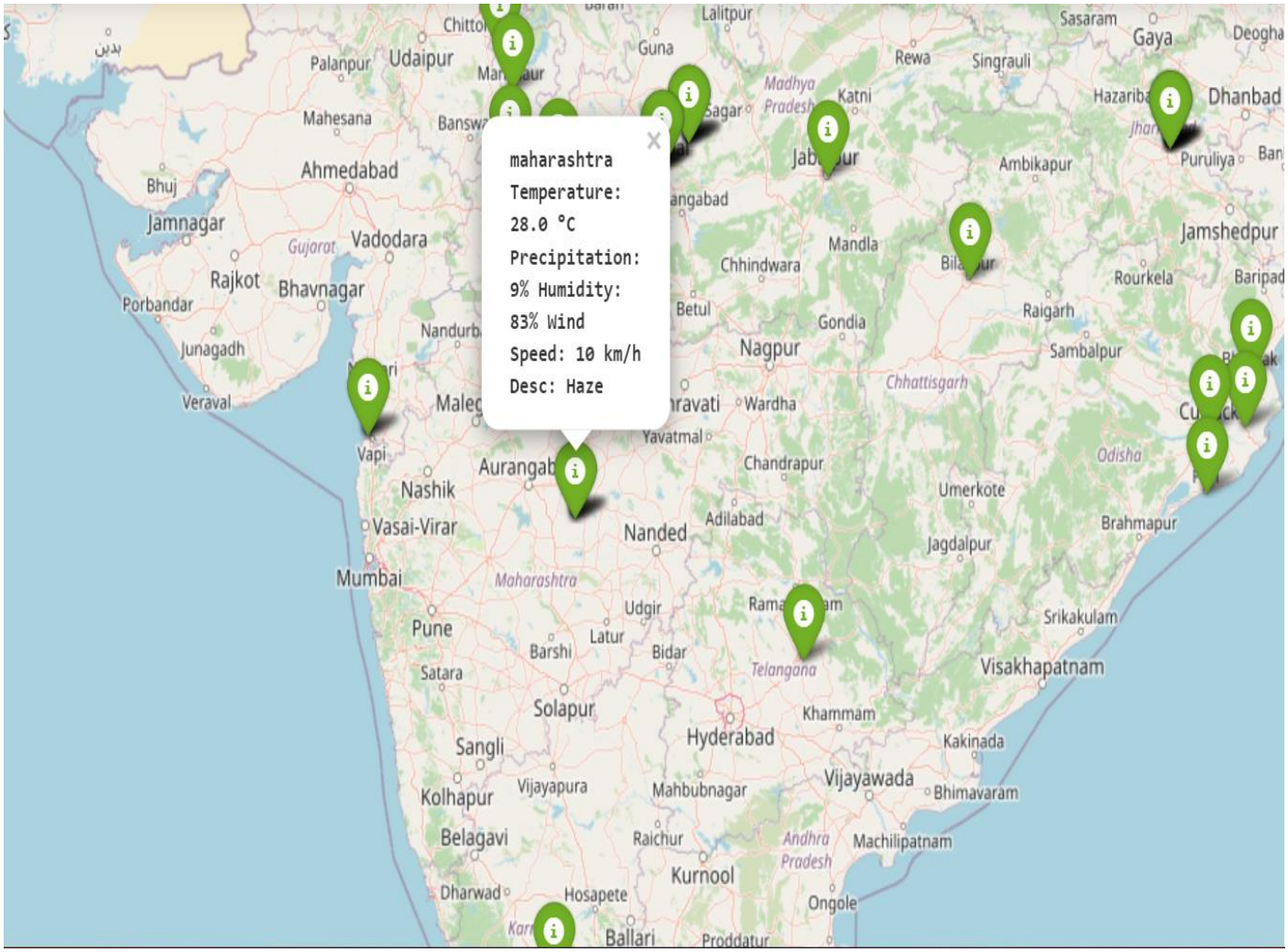


Fig 5.8 Map with flooded regions

6. CONCLUSION AND FUTURE SCOPE

The flooded regions are obtained in this method. This process can be repeated at specific time intervals, and data collected at each time is stored as historical data to create a GIS database. This historical data can be analyzed and studied to form patterns with which future floods can be forecasted.

A flood map can be incorporated with a google map at the time of the event so that google Maps' algorithm works based on it. Routes that are flooded can be identified with satellite images and displayed on google Maps.

Flood-prone regions can be identified by understanding historical data, and Government can perform mitigation works in those regions.

This system's end users are government agencies, NGOs, and responsible authorities. At the time of a disaster event, this system can be used to obtain the location data, and response action can be taken faster and help the public get out of complex situations.

A user-friendly system can be created that displays messages from time to time and severely affected locations requiring immediate actions and people needing help during a disaster.

7. REFERENCES

[1] Identifying and Analyzing Disaster-Related Tweet, Through Hashtag Monitoring Using Data Mining and NLP Techniques, IEEE, S. M. DedarAlam, 2021.

[2]A Framework for Massive Twitter Data Extraction and Analysis, Alvaro Cuesta, David F. Barrero and Marıa D. R-Moreno, 2014.

[3] The Application of NLTK Library for Python Natural Language Processing in Corpus Research, Meng WangJining Medical University, FanghuiHuJining Medical University,2021.

[4]https://towardsdatascience.com/data-101s-spatial-visualizations-and-analysis-in-python-with-folium-39730da2adf

[5]Detecting Disaster Related Tweets Using Hybrid Deep Neural Network Models, IEEE, Nayan Ranjan Paul; Manmohan Sahoo; Sukanta Kumar Hati; Tapaswini Sahoo, 2021

[6] Detection of informative tweets in crisis events Anna Kruspe, Jens Kersten, and Friederike Klan, German Aerospace Center (DLR), Jena, Germany,

[7] Event classification and location prediction from tweets during disasters, Jyoti Prakash Singh, Yogesh K. Dwivedi, Nripendra P. Rana, Abhinav Kumar, Kawaljeet Kaur Kapoor, Ann Oper Res 2017.

[8] https://www.youtube.com/watch?v=Lu1nskBkPJU

[9] https://www.youtube.com/watch?v=Sp3N5B1UqAE

[10]https://towardsdatascience.com/how-to-extract-data-from-the-twitter-api-using-python-b6fbd7129a33

[11]https://www.youtube.com/watch?v=g4rYdGkBw8&list=PLe7QnasSKJshukb8hv2H9MX0utQOPZN2P&index=31

[12]https://towardsdatascience.com/preprocessing-text-data-for-machine-learning-6b98f7bb0258

[13] https://towardsdatascience.com/using-mapquest-api-to-get-geo-data-a0b4375145e3

[14] https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a

[15]https://towardsdatascience.com/the-best-way-to-manage-unstructured-data-efficiently-b54dda2c24