IDENTIFYING TRAFFIC CONFLICT POINT USING GIS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

GEOINFORMATICS

By

ASHISH GUPTA

(2K20/GEO/04)

Under the supervision of

Dr. RAJAN YADAV, PROFESSOR

MULTIDISCIPLINARY CENTRE FOR GEOINFORMATICS
DEPARTMENT OF CIVIL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi – 110042

May 2022

MULTIDISCIPLINARY CENTRE FOR GEOINFORMATICS
DEPARTMENT OF CIVIL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Bawana Road, Delhi – 110042

CANDIDATE'S DECLARATION

I, Ashish Gupta, Roll No. - 2K20/GEO/04, Student of M.Tech (Geoinformatics Engineering), hereby declare that the project Dissertation titled "*Identifying traffic conflict point using GIS"* which is submitted by me to the Multidisciplinary Centre for Geoinformatics, Department of Civil Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previouslyformed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi ASHISH GUPTA

Date: 30th May 2022

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Rajan Yadav for guiding me throughout this research work, and for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me a lot during this research and writing of this thesis. I feel motivated and encouraged every time I attended his meeting. Without his encouragement and guidance this project would not have materialized.

It gives me great pleasure in acknowledging the support and help of all the members of Multidisciplinary Centre of Geoinformatics Lab at Delhi Technological University who were always there to advice and guide me throughout my M. Tech journey. I am thankful to all the staffs and members of Department of Civil Engineering, our HOD, Department of Civil Engineering and the entire fraternity of Delhi Technological University for giving me the opportunity to study at this institute.

I am thankful to all my classmates and friends who made my stay in this institute, an unforgettable and rewarding experience. Finally, I feel great reverence for all my family members and the Almighty, for their blessings and for being a constant source of encouragement.

ABSTRACT

According to the NCRB report, this has been found that 328 persons lose their lives every day on average, despite the COVID-19 lockdown and overspeeding was found to be the cause of more than 60% of road accidents. This leads to the emergent need of creating a life-saving phenomenon for a safe environment for society as loss of life is a huge problem around the globe and can impact emotional balance as well as the economic and harmonic loss for the country. This study focuses on developing an Application Programming Interface (API) system API that may notify users when they are about to enter an accident-prone zone.

We have developed a code using Python programming language. Afterwards, we have used four distance formulae to translate the difference in latitude and longitude between two sites into the distance as an input which can be compared to any threshold. It was one kilometre in this case. Furthermore, we constructed buffer zones and major crossings, and junctions for visualizing the current scenarios and compared the data obtained from users via a circulating Google form with the standard data accessible from the Delhi government's report.

Findings show the successful testing of the code which gives out output messages such as ‘You are probably Safe’ and ‘conflict zone is ahead’ to differentiate between the safe and conflict zone. The future scope of the study can be combined to study to define the high and low-risk zones based on the number of accidents published in annual government reports. This can also be further developed into a full-fledged mobile live application using different web mapping platforms.

TABLE OF CONTENTS

1. INTRODUCTION…….....1

1.1. Background…..……..……..

1.2. Traffic Scenario in India…..…..……..……..2

1.3. Traffic Conflict in India…..…….……3

1.4. Ways to Reduce Road Traffic Congestion …….…….....4

1.5. Geographic Information System in Controlling Traffic Conflict…….…...5

1.6. Power BI for Traffic Analysis…….…….……...6

1.7. Objectives……..6

2. LITERATURE REVIEW…….……...8

3. STUDY AREA..……..11

4. METHODOLOGY…….……15

4.1. Data Collection Phase……..…15

4.2. Data Cleaning Phase…….20

4.3. Creation of Map Using ArcGIS…….....20

4.4.Geographic Analysis…….22

4.4.1 Preparation of Overlay……..…22

4.4.2. Buffer Analysis…….……..25

4.5 Visualization of Data Using Power BI Tool…….…….26

4.6. Creation of API and Programming Code …….26

4.7. Comparison between Conflict Points and Current Location……....28

5. RESULTS…….30

6. CONCLUSIONS……34

7. REFERENCES……..35

APPENDIX: SOURCE CODE…..…….39

LIST OF FIGURES

Figure 1. Traffic Congestion in India's Top 4 Cities……..…….…….2

Figure 2. The Fatal Crashes Rates in Delhi…….…….4

Figure 3. District Wise Map of Delhi Prepared……..……12

Figure 4. Flow Chart for Creating the API…….…….……15

Figure 5. Data collection Overview (a) – (e)……. …….……..…….…….19

Figure 6. Conflict Points Defined by Users…….……...…..……21

Figure 7. Conflict Points as Per Road Crash report 2021……...…..…21

Figure 8. Overlay for User Defined Traffic Conflict Points and Road Crash Report 2021....23

Figure 9. Overlay for User Defined Traffic Conflict Points and Junctions/Crossings in Delhi……24

Figure 10. Buffer for Zones of Accidents…….…..25

Figure 11. Pseudo Code for API…….29

Figure 12. Prominent Conflict Points in Delhi …….….30

Figure 13. Major Reasons of having Traffic Conflicts …….30

Figure 14. Causes of Traffic Conflicts in Delhi …….31

Figure 15. Seasonal Impact on Traffic Conflict Points ……..31

Figure 16. Vehicle Influence on Traffic Conflicts …….32

Figure 17. API Output ……..…….33

LIST OF TABLES

Table 1. District Wise Delhi Population…….….11

Table 2. Comparison of Sex Ratio Between Urban Cities of India……..…...13

CHAPTER 1

1. INTRODUCTION

1.1 Background

Any country's economic progress is dependent on its transportation network, which includes road, rail, and air connections. The first of these roads is the most important. A well-maintained road network is essential for connecting rural and urban areas. In addition, road safety is a critical consideration. It is essential to a long-term transportation development strategy. Injury and death as a result of road accidents are a negative impact of modern road transportation systems. While the situation in high-income industrialised countries is improving, the situation in most poor countries is worse. The demand for transportation services, especially road transportation, has increased as a result of ongoing socioeconomic expansion over time. As the number of vehicles on the road increases, more road disputes and traffic accidents arise [1].

The majority of these collisions are thought to be caused by human error and negligence on the part of drivers or pedestrians. The likelihood of occurrence, as well as the severity of the event, can often be decreased by the use of adequate traffic control systems and sound roadway design practices. However, the success or failure of such control devices and design criteria is heavily reliant on the examination of traffic accident records at specific places. A methodical and scientific strategy based on precise and trustworthy traffic accident data has long been acknowledged as one of the most successful techniques of accident reduction.

However, the information needed for such an analysis isn't always available. Because the majority of accident information in police records is insufficient, it may not be used to its full potential. Furthermore, records are required to inform initiatives such as enforcement, education, maintenance, vehicle inspection, emergency medical services, and engineering to enhance streets and highways [1].

1.2 Traffic Scenario in India

India, the world's second-most populous country after China, is the most densely populated country on the planet. In India, vehicles are allocated inequitably. It is commonly known that traffic moves at the speed of its slowest component. Automobiles, buses, trucks, railroads, motor vehicles, motorcycles, scooters, and bicycles are all available in most counties. However, in India, in addition to regular urban transportation, networks of auto-rickshaws and two-wheelers, as well as bullock carts and hand-pulled rickshaws, have contributed significantly to traffic congestion, with a remarkable hundred-fold growth in the population of motorised vehicles. The extension of the road network, on the other hand, has not kept pace with this growth.

While the number of cars on the road has increased from 0.3 million in 1951 to over 30 million in 2004, the road network has grown from 0.4 million km to 3.32 million km, a mere 8-fold increase in length. In most Indian metros, traffic congestion is a major issue. Public transportation systems are overburdened, and cities are limited in how much extra infrastructure they can build, such as highways and train lines. In six cities: Mumbai, Delhi, Ahmadabad, Bangalore, Chennai, and Hyderabad, consider Figure.1, which shows the effects of traffic congestion. Traffic in cities is expanding [2], four times faster than the population.

COUNTRY RANK	WORLD RANK	CITY	TIME LOST PER YEAR	CONGESTION LEVEL 2021	CHANGE FROM 2019	CHANGE FROM 2020
1	5	Mumbai	121 hours	53%	↓12%	0%
2	10	Bengaluru	110 hours	48%	↓23%	↓3%
3	11	New Delhi	110 hours	48%	↓8%	↑1%
4	21	Pune	96 hours	42%	↓17%	0%

Figure1. Traffic Congestion in India's Top 4 Cities

Congestion on India's metropolitan highways is wreaking havoc on vehicle inventories and affecting the country's urban economies in a variety of ways. Congestion is described as an overabundance of demand for travel compared to supply.

In fact, governments are being pushed to reconsider policy for urban transportation, such as in Delhi, due to rising demand for restricted services such as public transportation. Congestion on city roadways [3], prevent traffic from moving, resulting in an unbearable increase in travel time.

Constructing a new road or widening an existing road provides a short solution to traffic congestion, but in the long run, it only encourages the growth of additional vehicles by increasing the volume of traffic, and it may also discourage the use of public transportation.

Small traffic control and information strategies are necessary to combat the major problem of rising and current congestion problems in Indian cities, which can then reduce traffic congestion and increase demand for public transportation.

According to the ministry of surface transport's transport research wing, the total number of registered vehicles in India increased from over 50000000 in 2007 to over 15000000 in 2012 [3].

1.3 Traffic Conflicts in India

Road traffic accident fatalities and injuries are a major and growing public health problem in India. The National Crime Records Bureau (NCRB) disclosed [4], in its annual 'Crime India' report for 2020 that 3.92 lakh lives had been lost in three years owing to negligence linked to traffic accidents. According to the report, 1.2 lakh deaths were registered in 2020, 1.36 lakh in 2019, and 1.35 lakh in 2018. The rates of accidents in Delhi also are increasing at alarming rates.

The rates of fatalities and accidents are illustrated using Figure.2. Over speeding was found to be the cause of more than 60% of road accidents, accounting for 75,333 deaths and 2,09,736 injuries, according to government data [5].

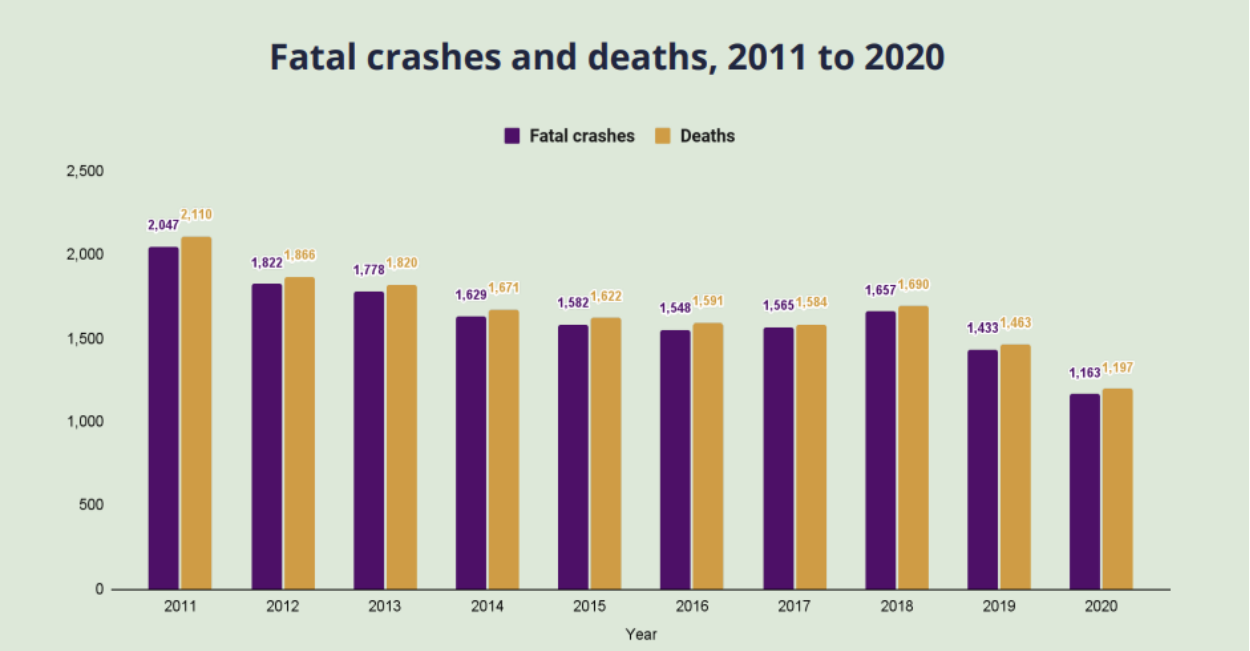


Figure2. The Fatal Crashes Rates in Delhi

According to the NCRB, two-wheeler riders made up 43.6 percent of road accident victims, followed by 13.2 percent of cars, 12.8 percent of trucks, and 3.1 percent of buses [4]. It noted that dangerous or irresponsible driving or overtaking was responsible for 24.3 percent of road accidents, which resulted in 35,219 deaths and 77,067 injuries. According to the official statistics, only 2.4 percent of traffic accidents were caused by bad weather.

1.4 Ways to Reduce Road Traffic Congestion

The congestion on road is one of the main reasons for rising traffic conflicts therefore, necessary steps are required to put this number as minimum as possible. Some of the basic methods that can be taken into account on daily basis to reduce [6], the amount of traffic on the roads listed in this section:

(a) Taking public transportation is a less expensive and less stressful option for getting from one location to another. Many people using public transit will reduce the number of cars on the road, resulting in less traffic congestion.

(b) Carpooling is an excellent way to lessen your contribution to traffic congestion. Taking turns driving to work is an excellent method to relieve the stress of daily traffic.

(c) We can contribute less to traffic congestion by avoiding peak hours and scheduling errands in the afternoons or evenings after the peak hours [6]. The amount of traffic on the roads could be greatly reduced if the existing traffic regulations were strictly enforced.

(d) Stricter enforcement of some existing traffic restrictions could significantly reduce the quantity of traffic on the roadways [5].

(e) Advanced traffic advisory is a system that can be used to monitor traffic on the roadways in real time. This uses a variety of subsystems on the road, such as CCTV cameras, sensors, and other devices, to deliver traffic advice 24 hours a day, seven days a week [6].

(f) Based on real-time traffic data, adaptive signal technology [6], can be used to change the amount of time for a green light in a specific signal.

(g) Variable lane capacity management is one example of a realistic, innovative technology that is needed. This system aids in the improvement of roadway efficiency [6]**.**

1.5 Geographic Information Systems in Controlling Traffic Conflicts:

The number of studies on techniques for assessing accidents and road design has risen dramatically in recent years. Geographical Information Systems (GIS) stand out among these tools for their capacity to undertake complicated geographical analysis [7]. However, the GIS has been used as a geographical database to store and represent data on accidents and road characteristics in some cases. It has also been used to illustrate the results of statistical studies of accidents, however, these statistical studies were not conducted with the use of the software.

A Geographic Information System (GIS) is a spatial system that creates, manages, analyzes, and maps all types of data. Many traffic agencies employ GIS technology since it is a popular tool for visualising accident data and analysing hot spots [8]. Understanding geographical and temporal crash patterns allow safety experts to identify areas with a higher number of collisions and compare them to other similar sites. These areas are known as hotspots [9].

1.6 Power BI for Traffic Analysis:

Power BI is an effective tool to visualize real-time data by connecting it to the warehouse or any data source like a spreadsheet, cloud, etc. which allows us to further connect it with automated services like power automate [10]. In short, by using power BI services, we can schedule a periodic refresh of the dataset and can also create interactive visuals which provide more insights to the end-user.

Before we use the dataset for producing insights, we require preprocessing. Preprocessing data is a data mining technique for transforming raw data into a usable and efficient forma. Raw data is frequently incomplete and formatted inconsistently. The success of every project that involves data analysis is directly proportional to the quality of data preparation. Data validation and imputation are both parts of preprocessing [10].

The purpose of data validation is to determine whether the data is comprehensive and accurate. The purpose of data imputation is to rectify errors and fill in blanks, either manually or automatically using Business Process Automation (BPA) programming [10]. Data goes through a series of steps during preprocessing:

I. Data Cleaning:

The process of repairing and eliminating erroneous records from a database or table is known as data cleaning or cleansing. Data cleansing entails filling in missing values or deleting rows with missing data, smoothing noisy data, and correcting data discrepancies [11]. All sets of data must be reliable and devoid of any flaws that could cause problems throughout subsequent use or analysis after thorough purification.

II. Data Integration:

Data from various formats is combined, and data conflicts are handled. Data integration is the process of combining data from various sources into a single dataset with the goal of providing users with consistent data access and delivery across a wide range of subjects and structure types, as well as meeting the information requirements of all business and operational processes [11]. The data integration is one of the most important parts of the total data management process, and it's becoming more common as big data integration and the need to share existing data become more important. In this research, data from various formats is combined, and data conflicts are handled.

III. Data Transformation:

Transformation of data includes standardization and Normalization of data in which it is ensured that no data is duplicated, that it is all saved in one location, and that any dependencies are logical [12]. This helps in transforming the data in one single format making the analysis easier and simpler.

IV. Data Reduction:

Databases can become slower, more expensive to access, and more difficult to store when the volume of data is large. In a data warehouse, data reduction seeks to give a simplified version of the data. Data reduction is a method of reducing the size of original data so that it can be represented in a much smaller size [11], [12]. By preserving the integrity of the original data, data reduction techniques are utilised to generate a reduced version of the dataset that is substantially smaller in volume. The efficiency of the data mining process is improved by reducing the data, which delivers the same analytical conclusions [13].

V. Data Discretization:

Data discretization is a technique for transforming a large number of data values into smaller ones, making data interpretation and management easier. To put it another way, data discretization is a technique for turning continuous data's attribute values into a finite collection of intervals with little data loss [14]. This phase involves dividing the range of attribute intervals to reduce the number of values of a continuous attribute. Data discretization can be classified into two types: supervised discretization and unsupervised discretization.

VI. Data Sampling:

A dataset may be too large or complex to deal with due to time, storage, or memory restrictions. Only a subset of the dataset can be selected and worked with using sampling techniques, as long as it has roughly the same properties as the original [15].

1.7. OBJECTIVES

The objective of the study are :

- To create a working Application Programming Interface (API) system by developing software code that determines the threshold for fetching the user's current position and comparing the predetermined conflict points to identify the safe/conflict zone.
- Comparison evaluation of the results through different distance formulae –Haversine and Vincenty.
- To establish the need for simplistic GIS software through a thorough literature review by studying different GIS models such as …. And their level of complexity and practicality.
- To evaluate the comparison between the data input actual conflict points by the user given input and by Road crash report,2021 published by Delhi Government and
- Establishing the Road network density estimation using ArcGIS through the Line density estimation tool

CHAPTER 2

2. LITERATURE REVIEW

An imaging system for road traffic conflict analysis is proposed by [16]. To compute traffic conflict metrics that can be analysed by professionals, the system uses geo-referenced stereo sequences and a tracking mechanism. Using the traffic conflict technique as a surrogate safety measure could be a useful tool for figuring out how a driver interacts with and modifies his or her behaviour about the vehicle, the road, traffic control devices, and the surroundings.

An enhanced deep learning model is proposed in the (ARIANNA BICHICCHI) study to analyse the intricate relationships between the road environment and driver's behaviour during the development of a graphical representation. An unsupervised Denoising Stacked Autoencoder (SDAE) capable of providing output layers in RGB hues is used in the proposed model. The data originates from an in-vehicle GPS device that was used to track kinematic measurements during an experimental driving test. The graphical results show the method's capacity to detect patterns of simple driving actions, as well as the complexity of the road environment and some events experienced along the way.

Due to multiple internal conflicts and various types of traffic accidents, the traffic situation at an urban crossroads is problematic. The automotive industry is going toward intelligent vehiclesvin order to improve road safety. The main difficulties are effectively recognising probable traffic problems and suggesting alternative driving techniques. The available research on traffic conflictsfrom the standpoint of intelligent vehicles is summarised in this paper. Intelligent vehicles can sense their surroundings, extract information about road conditions, and recognise impediments to avoid collisions or mitigate accidents.

The city of Dehradun, the capital of Uttaranchal in northern India, has been chosen for study by [1]. According to five years of police records, roughly 72 percent of collisions result in fatalities or serious injuries. Accidents are usually caused by cars, jeeps, and vans, and they occur most frequently between the hours of 2 and 10 p.m. According to the report, the city needs to improve its traffic management in order to reduce the number of traffic accidents.

In the study [17], They investigated the use of localization patterns and hot spot distribution with the help of temporal information to model accident hot spots. Hot spot analysis with data generation assists decision-makers in taking appropriate measures to reduce road accidents. To specify and analyse the distribution of accidents, information on accidents on the roads of Ilam Province was investigated [18]. The type of accident was included in the information (fatality, injury). According to the hot spot map, despite less traffic, the number (spatial weight) of fatalities is higher on northwest roads. This could be due to factors such as route geometry, a lack of appropriate relief, and so on [18].

According to [19], his study, Data analysis of positions and accident characteristics can greatly benefit prevention policies. His paper describes a practical integration of open-source software for the implementation of an instrument to insert, store, automatically analyse, and consult accident data.

A collaborative environment is required as various public bodies survey accident data, and a web-based solution is ideal for allowing multi-user access and data insertion. PostGIS and UMN Mapserver with PHP serverside scripting are classic solutions for this type of instrumen ; this note will discuss how to organise these elements in order to build a portal for optimal data insertion, automatic statistical analysis, and final result visualisation [19].

The road centre axes are processed to obtain homogeneous segments, which are then linked using a one-to-many spatial connection of "closest-point-to-segment" georeferenced road accident positions [20]. A set of tables presenting accident-specific information is also related to the accident spots.

An automated technique may derive certain risk and hazard indices, such as severity index, mortality index, vulnerability, and others, from this data. When datasets span numerous years, not only can this information be regarded as a thematic layer for quick interpretation, but it may also be used to studyvchange over time [20]. With this knowledge, public decision-makers may analyse the efficacy of previous policies and determine whether or not to intervene in the future.

For the observation of road traffic injuries (RTI), several sources are accessible, but each has its own set of limitations. They reported the findings of surveillance that combined healthcare data with information obtained by the municipal police in the southeastern part of Rome (630,000 people) in 2003. Using frequently obtained local data, this study illustrates the viability of an integrated RTI surveillance system. The high-risk regions found in this

study using the geographic analysis technique revealed infrastructure issues, indicating the need for quick intervention [21].

According to one research, the design and deployment of a Web Geographic Information Systems (WebGIS) for governmental comprehensive emergency preparedness, which is based on advancements such as Service-Oriented Architecture (SOA), Web Service, and others, and demonstrate how WebGIS might be used to assist decision making and manage geographic information in disaster risk management [22]. This system is based on a single geospatial architecture that uses loose coupling to successfully combine core geographical information, social economics, and demographic information with theme resource information from all industries. [23]

CHAPTER 3

3. STUDY AREA

Delhi is located at 76.96^{o} E, 28.44^{o} N, intersected by 77.40^{o}E, 28.76^{o} N. Delhi being the capital of India cover only 1484 square kilometers which consist of 559 sq. Kilometer of rural area and 685 sq. kilometers of Urban area [24]. Delhi having a population of 1.68 crore as per the Census of India 2011 [25], The world population review mentions the projected population of Delhi to be 3.2 crore as of 2022 with this much high population the density of Delhi is the highest among the world. Combining with the outskirt cities of Delhi make the NCT of Delhi. The sex ratio is an essential demographic statistic for studying population socioeconomic aspects. In 2001, the sex ratio (the proportion of females per 1000 males) was 821 [26].The district wise population is depicted using Table1.

All kinds of transportation, including aero planes, railroads, and roadways, connect Delhi to the surrounding regions and other major cities in India [24]. Delhi having Road network of around 28500 km as of march 2001 which make the road density of Delhi 19.2 km per square kilometers [25],

Table I.

District Wise Delhi Population

State / District	Population			& age of Urban
	Persons	Males	Females	Population
NCT of Delhi	13850507	944727	12905780	93.18
North West	2860869	265363	2595506	90.73
North	781525	46585	734940	94.04
North East	1768061	141547	1626514	91.99
East	1463583	18223	1445360	98.75
New Delhi	179112	--	179112	100.00
Central	646385	--	646385	100.00
West	2128908	86794	2042114	95.92
South West	1755041	225454	1529587	87.15
South	2267023	160761	2106262	91.91

Delhi is a heterogeneous culture where people have migrated from all across India. As a result, various languages are used in Delhi, but Hindi and English are the most commonly spoken languages throughout the NCT of Delhi [24]. Delhi now has one division, eleven districts, and 33 sub-divisions. The district map prepared is shown using Figure.3

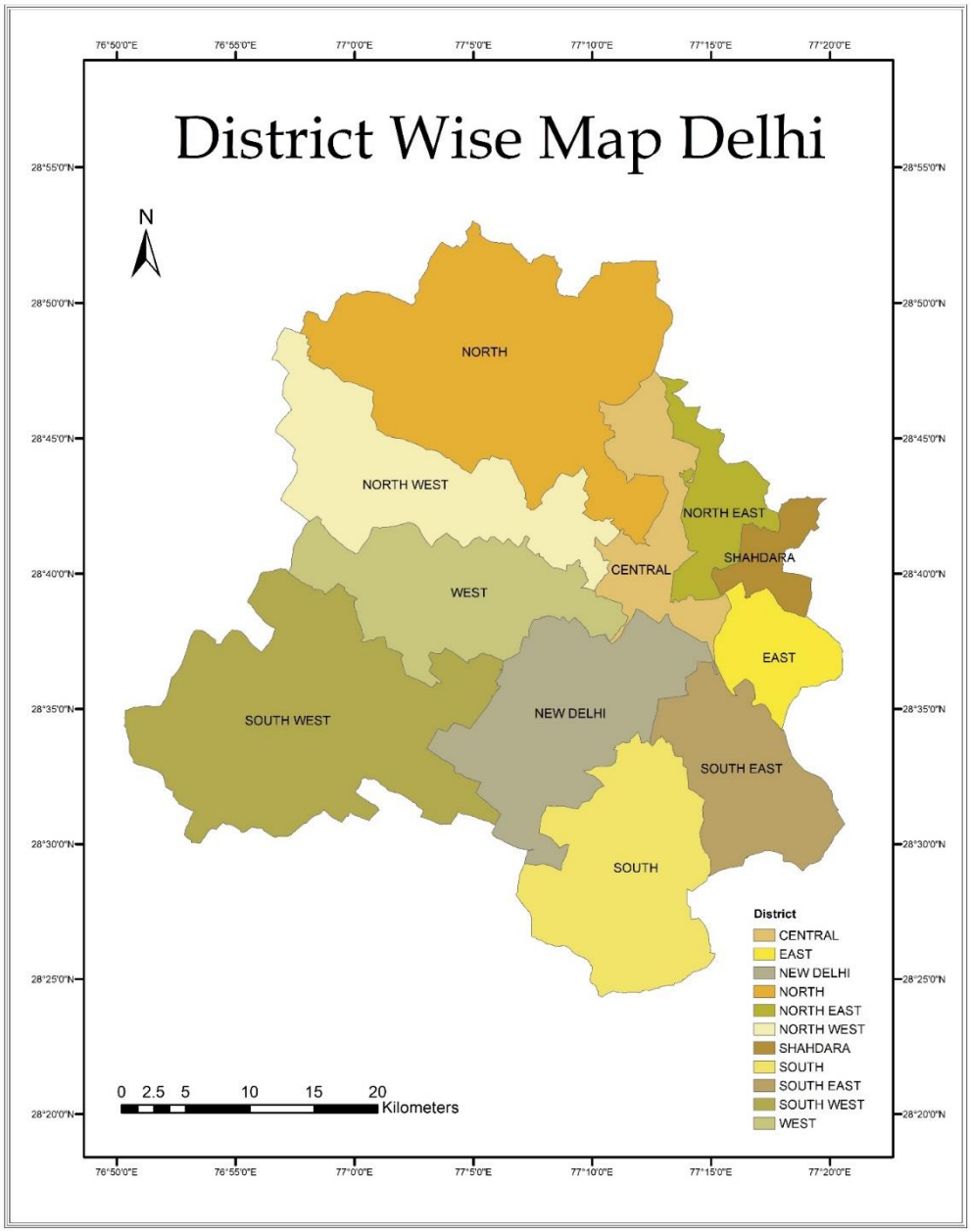


Figure3. District Wise Map of Delhi Prepared

The sex ratio of Delhi is compared with other big urban cities of India and is explained using Table.II. The sex ratio in comparison of 1991 has been increased in all the urban cities. Delhi also has shown a positive rise in sex ratio [26]. The number of females over males can be said close enough.

Table.2

Comparison of Sex Ratio between Urban Cities of India

S.No	City/Urban Agglomeration	Population			Sex Ratio	
		Persons	Males	Females	1991	2001
1	2	3	4	5	6	7
1.	Delhi Urban Agglomeration	12791458	7021896	5769562	830	822
2.	Mumbai Urban Agglomeration	16368084	8979172	7388912	828	823
3.	Kolkata Urban Agglomeration	13216546	7072114	6144432	830	869
4.	Chennai Urban Agglomeration	6424624	3294328	31302296	932	950

CHAPTER 4

4. METHODOLOGY

The methodology developed and implied to this research is explained using Figure.4, the derivation results depend on how accurately these steps have been performed.

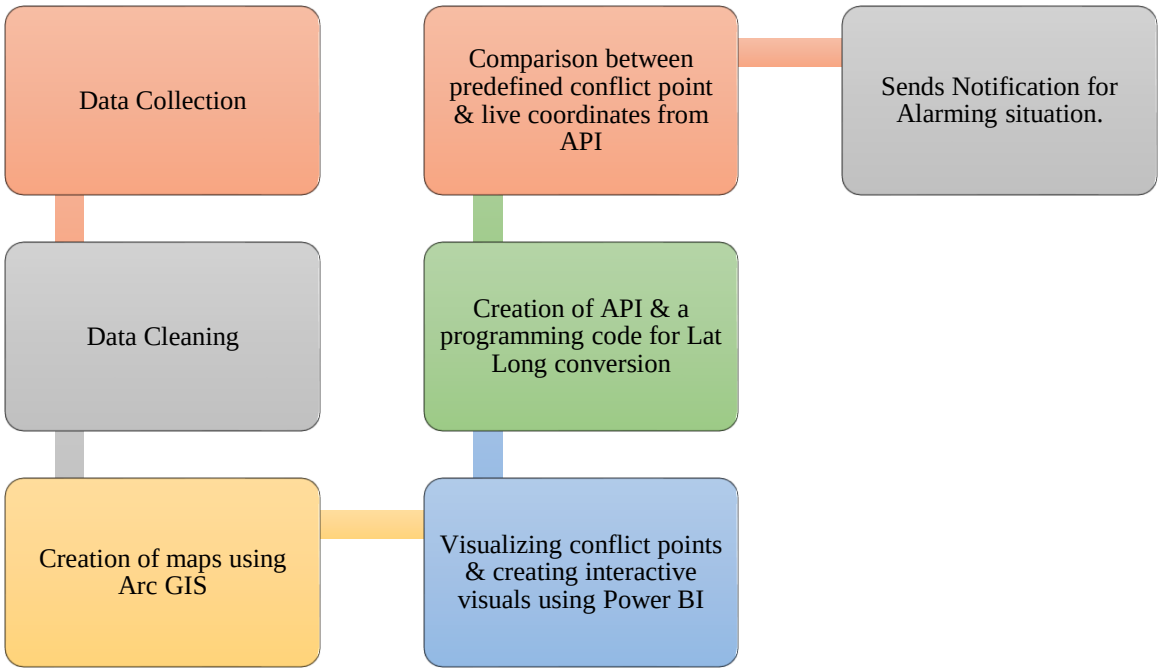
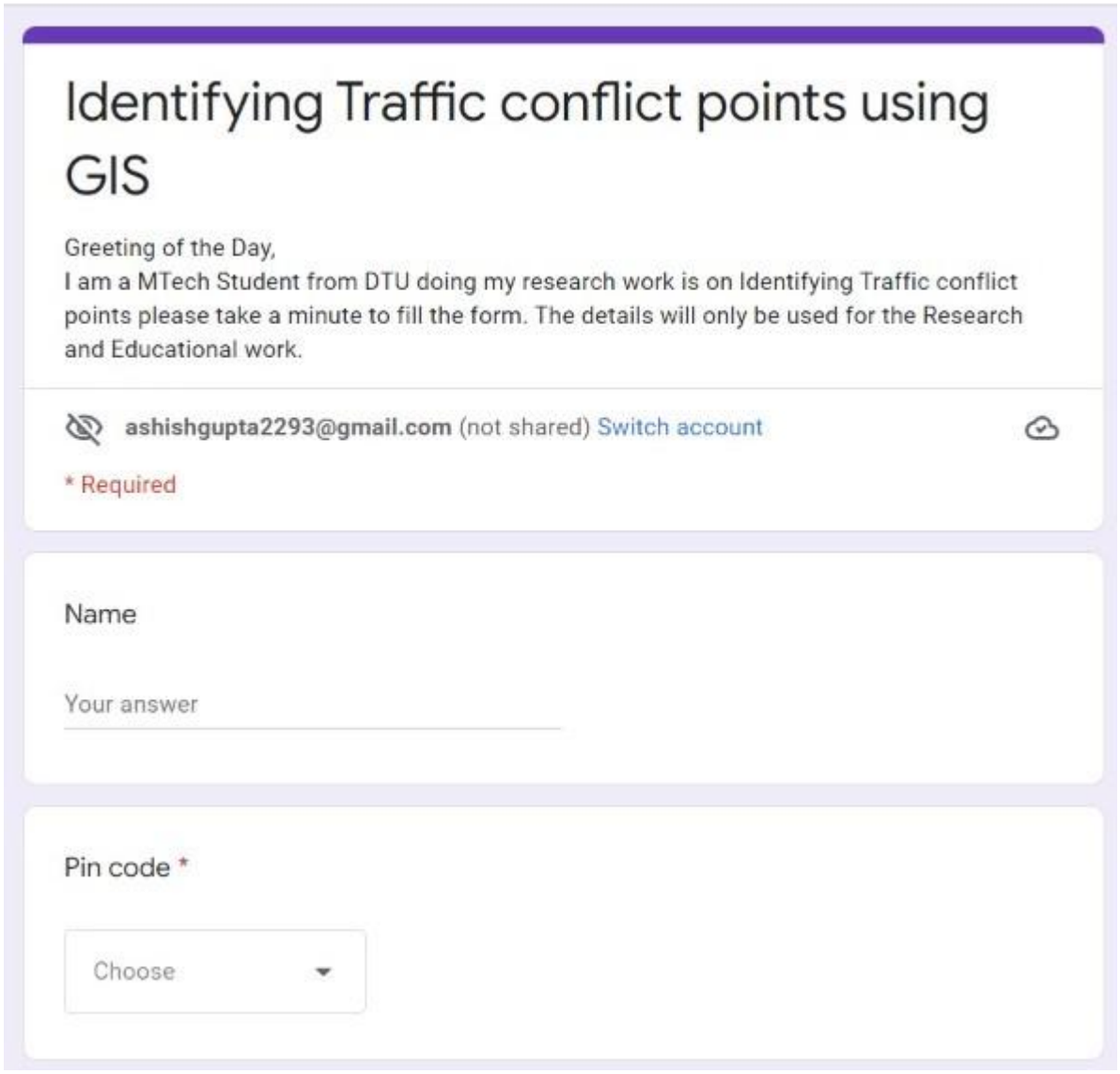


Figure4. Flow Chart for Creating the API

4.1 Data Collection Phase

Data has been created using Google Form for user perspective and major conflict points have been taken from the road crash report 2021 for Delhi which gives us some of the major conflict points present in the Delhi region [27]. However, for testing purposes, we have used Google Form Data Set which provides us Lat Long for conflict points. In addition, we have also gathered some information regarding those conflict points. The questionnaire is prepared such that no one would feel any difficulty and trouble while filling the form **[28]**. The questions circulated are illustrated using Figure.5(a), Figure.5(b), Figure.5(c), Figure.5(d) and Figure.5(e), data collected through this form is the primary source.


Identifying Traffic conflict points using GIS
Greeting of the Day,
I am a MTech Student from DTU doing my research work is on Identifying Traffic conflict points please take a minute to fill the form. The details will only be used for the Research and Educational work.
ashishgupta2293@gmail.com (not shared) Switch account
* Required
Name
Your answer
Pin code *
Choose

(a)

Q1. Which spot do you see as most prominent for conflicts? *

- [ ] Junction
- [ ] Crossing
- [ ] U-Turn
- [ ] Others

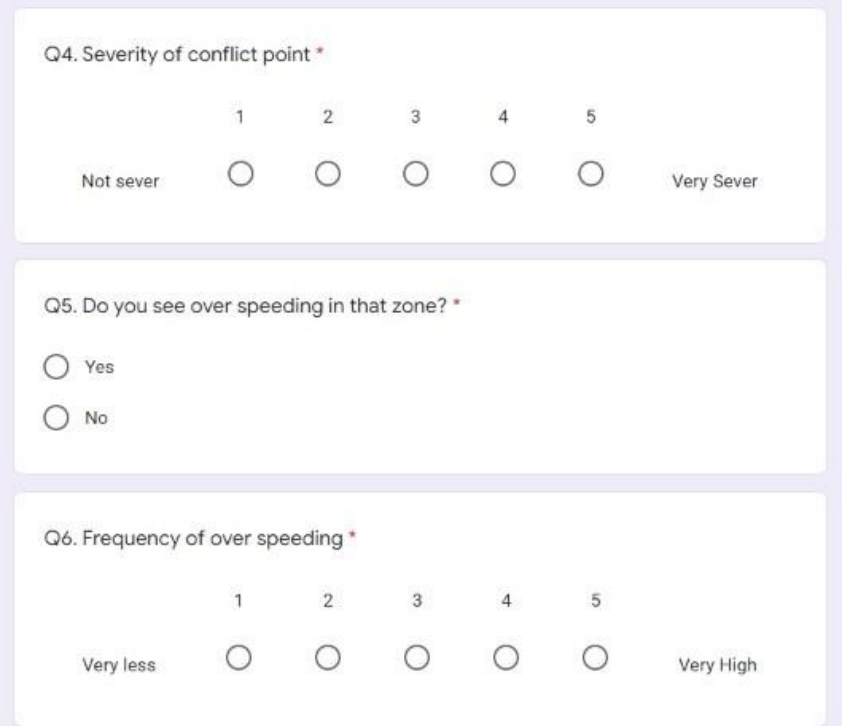
Q2. Specific location of Conflict?

Your answer

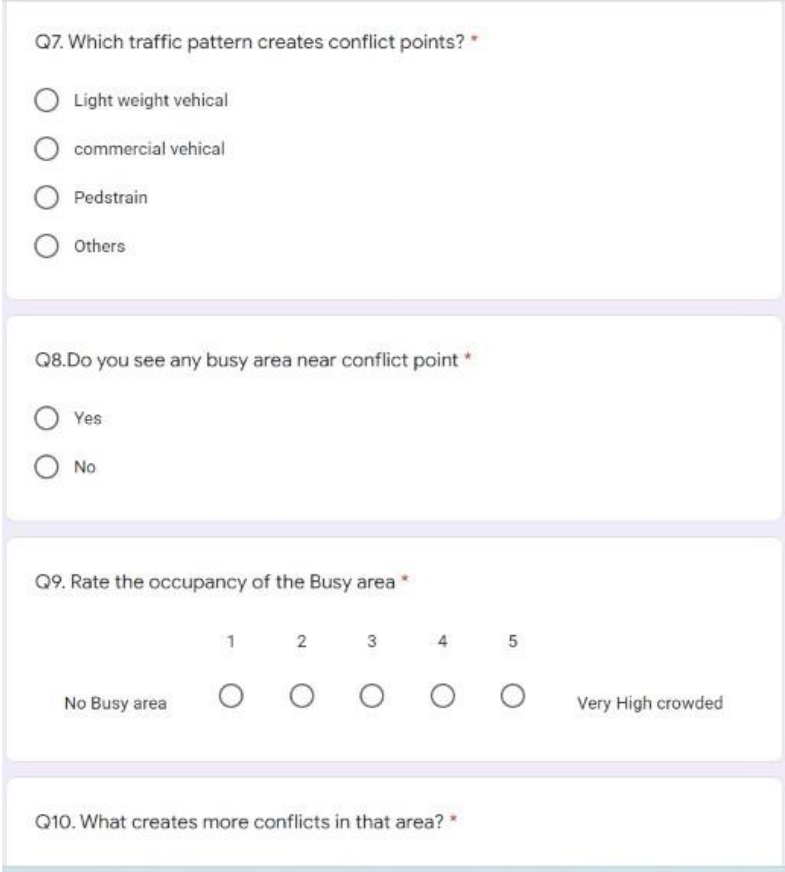
Q3. What could be the possible reason for conflict? *

- ◯ Lack of traffic signals
- ◯ Lack of traffic signs
- ◯ Poor Road condition
- ◯ Construction
- ◯ Others

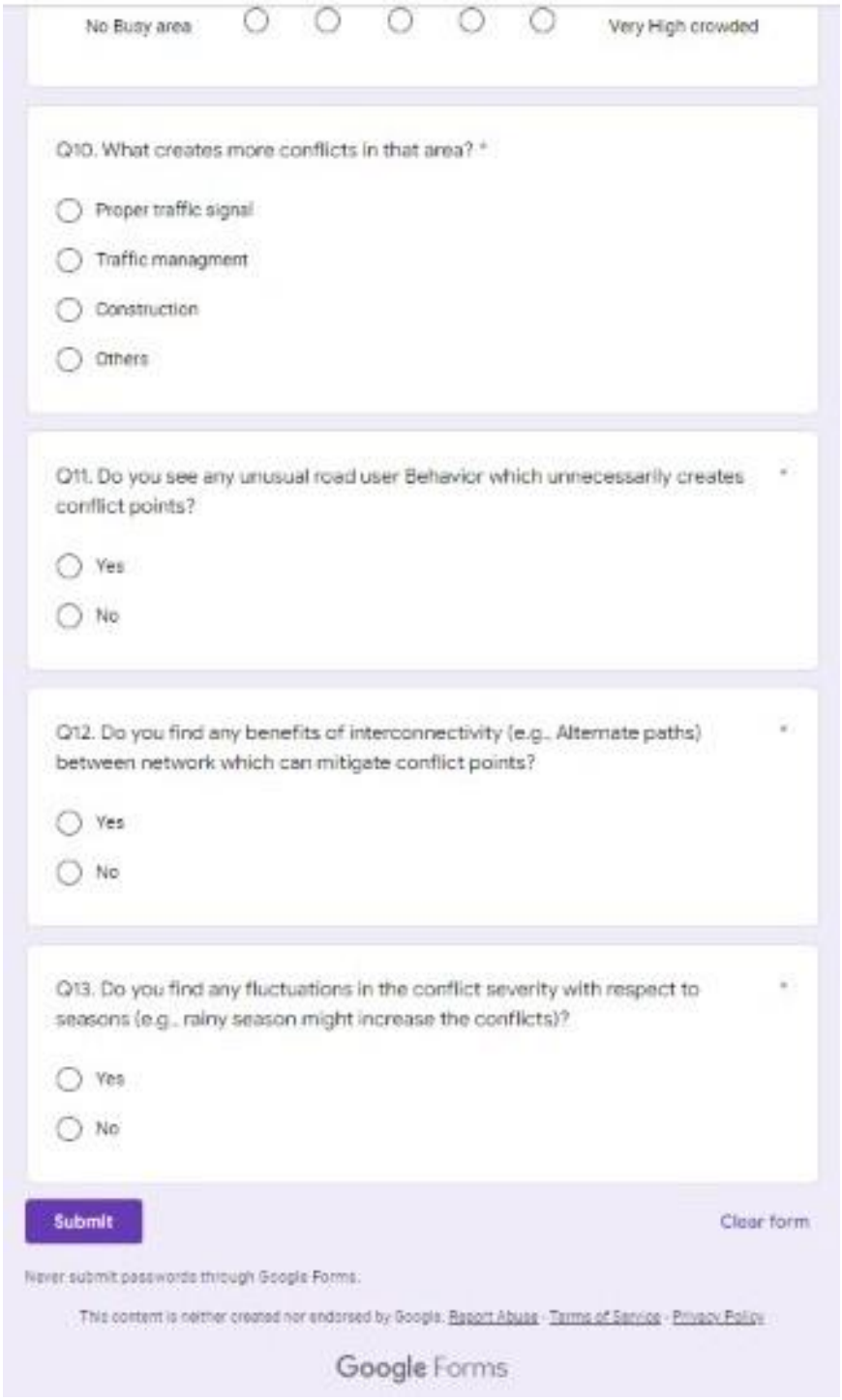
(b)


Q4. Severity of conflict point *
1 2 3 4 5
Not sever
Very Sever
Q5. Do you see over speeding in that zone? *
Yes
No
Q6. Frequency of over speeding *
1 2 3 4 5
Very less
Very High

(c)


Q7. Which traffic pattern creates conflict points? *
Light weight vehical
commercial vehical
Pedstrain
Others
Q8.Do you see any busy area near conflict point *
Yes
No
Q9. Rate the occupancy of the Busy area *
1 2 3 4 5
No Busy area
Very High crowded
Q10. What creates more conflicts in that area? *

(d)



(e)

Figure5. Data Collection (a)Overview 1 (b)Overview 2 (c) Overview 3 (d) Overview4 and (e) Overview 5

4.2 Data Cleaning Phase

The process of merging two or more data sets into a single data set is known as data merging. When you already have raw information stored in numerous files, workbooks, or data tables that you want to analyse all at once, this approach is usually required [29]. When you don't have data for particular variables or participants, you get missing data or missing values.

Data can just go missing for a variety of causes, including incorrect data entry, device failures, lost files, and so on. There will always be some missing data in any dataset. Missing values in quantitative research are shown as empty cells in your worksheet [30]. Data standardisation is the process of converting data (which might be in a variety of formats) to a standard format specified for consistency is required. standardised data follows a specific structure and set of criteria. It has been improved in terms of efficiency as well as filtering capability.

The process of structuring data from a database is known as normalisation. This comprises procedures such as generating tables and developing relationships between them according to rules aimed to safeguard data while also making the database more adaptable by removing repetition and inconsistency.

Data normalisation is a technique for arranging data such that it seems uniform across all entries and fields. When you do this, you will always get higher-quality data [29], [30]. To assure logical storage space, this step entails removing unstructured data and duplicates.

4.3 Creation of maps using Arc GIS

The location of conflict point collected as per user conflict points and conflict points collected through road crash report 2021 are plotted on Google Earth Pro [31]. These locations are then exported and mapped in ArcGIS as shapefiles. Consider Figure.6 and Figure.7, which shows the maps created using the coordinates of conflict points [32]. These maps will help us in visualizing the differences between the actual conflict points and user defined conflict points. These maps act as medium to represent the real world issues and enable us to demarcate the variations among real and theoretical world [33].

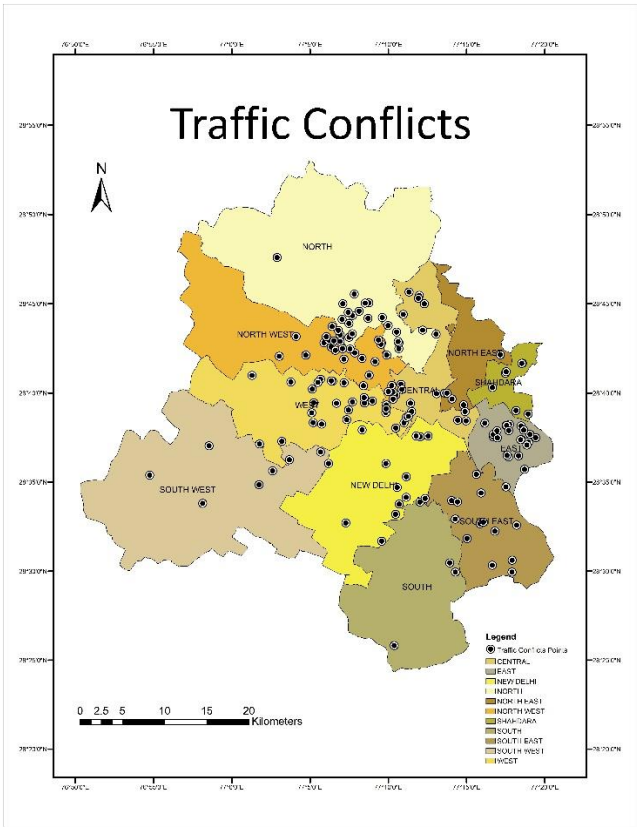


Figure6. Conflict Points Defined by User

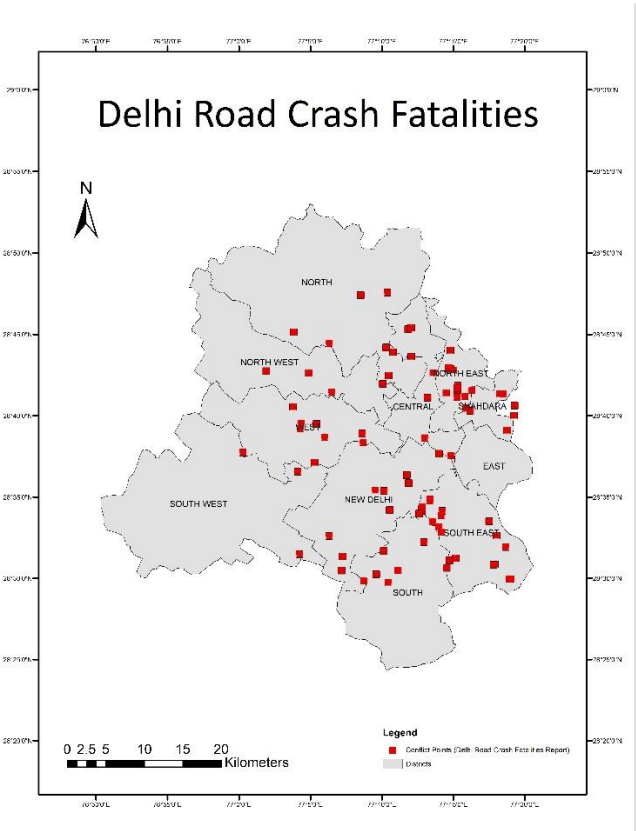


Figure7. Conflicts Points as per road crash report 2021

4.4 Geographical Analysis

Patterns, linkages, and connections are sought while analysing spatial data. It enables the analysis and interpretation of data, resulting in the discovery of meaningful patterns or processes [34]. The information can then be used to combine their findings into logical interpretations.

Some of the analysis that may be performed on spatial data include finding connections and similarities across places, recognising patterns, and making inferences from maps, graphs, diagrams, tables, and other sources. The user only needs basic statistics to check for patterns, correlations, and sequences [34], [35].

Geographic study necessitates a variety of thought processes. It's frequently difficult to distinguish between the techniques [35], for organising location data and those for evaluating it; in many circumstances, the two processes happen at the same time.

4.4.1 Preparation of Overlays

The Overlay toolset includes tools for combining, erasing, modifying, or updating spatial features across multiple feature classes, leading to a new feature layer [36]. When one feature set is superimposed upon other, new information is generated. To detect spatial links between the input features, all overlay processes require combining two feature sets into a single set of features.

In this study, we have prepared two overlays using ArcGIS software. One overlay is developed for user defined traffic conflict points collected through google form and road crash report 2021. The map is explained using Figure.8. The second overlay is created using user defined traffic conflict points and junctions crossings present in Delhi. Consider Figure.9, which shows the overlay layer [37].

These overlays will help us to better understand the causes of traffic conflicts in the region. It is generally seen that as the number of crossings and junctions increases in an area, chances of traffic conflict also rise [36], [37]. This overlay will allow us identify those critical sites so that we can work on their remedial measures.

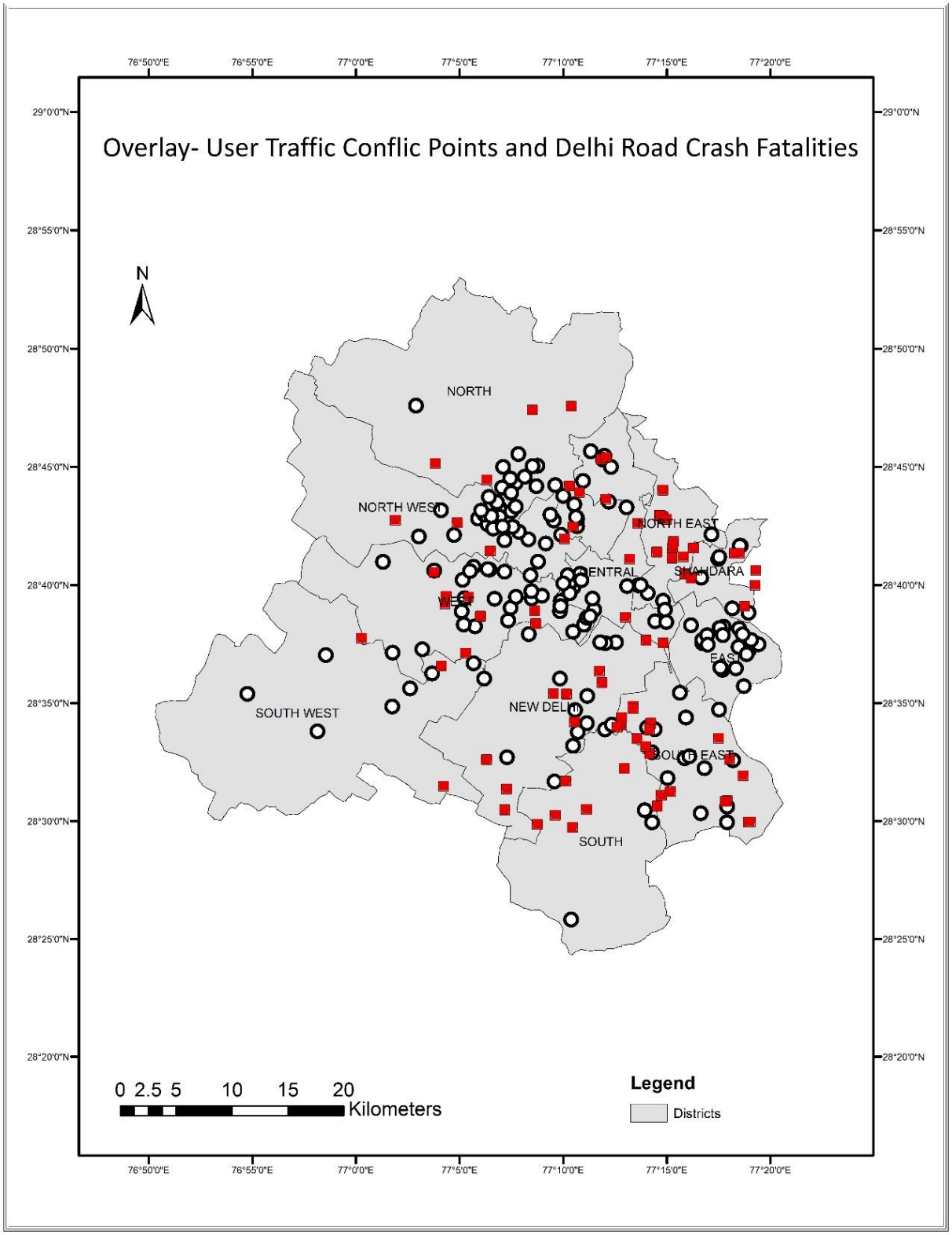


Figure8. Overlay for user defined traffic conflict points and road crash report 2021

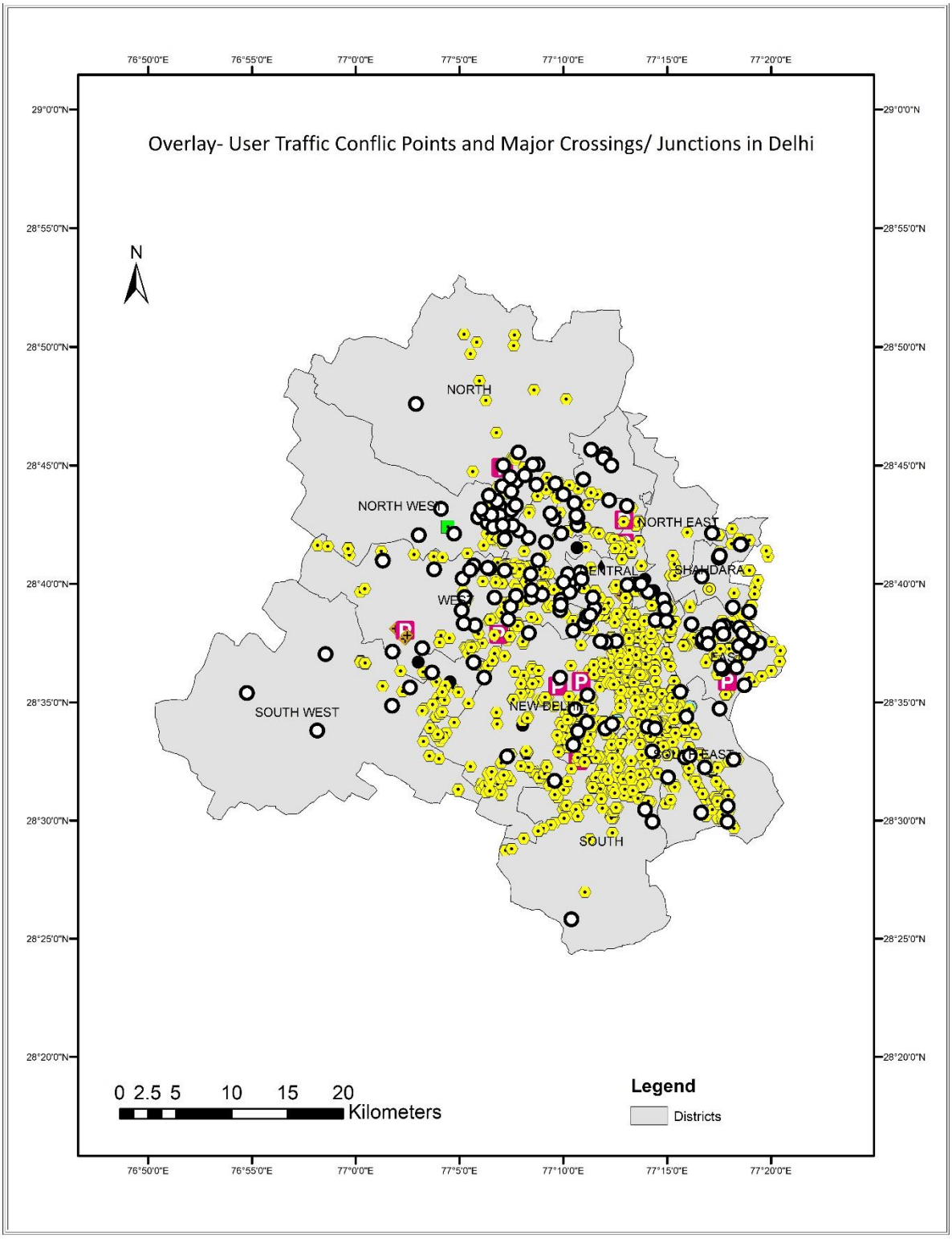


Figure9. Overlay for user defined traffic conflict points and Junctions/Crossings in Delhi

4.4.2 Buffer Analysis

One of the most essential functions of spatial analysis in GIS is buffer analysis [38]. The main concept is to build a zonal area of a particular distance outside its boundary, referred to as buffer zones, and to determine the effect range and service range on the surrounding environment [39].

The buffer analysis is done on the points identified from road crash report 2021 [40]. The user defined conflict points have been overlayed on the buffers to segregate the of zones of accidents [41]. A map of the analysis have been prepared illustrated using Figure.10.

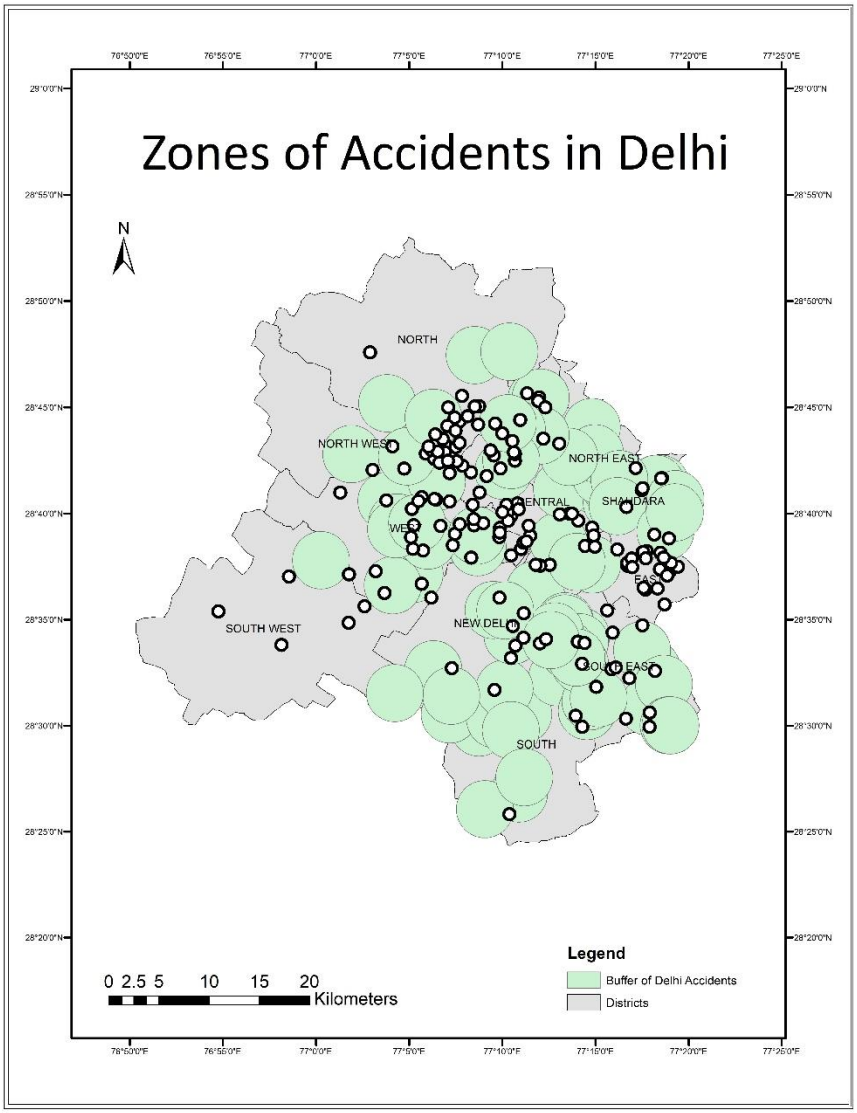


Figure10. Buffer for Zones of Accidents

4.5 Visualization of data using Power BI tool

Power BI is a very important tool nowadays for creating interactive visuals, it gives immense freedom to import data from different sources like Clouds, excel sheets, Data warehouses and many more [42]. As it is a tool from Microsoft its integration with different Microsoft enabled services makes it more valuable. For Instance, SharePoint and power.

Automate can be used for automating many facilities in power BI which in turn leads to creating an environment which provides more edge to the end-user. In this study, we have imported the CSV file which is extracted from the Google form circulated over people and after importing the dataset we have tried to create interactive visuals on the very same page so that any filter slicer can give good insights i.e., meaningful insights from the data [10], [42].

4.6. Creation of API and programming code for Lat Long Conversion

In the creation of the API of this study we have used various libraries like Geocoder, Flask, math, geopy, apscheduler, CSV, and os. Geocoder is the library which is written in Python and provides a facility for translating locations into geographic locations which are known as geocoding [43]. You'll be converting direct addresses into latitude/longitude combinations as a result of this. Flask is a development tool and a Python module which thus makes it simple to create web applications [43].

It has a simple and extensible core: it's a microframework without an ORM (Object Relational Manager) or other things like that. It has a lot of amazing features, such as URL routing and a template engine. It's a web application framework that uses the WSGI protocol. This is the backbone of any application development and is being used for API [44].

Python has a math module that can handle similar calculations. It covers very simple calculations such as addition, subtraction, multiplication, and division, as well as all advanced and complex operations such as trigonometric, logarithmic, and exponential functions.

This module has given us an edge to solve mathematical relations in calculating the distance between two coordinates [43], [44]. Using third-party geocoders and other data sources, geopy makes [45], it simple for Python developers i.e., it has allowed us to locate the

coordinates and to get the coordinates of addresses, cities, nations, and landmarks all around the Delhi in our case but it can be used for any location in the world.

The Advanced Python Scheduler (APScheduler) is a Python package that allows you to schedule your Python code to run later, either once or regularly. You have complete control over adding new tasks and removing existing ones and allow us to refresh our Coordinates every 5 seconds.

The abbreviation CSV refers to comma separated values. This file format is often used when exporting and importing data to and from spreadsheets and database tables. All things related to CSV files are being parsed, read and written by using the CSV library [46]. In Python, the OS module has methods for dealing with the operating system. Python's basic utility modules include OS. This module allows you to use operating system-dependent functions on the go.

Many functions to interface with the file system are included in this module. For Instance, our CSV file is in downloads so to fetch that file from that particular path we entail this module of the Python [46]. After Importing all these libraries now main concept begins, we have used four distance calculation formulas in our study.

(a) Euclidean distance

The length of a segment connecting two locations in either the plane or 3-dimensional space is measured by the Euclidean distance between them [47]. It is the most straightforward method of expressing the distance between two places. This formula says Distance between any two points in a space with coordinates s (x1 , y1) and (x2 , y2) is distance = $\sqrt{[(x2 – x1)2 + (y2 – y1)2]}$.

(b) Manhattan distance:

The total of the actual differences between different vectors is used to determine the Manhattan distance. The L1 vector norm, as well as the sum absolute error and mean absolute error metrics, are all connected to the Manhattan distance [48]. The Manhattan Distance between any two points in a space with coordinates (X1, Y1) and (X2, Y2) is given by distance $|X1 – X2| + |Y1 – Y2|$.

(c) Haversine distance:

The angular difference between adjacent places on the celestial sphere is known as the Haversine (or big circle) distance [49]. The latitude is considered to be the very first attribute of each location, followed by the longitude in radians [50].

$$a = \sin^2(\Delta\varphi/2) + \cos \varphi1 \cdot \cos \varphi2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

$$\text{distance} = R \cdot c$$

where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km);

(d) Vincenty distance:

Vincenty's formulas are two iterative geodesic techniques for estimating the difference in distances between two locations on a spheroid's surface [51].

Above all these four formulas, Euclidean and Manhattan are not significant as they do not entertain the values in the Radians, they expect values in terms of distances. However, we can compare the difference between the two coordinates with the threshold value, which is again in the form of Radians or degrees. In addition, these two formulas are not giving distances with minimum error so the results are not useful [52].

Haversine formula considers the curvature of the earth but due to very complex curvature it gives an error but within permissible limit and hence results have been shown in this study. Last but not least, the Vincenty formula which is again a complex one has been used for the calculation, using python dedicated library which further reduces [52], the error in terms of calculation.

After this step, the CSV file which has been exported from the google form is used for the comparison of conflict points present in the file and the current location. The data from this file has been compared automatically and the result that came across will be shown through various commands.

5.7. Comparison between Conflict points and the current location:

Comparison has been made for creating alerting mode. In this study, we have used a class and created an object for handling different datasets for various locations if somebody wants to use the same code without editing it, again and again, the object creation came into

the picture and the class is the collection of functional methods which has been applied for triggering alerting situation using the same API parallelly for all objects or anyone at a time.

I. Sending Notification for alarming Situation

While this algo runs calls multiple services and does a comparison, the notification will be sent when the current location will get compared with a list of points and if the difference is less than the threshold value which is in our case is 1 km. It notifies the user with the location name and coordinates of the location that the accident-prone area is ahead alert.

The code implemented for building the API is shown using Figure.11, in a pseudo format.

Algorithm 1: Checking User is in Conflict Zone or not

Input: C_p
/* C_p Array of pairs $(lat, long)$ latitude and longitudes of conflicting zones */
Output: User is in conflicting zone or not
Initialization: $conflictFlag = 0, userLocation = (lat, long), threshold = 3, HaverSine(), Vincenty()$;
/* $userLocation$ $(lat, long)$ of user calculated from GeoApi */
/* For distance measurement Hiversine and Vincenty distance is used */
for *each conflict zone* C_i *location in* $conflictLocation$ **do**
    Calculate distance D_i from $Haversine(userLocation, C_i)$;
    if *distance* $D_i < threshold$ **then**
        print distance D_i, address with Alert;
        $conflictFlag = 1$;
    end
    Calculate distance D_i from $Vincenty(userLocation, C_i)$;
    if *distance* $D_i < threshold$ **then**
        print distance D_i, address with Alert;
        $conflictFlag = 1$;
    end
end
if $conflictFlag == 0$ **then**
    print "You are in safe Zone"
end

Figure11. Pseudo Code for API

CHAPTER 5

5. RESULTS

The results obtained after all the analysis involved in this study are mentioned in the section. The insights of data analysis are explained using Figure.12, Figure.13, Figure.14, Figure.15 and Figure.16.

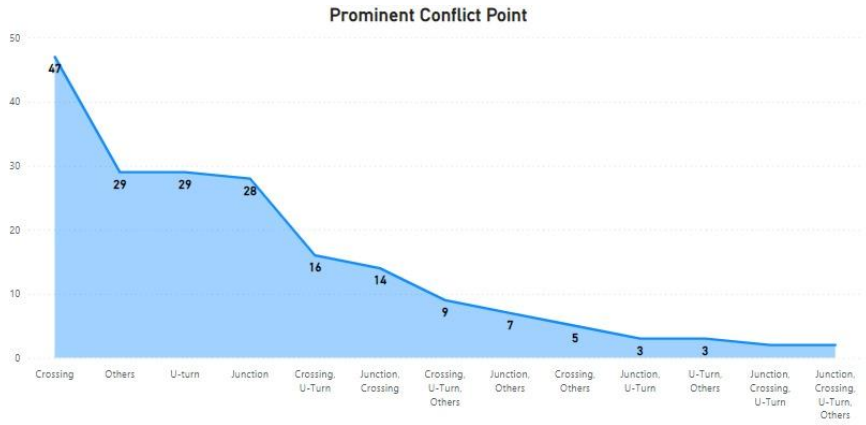


Figure12. Prominent Conflict Points in Delhi

The Prominent conflict point area chart shows that the majority of the conflict points are crossings and U-turns for the users.

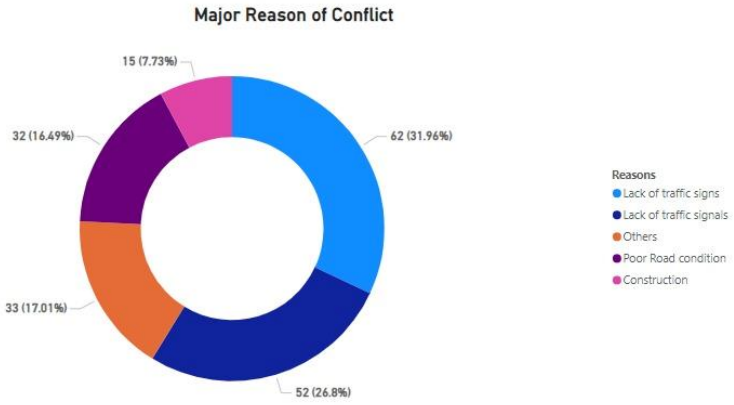


Figure13. Major Reasons of having Traffic Conflicts

Major reasons pie chart shows that lack of signals approx. 26% and signs approx. 31% are the root cause of these accidents.

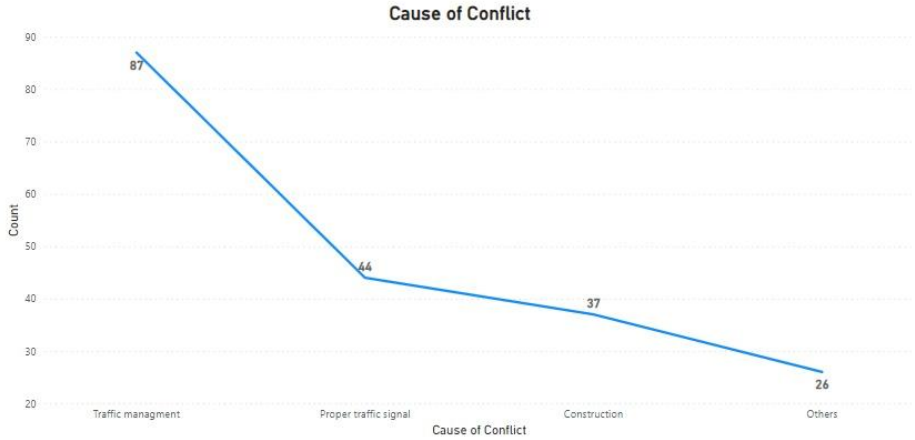


Figure14. Causes of Traffic Conflicts in Delhi

The majority of the people face traffic management as the main reason for these conflict points.

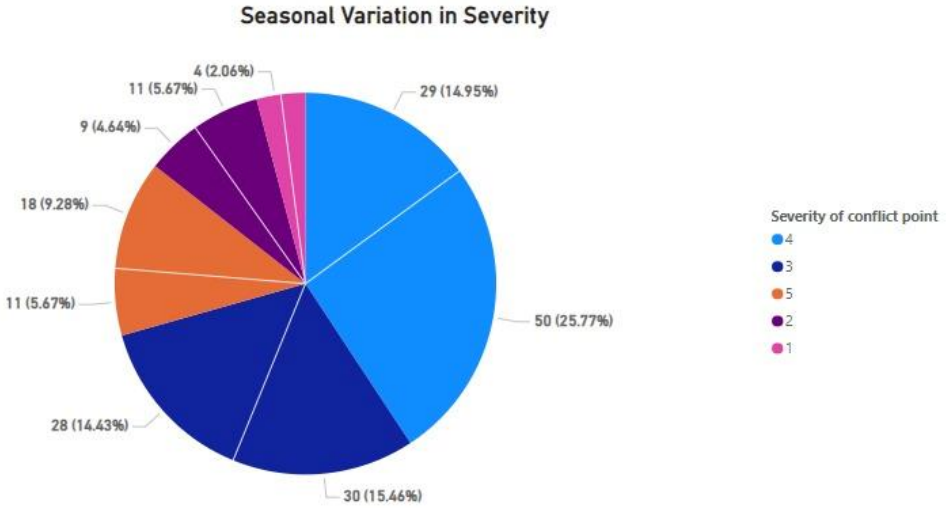


Figure15. Seasonal Impact on Traffic Conflict Points

Seasonal variation has also a very good impact on creating these conflict points. Many respondents have rated 4 on the Likert scale.

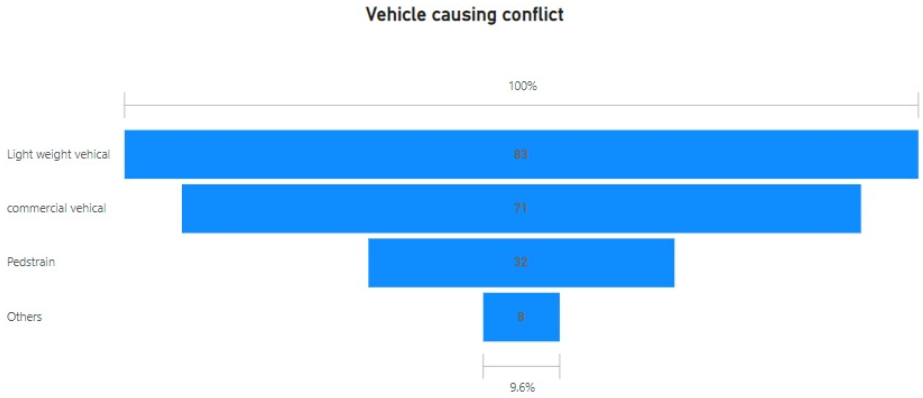


Figure16. Vehicle Influence on Traffic Conflicts

Approximately 75% of people suggest that light vehicles are more prominent for these conflict points. It can be seen in the Funnel chart presented in this report.

A mixed reaction can be seen for an overspeeding relationship with conflict points as an almost equal proportion is there on the Likert scale

The overlay analysis enlightened that the conflicts points obtained from user through Google form are nearly same as that of the points of crash mentioned in road crash report 2021 for Delhi. The overlay analysis of traffic conflicts points with the junctions/crossings of Delhi showed that the tendency of having conflicts is maximum near junctions/crossings.

The buffer analysis segregated the areas of conflicts and describes the areas having high or low number of traffic conflicts in Delhi. The user defined conflict points are found to be present in these areas.

The API developed is also giving suitable results. It is alarming a user about the nearest conflict point as soon as the user reaches within 1Km of area of conflict. The output of the API is illustrated using Figure.17.

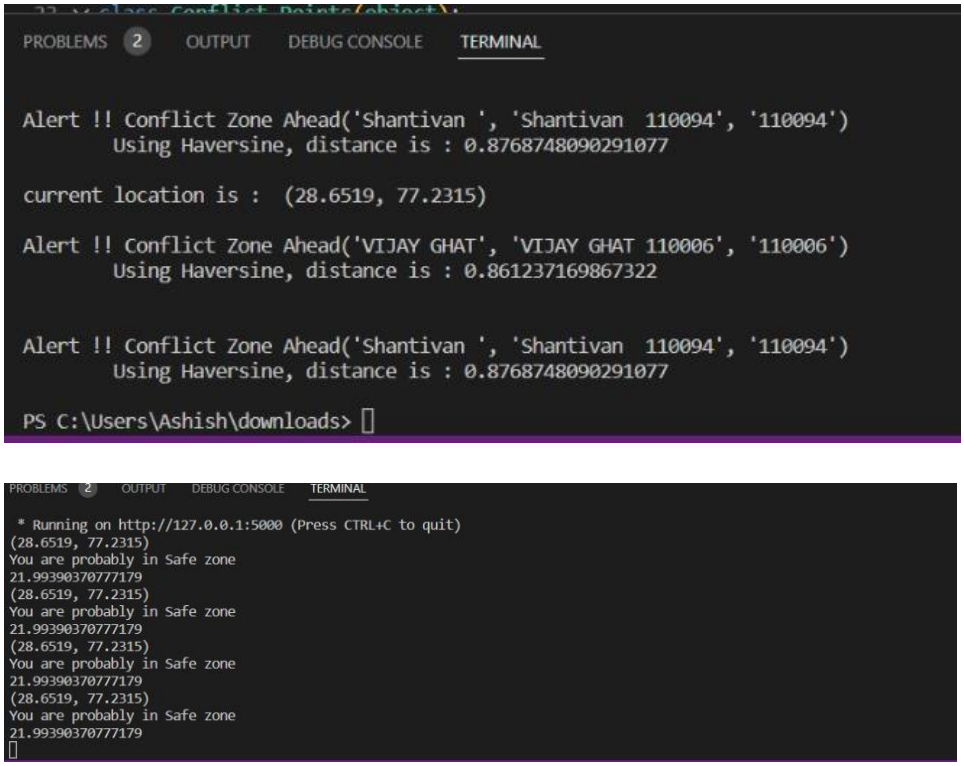


Figure17. API Output

CHAPTER 6

6. Conclusions

This study was developed with the intent to impart alertness among road users to create a safe environment. This was done through the development of an API system which imparts clear and direct output messages to users whether they are in a safe or conflict zone. The findings reveal that the code, which distinguishes between the safe and conflict zones with output signals like 'You are probably Safe' and 'conflict zone is ahead,' was successfully tested.

This application could help in bringing clarity to the minds of the users to proceed confidently on roads having various conflicts and elimination of accidents in real-life scenarios. The efficacy of the developed code lies in its simplistic approach which can further be implicated in various models and gadgets such as smartwatches, mobiles, laptops and other GPS systems. This system can bring an edge to the Indian road users which seem to be making progress towards integrating its road and highway portfolio with fast pace infrastructure. Users may benefit from this system in a variety of weather and road conditions as well.

The limitation of the study is that we are currently retrieving coordinates using an IP address that is updated every 5 seconds though we can the frequency of fetching the live coordinates, it is still lagging inaccuracy. If we can connect to a GPS, we will get a more accurate and immediate data response. The knowledge gained from this system can be enhanced further in the improvement of existing road infrastructure by mapping the conflict zones in a simulated environment form. This system can also be used to inhibit better road driving sense for future generations. Using several web mapping systems, this may be further expanded into a full-fledged mobile live application.

This may also be advanced by recording data in the form of inputs from road users and competent government authorities for the purpose of detecting the creation of potential conflict zones. Machine learning and analytics techniques may be used to capture, analyse, and understand these trends. This may be shown in real-time on a Power BI dashboard and made available to the general public on a daily basis.

REFERNCES

[1] “Ghosh, S. K., Parida, M., & Uraon, J. K. (2004). Traffic accident analysis for Dehradun city using GIS. ITPI journal, 1(3), 40-54.”.

[2] “TomTom India Traffic,” [Online]. Available: https://www.tomtom.com.

[3] “Legal Services India,” [Online]. Available: https://www.legalserviceindia.com.

[4] “National Crime Records Bureau,” [Online]. Available: https://ncrb.gov.in/.

[5] “Transport Delhi Government,” [Online]. Available: https://transport.delhi.gov.in.

[6] “Traffic Congestion,” [Online]. Available: https://www.bankbazaar.com.

[7] “Kazmi, J. H., & Zubair, S. (2014). Estimation of vehicle damage cost involved in road traffic accidents in Karachi, Pakistan: a geospatial perspective. Procedia engineering, 77, 70-78.”.

[8] “Jayan, D. K., & Ganeshkumar, B. (2010). Identification of accident hot spots: a GIS based implementation for Kannur district, Kerala. International journal of Geomatics and Geosciences, 1(1), 51.”.

[9] “Mohaymany, A. S., Shahri, M., & Mirbagheri, B. (2013). GIS-based method for detecting high-crash-risk road segments using network kernel density estimation. Geo-spatial Information Science, 16(2), 113-119.”.

[10] “Sousa, R., Miranda, R., Moreira, A., Alves, C., Lori, N., & Machado, J. (2021). Software Tools for Conducting Real-Time Information Processing and Visualization in Industry: An Up-to-Date Review. Applied Sciences, 11(11), 4800.”.

[11] “Mabry, P. L. (2011). Making sense of the data explosion: the promise of systems science. American journal of preventive medicine, 40(5), S159-S161.”.

[12] Myatt, G. J. (2007). Making sense of data: a practical guide to exploratory data analysis and data mining. John Wiley & Sons..

[13] “Masse, L. C., Fuemmeler, B. F., Anderson, C. B., Matthews, C. E., Trost, S. G., Catellier, D. J., & Treuth, M. (2005). Accelerometer data reduction: a comparison of four reduction algorithms on select outcome variables. Medicine and science in sports and”.

[14] “Jin, R., Breitbart, Y., & Muoh, C. (2009). Data discretization unification. Knowledge and Information Systems, 19(1), 1-29.”.

[15] “Hong, S. H. (1995). Scheduling algorithm of data sampling times in the integrated communication and control systems. IEEE Transactions on Control Systems Technology, 3(2), 225-230.”.

[16] “Battiato, S., Cafiso, S., Graziano, A. D., Farinella, G. M., & Giudice, O. (2013, September). Road traffic conflict analysis from geo-referenced stereo sequences. In International Conference on Image Analysis and Processing (pp. 381-390). Springer, Berlin”.

[17] “Zheng, L., Ismail, K., & Meng, X. (2014). Traffic conflict techniques for road safety analysis: open questions and some insights. Canadian journal of civil engineering, 41(7), 633-641.”.

[18] “Azami-Aghdash, S., Sadeghi-Bazargani, H., Shabaninejad, H., & Gorji, H. A. (2017). Injury epidemiology in Iran: a systematic review. Journal of injury and violence research, 9(1), 27.”.

[19] “Pirotti, F., Guarnieri, A., & Vettore, A. (2010). Road safety analysis using WEB-BASED collaborative GIS. In ISPRS) conference,(34), Part XXX.”.

[20] “Cuenca, L. G., Aliane, N., Puertas, E., & Andres, J. F. (2018, September). Traffic hotspots visualization and warning system. In 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES) (pp. 1-5). IEEE.”.

[21] “Chini, F., Farchi, S., Ciaramella, I., Antoniozzi, T., Rossi, P. G., Camilloni, L., ... & Borgia, P. (2009). Road traffic injuries in one local health unit in the Lazio region: results of a surveillance system integrating police and health data. Internati”.

[22] “Jiugang, L., Xinming, T., Zhengjun, L., & Minyan, D. (2009). Design and implementation of WebGIS for government emergency management based on SOA. International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences, 28(7), C4.”.

[23] “Kadiyali, L. R., & Venketesan, S. (1984). Traffic Accidents, forecast and remedies. Indian Highway, 12(4), 7-12.”.

[24] “Delhi Government,” [Online]. Available: https://portal.delhi.gov.in/.

[25] “Census India,” [Online]. Available: https://www.censusindia.co.in/.

[26] “Delhi Planning,” [Online]. Available: http://delhiplanning.nic.in.

[27] Sapsford, R., & Jupp, V. (Eds.). (1996). Data collection and analysis. Sage..

[28] Weller, S. C., & Romney, A. K. (1988). Systematic data collection (Vol. 10). Sage publications..

[29] “Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016, June). Data cleaning: Overview and emerging challenges. In Proceedings of the 2016 international conference on management of data (pp. 2201-2206).”.

[30] Brownlee, J. (2020). Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery..

[31] “Graham, S. R., Carlton, C., Gaede, D., & Jamison, B. (2011). Student column: The benefits of using geographic information systems as a community assessment tool. Public Health Reports, 126(2), 298-303.”.

[32] “Hamza, M. H., & Chmit, M. (2022). GIS-Based Planning and Web/3D Web GIS Applications for the Analysis and Management of MV/LV Electrical Networks (A Case Study in Tunisia). Applied Sciences, 12(5), 2554.”.

[33] “Kolejka, J. (2018). Landscape mapping using GIS and Google Earth data. Geography and Natural Resources, 39(3), 254-260.”.

[34] “Franch-Pardo, I., Napoletano, B. M., Rosete-Verges, F., & Billa, L. (2020). Spatial analysis and GIS in the study of COVID-19. A review. Science of the total environment, 739, 140033.”.

[35] “Fischer, M. M., & Nijkamp, P. (1992). Geographic information systems and spatial analysis. The Annals of Regional Science, 26(1), 3-17.”.

[36] “Walke, N., Reddy, G. O., Maji, A. K., & Thayalan, S. (2012). GIS-based multicriteria overlay analysis in soil-suitability evaluation for cotton (Gossypium spp.): A case study in the black soil region of Central India. Computers & Geosciences, 41, 108-118.”.

[37] “Tiede, D. (2014). A new geospatial overlay method for the analysis and visualization of spatial change patterns using object-oriented data modeling concepts. Cartography and Geographic Information Science, 41(3), 227-234.”.

[38] “Ma, M., Wu, Y., Luo, W., Chen, L., Li, J., & Jing, N. (2018). HiBuffer: buffer analysis of 10-million-scale spatial data in real time. ISPRS International Journal of Geo-Information, 7(12), 467.”.

[39] “Dong, P., Yang, C., Rui, X., Zhang, L., & Cheng, Q. (2003, July). An effective buffer generation method in GIS. In IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477) (Vol. 6, pp. 3706-3708).”.

[40] “Rahman, N. H., Rainis, R., Noor, S. H., & Mohamad, S. M. S. (2016). The Buffering analysis to identify common geographical factors within the vicinity of severe injury related to motor vehicle crash in Malaysia. World journal of emergency medicine, 7(4),”.

[41] “Li, X., Zhang, L., & Liang, C. (2010). A GIS-based buffer gradient analysis on spatiotemporal dynamics of urban expansion in Shanghai and its major satellite cities. Procedia Environmental Sciences, 2, 1139-1156.”.

[42] “Becker, L. T., & Gould, E. M. (2019). Microsoft power BI: extending excel to manipulate, analyze, and visualize diverse data. Serials Review, 45(3), 184-188.”.

[43] “API Amazon,” [Online]. Available: https://aws.amazon.com.

[44] “Application User Interface,” [Online]. Available: https://www.ibm.com.

[45] “Rahimi, A., Cohn, T., & Baldwin, T. (2016, August). pigeo: A python geotagging tool. In Proceedings of ACL-2016 System Demonstrations (pp. 127-132).”.

[46] “Mitlöhner, J., Neumaier, S., Umbrich, J., & Polleres, A. (2016, August). Characteristics of open data CSV files. In 2016 2nd International Conference on Open and Big Data (OBD) (pp. 72-79). IEEE.”.

[47] “Sharmila, S., & Sabarish, B. A. (2021, February). Analysis of distance measures in spatial trajectory data clustering. In IOP Conference Series: Materials Science and Engineering (Vol. 1085, No. 1, p. 012021). IOP Publishing.”.

[48] “Mora-Garcia, R. T., Marti-Ciriquian, P., Perez-Sanchez, R., & Cespedes-Lopez, M. F. (2018). A comparative analysis of manhattan, euclidean and network distances. Why are network distances more useful to urban professionals?. International Multidisciplinar”.

[49] “Prasetyo, S. E., Utomo, A. B., & Hudallah, N. (2018). Implementation of Google Maps API 3 with Haversine Algorithm in the Development of Geographic Information System Boarding House Finder. In Proceedings of the 7th Engineering International Conference on”.

[50] “Chopde, N. R., & Nichat, M. (2013). Landmark based shortest path detection by using A* and Haversine formula. International Journal of Innovative Research in Computer and Communication Engineering, 1(2), 298-302.”.

[51] “Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Survey review, 23(176), 88-93.”.

[52] “Distance on an ellipsoid,” [Online]. Available: https://community.esri.com.

APPENDIX: SOURCE CODE

The source code for the API

```
"""
Given the coordinates of an Accidental Zones in your city, and user travelling
in the city
calculate whether user is in accidental zone or not. Use the lattitude
langitude based distance messurement.

we will use following distance formulas:
1. Euclidiean
2. Haversine
3. Manhattan
4. Vincenty

"""
#Important Libraries
import geocoder
from geopy import distance as vinc
from math import *
from flask import Flask
from apscheduler.schedulers.background import BackgroundScheduler
import csv

#using flas app
app = Flask(__name__)

class Conflict_Points(object):
    def __init__(self, csv_file, threshold = 1):
        self.dist = 0
        self.threshold = 1
        self.csv_file = csv_file

        #Radius of earth in km
        self.R = 6371.0

    def euclid_distance(self, x,y):
        x = (radians(x[0]), radians(x[1]))
        y = (radians(y[0]), radians(y[1]))
        # x:  (x1, x2), y: (y1, y2) tuple or list of coordinates
        self.dist = sqrt( (x[1] - x[0])**2 + (y[1] - y[0])**2 )
        return self.dist

    def haversine(self, x, y):
        x = (radians(x[0]), radians(x[1]))
        y = (radians(y[0]), radians(y[1]))
```

```
        dlon = y[1] - x[1]
        dlat = y[0] - x[0]
        a = (sin(dlat/2))**2 + cos(x[0]) * cos(x[1]) * (sin(dlon/2))**2
        a = abs(a)
        c = 2 * atan2(sqrt(a), sqrt(1-a))
        self.dist = self.R * c
        return self.dist

    def manhatten(self, x, y):
        x = (radians(x[0]), radians(x[1]))
        y = (radians(y[0]), radians(y[1]))
        self.dist =  (x[1] - x[0])**2 + (y[1] - y[0])**2
        return self.dist

    def vincenty(self, x, y):
        self.dist = vinc.distance(x,y).km
        return self.dist

    def location_fetch_job(self):
        g = geocoder.ip('me')
        current_location = tuple(g.latlng)
        print("current location is : ", current_location)

        # print(" Distance name |  Difference  | Conflict or not")
        # self.dist = self.euclid_distance(current_location, self.acc_zone)
        # msg = "You are in safe zone" if self.dist > self.threshold else "you
are in conflict zone"

        # print(f"Euclidiean    | {self.dist}      | {msg} \n")

        # self.dist = self.manhatten(current_location, self.acc_zone)
        # msg = "You are in safe zone" if self.dist > self.threshold else "you
are in conflict zone"

        with open(self.csv_file, 'r') as file:
            data = list(csv.reader(file))
            data = data[1:]
            confict_flag = 0
            for row in data:
                try:
                    lat, long = float(row[6]), float(row[7])
                    self.acc_zone = (lat,long)
                except:
                    continue

                confict_flag = 0
```

```
                self.dist = self.haversine(current_location, self.acc_zone)

                if self.dist < self.threshold:
                    confict_flag = 1

                if confict_flag:
                    print(f"\nAlert !! Conflict Zone is Ahead{row[4], row[5],
row[2]}\n\tUsing Haversine, distance is : {self.dist} \n")

                confict_flag = 0
                self.dist = self.vincenty(current_location, self.acc_zone)

                if self.dist < self.threshold:
                    confict_flag = 1

                if confict_flag:
                    print(f"\nAlert !! Conflict Zone is Ahead{row[4], row[5],
row[2]}\n\tUsing Vincenity, distance is : {self.dist} \n")
            if not confict_flag: print("\nYou are probably in safe zone")



scheduler = BackgroundScheduler()
ob1 = Conflict_Points(csv_file='data.csv')
job = scheduler.add_job(ob1.location_fetch_job, 'interval', seconds=5)
scheduler.start()

if __name__ =="__main__":
    app.run()
```