CRIME HOTSPOT PREDICTION

A PROJECT REPORT

Submitted by

ARCHANA R M	**2116231801011**
DEEPIKA M	**2116231801028**
SUHIRTHA M P	**2116231801175**

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS), CHENNAI – 602 105

OCTOBER 2025

BONAFIDE CERTIFICATE

Certified that this Report titled "**CRIME HOTSPOT PREDICTION**" is the Bonafide work of "**ARCHANA R M (2116231801011), DEEPIKA M (2116231801028) and SUHIRTHA M P (2116231801175)**" who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. S. Sureshkumar M.E., Ph.D,

Professor,

Department of Artificial
Intelligence and Data science,

Rajalakshmi Engineering College

Thandalam – 602 105

Submitted to Project Viva-Voce Examination held on _____________

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

Initially I thank the Almighty for being with us through every walk of my life and showering his blessings through the endeavor to put forth this report.

My sincere thanks to our Chairman **Mr. S. MEGANATHAN, M.E., F.I.E.,** and our Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, M.E., Ph.D.,** for providing me with the requisite infrastructure and sincere endeavoring educating me in their premier institution.

My sincere thanks to **Dr. S.N. MURUGESAN M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time.

I express my sincere thanks to **Dr. J M Gnanasekar M.E., Ph.D.,** Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. I convey my sincere and deepest gratitude to our internal guide, **Dr. Suresh Kumar S M.E., Ph.D.,** Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally, I express my gratitude to my parents and classmates for their moral support and valuable suggestions during the course of the project.

ARCHANA R M	**DEEPIKA M**	**SUHIRTHA M P**
2116231801011	**2116231801028**	**2116231801175**

ABSTRACT

Urban crime is one of the most significant challenges facing modern cities today. Rapid urbanization, increasing population density, socio-economic disparity, and limited policing resources contribute to a rise in crime incidents across metropolitan and developing cities. Traditional crime analysis methods, which often rely on static Excel sheets, manual record-keeping, or simple GIS tools, are unable to handle the sheer volume, variety, and velocity of modern urban crime data. Moreover, manual processing limits timely decision-making and often results in inefficient resource allocation. The purpose of this project is to design a scalable Big Data-based system that can handle large volumes of multi-city crime data, identify high-crime areas or hotspots, and provide interactive visualizations to support law enforcement and urban planning authorities. Using Hive for distributed storage and querying, the system aggregates city-level crime data and computes essential metrics such as total crimes per city, top 20 hotspots, and the top three hotspots per city. The system also includes interactive visualization tools such as Folium maps for spatial representation and Chart.js charts for statistical summaries. By integrating data collection, preprocessing, Hive-based analysis, and visualization, this project provides a modular, flexible, and updatable system. This solution not only allows authorities to identify crime-prone regions quickly but also enables data-driven policy decisions, targeted patrol allocation, and future incorporation of predictive analytics and machine learning models. Ultimately, the project demonstrates the critical role of Big Data technologies in improving urban safety, community well-being, and governance efficiency.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

Urban areas worldwide are experiencing a consistent increase in crime due to complex socio-economic, demographic, and infrastructural factors. For example, metropolitan cities such as Delhi, Mumbai, and Chennai have faced challenges in monitoring and managing crime due to rapid population growth, unplanned urban sprawl, and high-density residential areas. Law enforcement agencies historically relied on manual reporting methods, paper records, or simple spreadsheets for crime analysis.

While GIS-based systems provide spatial visualization of crime incidents, they struggle to process large volumes of data, especially when multiple datasets from different sources are combined. The advent of Big Data technologies offers a solution, enabling distributed storage, parallel processing, and interactive visualization of crime datasets spanning multiple cities and years.

1.2 Motivation

The primary motivation behind this project is to provide authorities with a robust, scalable, and efficient system for crime analysis. Accurate identification of crime hotspots ensures efficient deployment of resources, reducing response time to incidents and enhancing public safety.

Furthermore, policymakers can leverage this data to design preventive strategies, improve urban infrastructure security, and implement community-based policing initiatives. From a research perspective, this project also demonstrates how Hive and Big Data architecture can be applied to practical, real-world social challenges, bridging the gap between theoretical analytics and actionable intelligence.

1.3 Objectives

1. To collect comprehensive city-level crime datasets from government portals, police records, and public repositories.
2. To preprocess data, handling missing values, duplicates, and inconsistent formatting, ensuring accurate analysis.
3. To store, manage, and query data efficiently using Hive for large-scale distributed storage.
4. To identify top 20 crime hotspots across multiple cities and top 3 hotspots per city using Hive queries and aggregation techniques.
5. To develop interactive dashboards combining Folium maps and Chart.js charts for visual exploration of crime data.
6. To provide insights that support policing strategies, urban planning, and public safety initiatives.
7. To design a modular system that can be updated for future enhancements, including predictive analytics and real-time monitoring.

1.4 Problem Statement

Crime data is inherently large-scale, heterogeneous, and spatially distributed. Traditional methods are unable to handle multiple datasets efficiently, leading to delays in analysis and decision-making. Manual identification of crime hotspots is time-consuming, prone to errors, and lacks interactivity, making it difficult for authorities to respond proactively. Furthermore, city planners and policymakers require integrated visualizations that combine spatial patterns, temporal trends, and crime types for informed decision-making.

1.5 Scope of Project

This project delivers a comprehensive multi-city crime analysis platform, designed to assist authorities in making data-driven decisions. It features interactive dashboards that visualize crime hotspots and provide detailed statistics for each city, enabling a clear understanding of local crime dynamics. Users can analyze trends over time, identify recurring patterns, and assess the severity and frequency of various criminal activities. The system is also highly scalable, with future enhancements planned to include predictive crime analysis, as well as integration with traffic data, population demographics, and urban infrastructure information, offering a holistic view for proactive crime prevention and urban planning.

CHAPTER 2

LITERATURE SURVEY

Crime hotspot analysis has been the subject of extensive research, spanning GIS-based studies, machine learning approaches, and Big Data analytics. Several notable works include:

1. GIS Crime Mapping: ArcGIS has been widely used to visualize crime incidents geographically. While effective for single-city or small datasets, GIS fails to scale when datasets exceed millions of records, limiting real-time insights.

2. Predictive Policing using Machine Learning: Studies using Python-based ML models such as Random Forest, Logistic Regression, and Neural Networks demonstrate crime prediction at granular levels. However, these models often rely on limited datasets and do not provide interactive dashboards, reducing their practical applicability.

3. Big Data Analytics with Hadoop and Hive: Hive enables distributed storage and processing, allowing cities to analyze crime trends across years and districts. However, prior research often neglects dashboard integration, leaving outputs as static tables that are difficult for decision-makers to interpret.

4. Dashboard Visualization using HTML/JS: Dashboards improve data interpretation for non-technical users. Integration with real-time datasets and interactive mapping can revolutionize decision-making, but many existing systems lack scalability and automation.

CHAPTER 3

ARCHITECTURE DIAGRAM

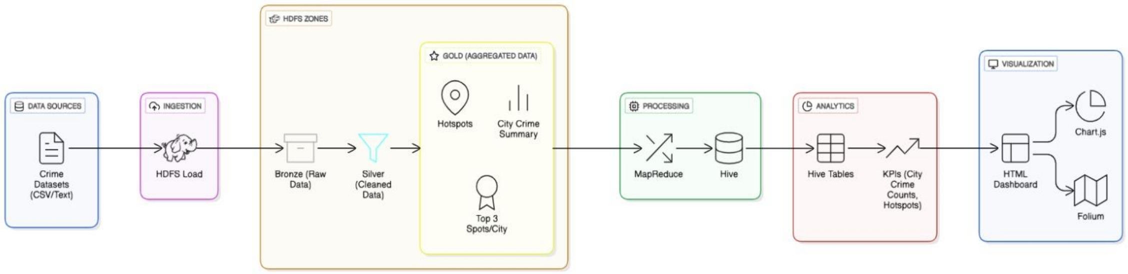


Fig 3. Architecture diagram

The architecture of our crime analysis project follows a structured Big Data pipeline, starting from data collection to visualization. Raw crime data is gathered from multiple sources, including government portals, police records, and public repositories, containing fields such as city, grid coordinates, crime type, and date. The data enters the ingestion layer using Hive on Hadoop, where it is stored in a distributed environment for scalability. HDFS organizes data into Bronze (raw), Silver (cleaned), and Gold (aggregated summaries like city-level totals and top hotspots) layers, ensuring quality and simplified analysis.

The processing and analytics layer uses Hive queries to perform aggregations, filtering, and ranking, producing insights such as city-wise crime counts, top 20 hotspots overall, and the top 3 spots per city. The visualization and dashboard layer presents the data interactively: Folium maps display hotspots and high-risk grids, while Chart.js pie charts summarize city-level crime distribution. Users can zoom, click markers for details, and filter data by city or time period, enabling police, city planners, and analysts to explore trends, prioritize resources, and make informed, data-driven decisions efficiently.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Existing System

The current approach to crime monitoring in most Indian cities relies heavily on manual reporting, police records, and Excel-based data storage. While these methods provide raw information, they have several limitations.

First, manual entry is error-prone, resulting in incomplete or inconsistent data. Second, analytical insights require aggregation and filtering, which is time-consuming and often performed monthly or annually.

Consequently, authorities cannot respond to emerging crime patterns promptly. Third, static reports cannot depict spatial distribution effectively, making it difficult to identify high-risk zones.

For example, in Chennai, repeated incidents in specific localities might only be noticed after a significant delay, preventing proactive policing. Moreover, traditional systems lack integration with demographic or socio-economic datasets, which could provide context for crime trends.

4.2 Proposed System:

The proposed system addresses the challenges of analyzing urban crime data by leveraging a Big Data architecture using Hive on Hadoop for distributed storage and efficient query execution. Crime data collected from multiple sources is standardized, cleaned, and aggregated into a structured format suitable for analysis. By organizing data into city-level and grid-level summaries, authorities can quickly identify high-crime areas and emerging hotspots.

The system integrates interactive dashboards, combining Folium maps for geospatial visualization of hotspots and Chart.js pie charts for statistical insights, such as city-wise crime distribution. Users can explore crime data by city, hotspot rank, and time period, enabling quick interpretation of trends and spatial patterns.

Designed with modularity and scalability in mind, the system can easily accommodate new cities, additional crime types, or supplementary data layers without extensive reconfiguration. For instance, stakeholders can overlay crime maps with other contextual information like population density to prioritize high-risk zones effectively. Overall, this system provides an intuitive and interactive platform for analyzing crime data, supporting informed decision-making for urban safety planning.

Proposed System Advantages:

- Automates aggregation and analysis of large-scale crime datasets.
- Provides interactive maps, charts, and dashboards for decision support.
- Modular and scalable, supporting future predictive analytics integration.

4.3 System Requirements

Software Requirements:

The system requires Ubuntu running in a VMware virtualized environment to facilitate the deployment of Hadoop and Hive clusters. Hive serves as a distributed query engine for managing structured crime datasets, while Hadoop provides scalable distributed storage to handle large volumes of multi-city data. Python is used for preprocessing, data cleaning, geocoding, and visualization tasks. For dashboard development, HTML, CSS, and JavaScript are utilized to create interactive charts and ensure a responsive design. Additionally, Folium is employed for geospatial mapping to visualize crime hotspots, and Chart.js is used to generate dynamic charts summarizing crime statistics.

Hardware Requirements:

The system requires a minimum of 8GB RAM and a dual-core processor to support small-scale testing. It also needs approximately 100GB of storage to accommodate datasets and intermediate results generated during processing.

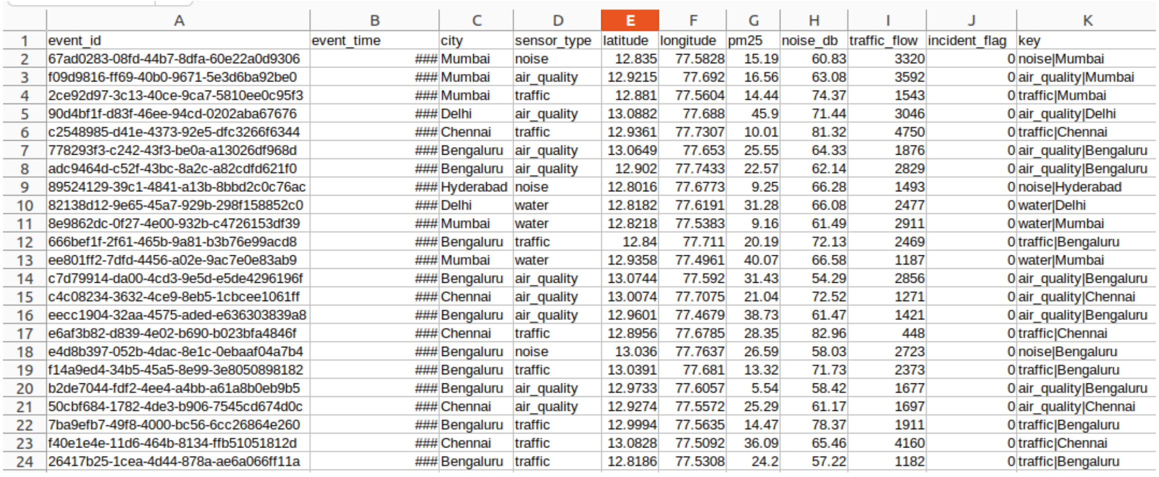
CHAPTER 5

MODULES DESCRIPTION

5.1 Data Collection Module

Data collection is the foundation of the project. Crime datasets are sourced from government open data portals, city police databases, and Kaggle repositories. The fields collected include City, Crime Type, Latitude, Longitude, and Date.

The module includes data validation steps, ensuring latitude and longitude values are within valid ranges and crime types are standardized. For example, entries like “theft” and “robbery” are consolidated under a single standardized category to maintain consistency. Real-world examples include Chennai reporting over 3000 crimes per month, requiring structured aggregation to avoid manual errors.



	A	B	C	D	E	F	G	H	I	J	K
1	event_id	event_time	city	sensor_type	latitude	longitude	pm25	noise_db	traffic_flow	incident_flag	key
2	67ad0283-08fd-44b7-8dfa-60e22a0d9306	###	Mumbai	noise	12.835	77.5828	15.19	60.83	3320	0	noise|Mumbai
3	f09d9816-ff69-40b0-9671-5e3d6ba92be0	###	Mumbai	air_quality	12.9215	77.692	16.56	63.08	3592	0	air_quality|Mumbai
4	2ce92d97-3c13-40ce-9ca7-5810ee0c95f3	###	Mumbai	traffic	12.881	77.5604	14.44	74.37	1543	0	traffic|Mumbai
5	90d4bf1f-d83f-46ee-94cd-0202aba67676	###	Delhi	air_quality	13.0882	77.688	45.9	71.44	3046	0	air_quality|Delhi
6	c2548985-d41e-4373-92e5-dfc3266f6344	###	Chennai	traffic	12.9361	77.7307	10.01	81.32	4750	0	traffic|Chennai
7	778293f3-c242-43f3-be0a-a13026df968d	###	Bengaluru	air_quality	13.0649	77.653	25.55	64.33	1876	0	air_quality|Bengaluru
8	adc9464d-c52f-43bc-8a2c-a82cdfd621f0	###	Bengaluru	air_quality	12.902	77.7433	22.57	62.14	2829	0	air_quality|Bengaluru
9	89524129-39c1-4841-a13b-8bbd2c0c76ac	###	Hyderabad	noise	12.8016	77.6773	9.25	66.28	1493	0	noise|Hyderabad
10	82138d12-9e65-45a7-929b-298f158852c0	###	Delhi	water	12.8182	77.6191	31.28	66.08	2477	0	water|Delhi
11	8e9862dc-0f27-4e00-932b-c4726153df39	###	Mumbai	water	12.8218	77.5383	9.16	61.49	2911	0	water|Mumbai
12	666bef1f-2f61-465b-9a81-b3b76e99acd8	###	Bengaluru	traffic	12.84	77.711	20.19	72.13	2469	0	traffic|Bengaluru
13	ee801ff2-7dfd-4456-a02e-9ac7e0e83ab9	###	Mumbai	water	12.9358	77.4961	40.07	66.58	1187	0	water|Mumbai
14	c7d79914-da00-4cd3-9e5d-e5de4296196f	###	Bengaluru	air_quality	13.0744	77.592	31.43	54.29	2856	0	air_quality|Bengaluru
15	c4c08234-3632-4ce9-8eb5-1cbcee1061ff	###	Chennai	air_quality	13.0074	77.7075	21.04	72.52	1271	0	air_quality|Chennai
16	eecc1904-32aa-4575-aded-e636303839a8	###	Bengaluru	air_quality	12.9601	77.4679	38.73	61.47	1421	0	air_quality|Bengaluru
17	e6af3b82-d839-4e02-b690-b023bfa4846f	###	Chennai	traffic	12.8956	77.6785	28.35	82.96	448	0	traffic|Chennai
18	e4d8b397-052b-4dac-8e1c-0ebaaf04a7b4	###	Bengaluru	noise	13.036	77.7637	26.59	58.03	2723	0	noise|Bengaluru
19	f14a9ed4-34b5-45a5-8e99-3e8050898182	###	Bengaluru	traffic	13.0391	77.681	13.32	71.73	2373	0	traffic|Bengaluru
20	b2de7044-fdf2-4ee4-a4bb-a61a8b0eb9b5	###	Bengaluru	air_quality	12.9733	77.6057	5.54	58.42	1677	0	air_quality|Bengaluru
21	50cbf684-1782-4de3-b906-7545cd674d0c	###	Chennai	air_quality	12.9274	77.5572	25.29	61.17	1697	0	air_quality|Chennai
22	7ba9efb7-49f8-4000-bc56-6cc26864e260	###	Bengaluru	traffic	12.9994	77.5635	14.47	78.37	1911	0	traffic|Bengaluru
23	f40e1e4e-11d6-464b-8134-ffb51051812d	###	Chennai	traffic	13.0828	77.5092	36.09	65.46	4160	0	traffic|Chennai
24	26417b25-1cea-4d44-878a-ae6a066ff11a	###	Bengaluru	traffic	12.8186	77.5308	24.2	57.22	1182	0	traffic|Bengaluru

Fig 5.1. Dataset

5.2 Data Preprocessing Module

Raw crime datasets often contain duplicates, missing values, inconsistent naming, and incorrect geolocation. Preprocessing involves:

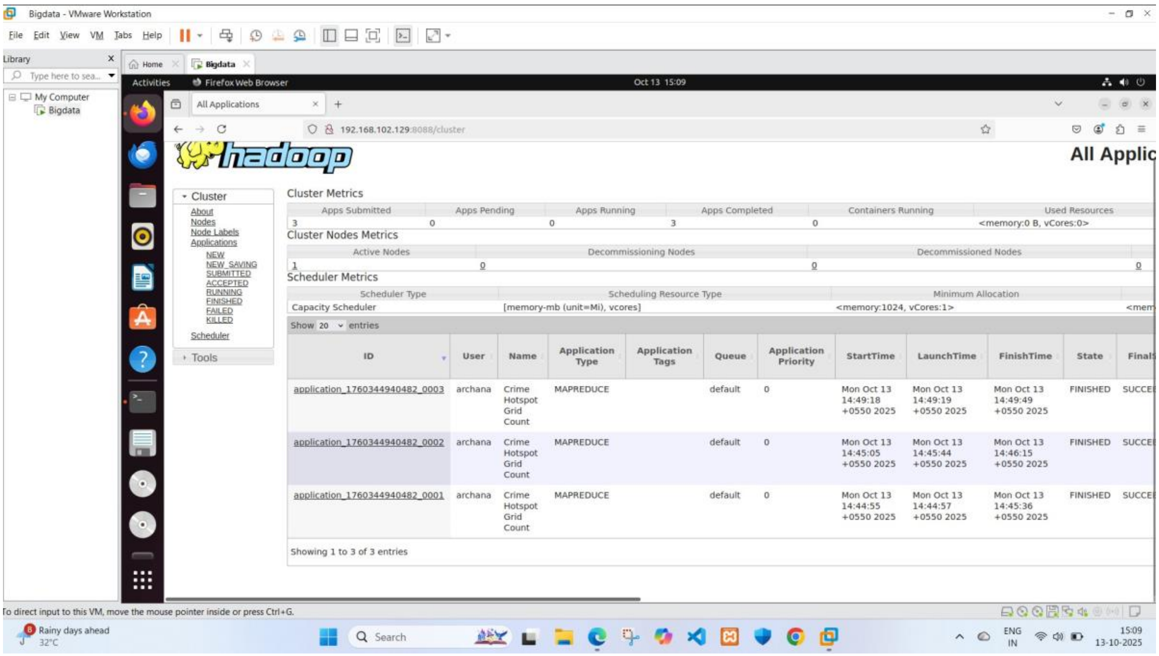
1. Removing duplicates and nulls: Ensures accurate counts and reduces storage redundancy.
2. Standardizing city and crime names: For example, “Chennai” is corrected to “Chennai.”
3. Geocoding: Converts addresses or location names to latitude/longitude coordinates for map plotting.
4. Grid creation: The city map is divided into grids, and each crime is assigned to a grid for hotspot analysis.

   By preprocessing data accurately, we ensure Hive queries produce meaningful results, and visualizations reflect true patterns.

5.3 MapReduce Module

Before Hive analysis, MapReduce is used for large-scale data aggregation.

- Mapper: Reads each crime record, outputs key-value pairs by city or grid.
- Reducer: Aggregates counts per city, grid, or crime type.
  This parallel processing reduces computation time on large datasets and produces intermediate outputs like total crimes per city or crime counts per grid. The output is stored as hotspot.csv



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources
3	0	0	3	0	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	
1	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<mem

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalS
application_1760344940482_0003	archana	Crime Hotspot Grid Count	MAPREDUCE		default	0	Mon Oct 13 14:49:18 +0550 2025	Mon Oct 13 14:49:19 +0550 2025	Mon Oct 13 14:49:49 +0550 2025	FINISHED	SUCCE
application_1760344940482_0002	archana	Crime Hotspot Grid Count	MAPREDUCE		default	0	Mon Oct 13 14:45:05 +0550 2025	Mon Oct 13 14:45:44 +0550 2025	Mon Oct 13 14:46:15 +0550 2025	FINISHED	SUCCE
application_1760344940482_0001	archana	Crime Hotspot Grid Count	MAPREDUCE		default	0	Mon Oct 13 14:44:55 +0550 2025	Mon Oct 13 14:44:57 +0550 2025	Mon Oct 13 14:45:36 +0550 2025	FINISHED	SUCCE

Showing 1 to 3 of 3 entries

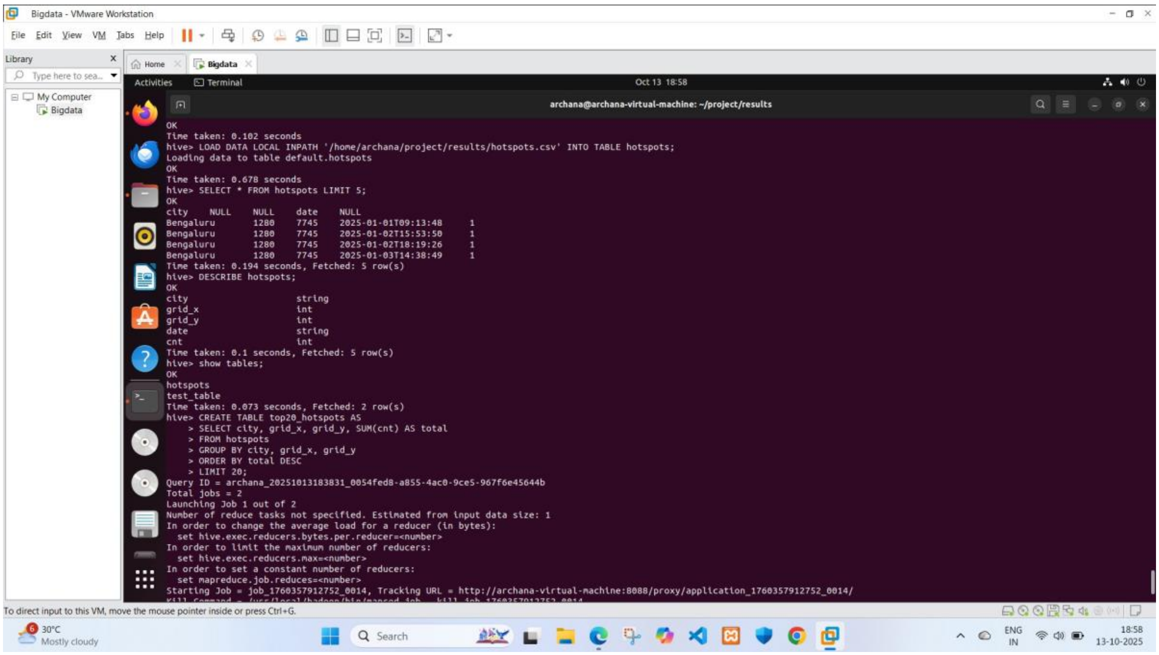
Fig 5.3. MapReduce

5.4 Hive Query & Analysis Module

Hive is used to perform structured analysis on large-scale datasets.

Hive Table Creation:

```
CREATE TABLE crime_data (
 city STRING,
 crime_type STRING,
 latitude DOUBLE,
 longitude DOUBLE,
 date STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```



```
OK
Time taken: 0.102 seconds
hive> LOAD DATA LOCAL INPATH '/home/archana/project/results/hotspots.csv' INTO TABLE hotspots;
Loading data to table default.hotspots
OK
Time taken: 0.678 seconds
hive> SELECT * FROM hotspots LIMIT 5;
OK
city    NULL    NULL    date    NULL
Bengaluru       1280    7745    2025-01-01T09:13:48     1
Bengaluru       1280    7745    2025-01-02T15:53:50     1
Bengaluru       1280    7745    2025-01-02T18:19:26     1
Bengaluru       1280    7745    2025-01-03T14:38:49     1
Time taken: 0.194 seconds, Fetched: 5 row(s)
hive> DESCRIBE hotspots;
OK
city                    string
grid_x                  int
grid_y                  int
date                    string
cnt                     int
Time taken: 0.1 seconds, Fetched: 5 row(s)
hive> show tables;
OK
hotspots
test_table
Time taken: 0.073 seconds, Fetched: 2 row(s)
hive> CREATE TABLE top20_hotspots AS
    > SELECT city, grid_x, grid_y, SUM(cnt) AS total
    > FROM hotspots
    > GROUP BY city, grid_x, grid_y
    > ORDER BY total DESC
    > LIMIT 20;
Query ID = archana_20251013183831_0054fed8-a855-4ac0-9ce5-967f6e45644b
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1760357912752_0014, Tracking URL = http://archana-virtual-machine:8088/proxy/application_1760357912752_0014/
```

Bigdata - VMware Workstation
File
Edit
View
VM
Tabs
Help
Library
Type here to sea...
My Computer
Bigdata
Home
Bigdata
Activities
Terminal
Oct 13 18:58
archana@archana-virtual-machine: ~/project/results
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
30°C
Mostly cloudy
Search
ENG
IN
18:58
13-10-2025

Fig 5.4.1. Hotspots table

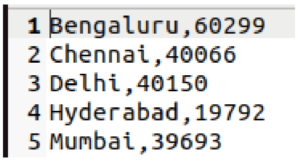
City-wise Crime Count:

SELECT city, COUNT(*) AS total_crimes

FROM crime_data

GROUP BY city

ORDER BY total_crimes DESC;



```
Bengaluru,60299
Chennai,40066
Delhi,40150
Hyderabad,19792
Mumbai,39693
```

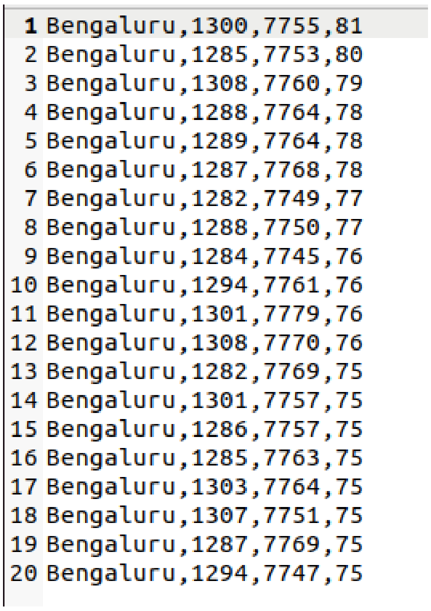
Fig 5.4.2. City wise crime count

Top 20 Hotspots Overall:

SELECT latitude, longitude, COUNT(*) AS crime_count

FROM crime_data

```
GROUP BY latitude, longitude

ORDER BY crime_count DESC

LIMIT 20;
```

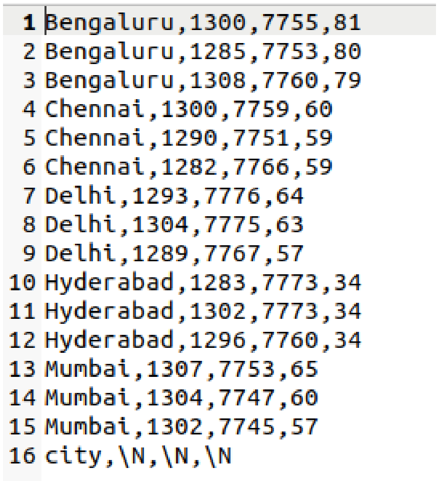


```
Bengaluru,1300,7755,81
Bengaluru,1285,7753,80
Bengaluru,1308,7760,79
Bengaluru,1288,7764,78
Bengaluru,1289,7764,78
Bengaluru,1287,7768,78
Bengaluru,1282,7749,77
Bengaluru,1288,7750,77
Bengaluru,1284,7745,76
Bengaluru,1294,7761,76
Bengaluru,1301,7779,76
Bengaluru,1308,7770,76
Bengaluru,1282,7769,75
Bengaluru,1301,7757,75
Bengaluru,1286,7757,75
Bengaluru,1285,7763,75
Bengaluru,1303,7764,75
Bengaluru,1307,7751,75
Bengaluru,1287,7769,75
Bengaluru,1294,7747,75
```

Fig 5.4.3. Top 20 Hotspots

Top 3 Grids per City:

```
CREATE TABLE city_grid_crime AS

SELECT city, grid_x, grid_y, SUM(cnt) AS total_crime

FROM hotspots

GROUP BY city, grid_x, grid_y;



CREATE TABLE top3_spots_per_city AS

SELECT city, grid_x, grid_y, total_crime

FROM (

    SELECT city,
        grid_x,

        grid_y,

        total_crime,

        ROW_NUMBER() OVER (PARTITION BY city ORDER BY
total_crime DESC) AS rank

   FROM city_grid_crime

) ranked

WHERE rank <= 3;
```



```
Bengaluru,1300,7755,81
Bengaluru,1285,7753,80
Bengaluru,1308,7760,79
Chennai,1300,7759,60
Chennai,1290,7751,59
Chennai,1282,7766,59
Delhi,1293,7776,64
Delhi,1304,7775,63
Delhi,1289,7767,57
Hyderabad,1283,7773,34
Hyderabad,1302,7773,34
Hyderabad,1296,7760,34
Mumbai,1307,7753,65
Mumbai,1304,7747,60
Mumbai,1302,7745,57
city,\N,\N,\N
```

Fig 5.4.4. Top 3 Grids per City

The queries aggregate crime counts by city and grid, then rank grids to identify priority hotspots for policing. This allows authorities to visualize recurring crime locations efficiently.

5.5 Visualization Module

The Visualization Module transforms processed crime data into interactive and easy-to-understand visual representations, helping users quickly identify patterns, hotspots, and trends.

Folium Maps:

Incident Mapping: Crime records are plotted on interactive maps using latitude and longitude. Marker clustering is applied to group nearby incidents and reduce clutter.

Hotspot Heatmaps: High-risk zones are highlighted using color-coded intensity layers:

Red: Areas with frequent incidents

Yellow: Moderate crime concentration

Green: Low crime areas

Popups: Clicking a marker shows contextual details such as city, number of incidents, and grid location.

Chart.js Visualizations:

Pie Charts: Show the distribution of total crimes across cities (e.g., Bengaluru, Chennai, Delhi, Hyderabad, Mumbai).

Bar Charts (optional): Can be used to display monthly crime trends or compare total incidents across cities.

Interactive Design: All maps and charts are responsive and adjust automatically for different screen sizes.

The visualization module provides a clear overview of crime data for decision-making and analysis.

5.6 Dashboard Module

The Dashboard Module integrates all visualizations into a single interactive platform for exploring and interpreting crime data.

Key Features Implemented:

City-Wise Summary: A pie chart shows total crimes per city for quick comparison.

Hotspot Maps:

Top 20 hotspots across all cities.

Top 3 hotspots per city for local analysis.

Interactivity: Users can zoom, click markers for details, and filter data by city or hotspot.

User Benefits:

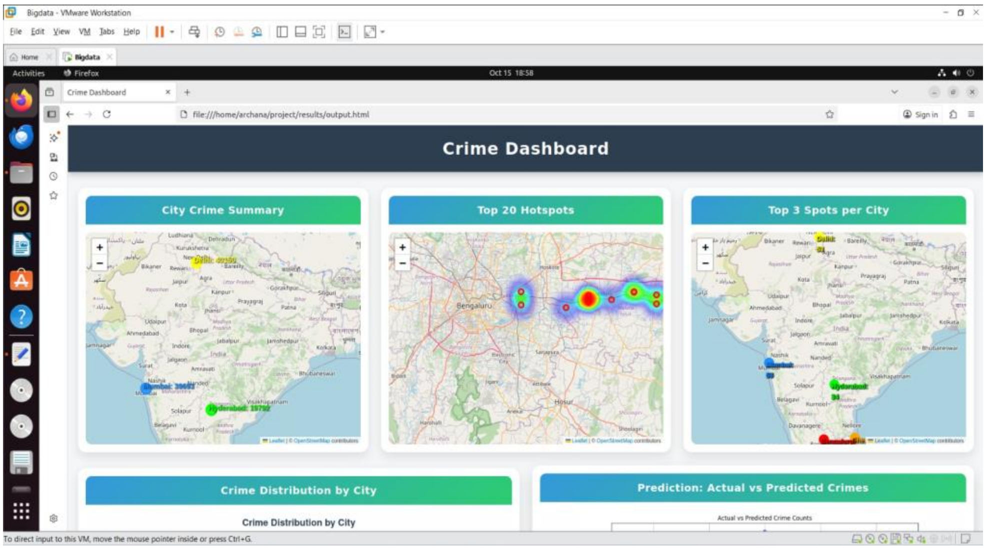
Helps police and officials prioritize resources.

Enables analysts to spot trends and high-risk areas.

Provides a clear visual overview for all stakeholders.

Summary: Combines processed crime data into a single, visual, interactive interface for quick decision-making.

DASHBOARD OUTPUT:


Bigdata - VMware Workstation
File Edit View VM Tabs Help
Home
Bigdata
Activities
Firefox
Oct 15 18:58
Crime Dashboard
file:///home/archana/project/results/output.html
Sign in
Crime Dashboard
City Crime Summary
Top 20 Hotspots
Top 3 Spots per City
Crime Distribution by City
Prediction: Actual vs Predicted Crimes
Crime Distribution by City
Actual vs Predicted Crime Counts
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

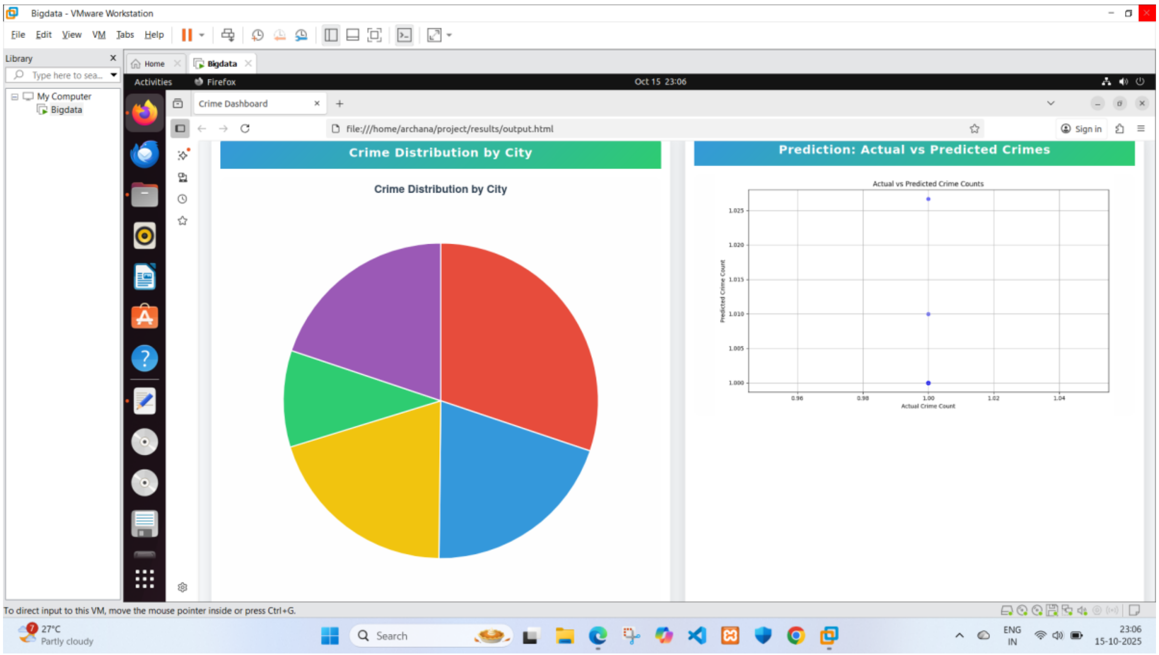

Bigdata - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
Bigdata
Home
Bigdata
Activities
Firefox
Oct 15 23:06
Crime Dashboard
file:///home/archana/project/results/output.html
Sign in
Crime Distribution by City
Prediction: Actual vs Predicted Crimes
Crime Distribution by City
Actual vs Predicted Crime Counts
1.025
1.020
1.015
1.010
1.005
1.000
Predicted Crime Count
0.96
0.98
1.00
1.02
1.04
Actual Crime Count
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
27°C
Partly cloudy
Search
ENG
IN
23:06
15-10-2025

Fig 4.5. Dashboard

CHAPTER 6

IMPLEMENTATION

The implementation phase involves bringing together all modules—data collection, preprocessing, Hive analytics, and visualization—into a functional system. The workflow begins with uploading raw crime datasets into Hive, ensuring that each record contains structured fields such as city, crime type, latitude, longitude, and date. Data from multiple sources often comes in inconsistent formats, so preprocessing scripts written in Python are executed first. These scripts handle missing or incorrect coordinates, duplicates, inconsistent city or crime type naming, and any outliers. For example, if a crime location is recorded as latitude 0 or longitude 0, it is flagged and corrected based on address geocoding APIs.

Once the datasets are clean, Hive tables are created to store and query the data. Queries are executed to compute city-wise crime counts, top 20 hotspots, and top 3 hotspots per city, using aggregations, groupings, and window functions. Each Hive query is validated with sample datasets to ensure correctness. For instance, the top 3 hotspot query uses ROW_NUMBER() with partitioning by city, ensuring that each city's top grids are calculated independently, avoiding bias towards cities with higher overall crime counts.

After the analysis, results are exported as CSV or JSON files to feed into the visualization module. Using Folium in Python, interactive maps are created. Each hotspot grid is color-coded based on crime intensity, with red indicating high-density areas and green for low-density regions. Marker pop-ups provide detailed information on crime counts and types in each grid. Simultaneously, Chart.js is used to create dynamic charts: pie charts for city-wise crime distribution and bar/line charts for temporal trends.

These charts are linked to the dashboard, enabling filtering by city, crime type, and date range.

Finally, the dashboard integrates all visualizations into a single HTML page. Users can interactively explore hotspots, view top crime grids, and analyze temporal patterns. Real-time features, such as updating the dashboard with newly uploaded CSV files, are implemented using Python scripts scheduled via cron jobs. The modular design ensures that new cities, additional data layers, or predictive models can be added with minimal effort, making the system future-ready.

CHAPTER 7

RESULTS AND DISCUSSION

The crime analysis project successfully processed and visualized crime data across multiple cities, providing insights into patterns, trends, and high-risk areas. Using the ingestion pipeline, raw data was loaded into structured formats, and processed through aggregation steps to generate meaningful metrics such as total crimes per city and hotspot intensity. The processed data was then transformed into interactive visualizations, making it accessible for decision-making without requiring deep technical knowledge.

The Dashboard Module consolidates all visualizations into a single platform. The city-wise summary, presented as a pie chart, clearly highlights which cities experience higher crime rates, allowing for quick comparison. Hotspot maps, including the top 20 locations across all cities and the top 3 per city, provide spatial context, helping stakeholders identify specific high-risk areas. These maps include interactive features like zooming, panning, and popups displaying detailed information, which improves understanding of local crime dynamics.

From the visualizations, it is evident that certain cities consistently show higher crime counts, while specific grids within these cities emerge as recurrent hotspots. The interactive nature of the dashboard allows users to filter and focus on areas of interest, revealing temporal and spatial trends that could guide resource allocation. Police officials, city planners, and analysts can leverage these insights for planning targeted interventions, improving public safety, and monitoring the effectiveness of policies.

Overall, the project demonstrates how raw crime data can be transformed into actionable intelligence. The combination of maps and charts in an interactive dashboard provides a comprehensive view of urban crime patterns, enabling stakeholders to make informed decisions and prioritize preventive measures effectively. The results confirm that visualization is a powerful tool for understanding complex datasets and communicating findings clearly to a wide audience.

CHAPTER 8

FUTURE ENHANCEMENTS

The proposed system can be expanded in multiple ways:

1. Real-time Data Integration: By connecting the dashboard directly to police databases and IoT-enabled surveillance systems, authorities can receive instantaneous updates on new incidents. This would enable dynamic hotspot updates and faster decision-making.
2. Predictive Crime Analytics: Using machine learning models such as Random Forests, XGBoost, or Neural Networks, the system can predict potential crime locations based on historical trends, temporal patterns, and socio-economic factors. This proactive approach would allow police forces to anticipate crime surges and deploy resources accordingly.
3. Integration with Urban Data Layers: By overlaying population density, traffic flow, CCTV camera coverage, and socio-economic indicators, authorities can gain deeper insights into crime causality and risk factors.
4. Public Awareness Portal: A web-accessible dashboard for citizens can inform residents of high-risk areas, promoting community vigilance and self-protective behavior.
5. Advanced Visualization Techniques: Implementing heatmaps, 3D grids, and interactive temporal sliders can enhance the visualization of crime patterns, making the system more intuitive for decision-makers.
6. Mobile app development: People can use it and be safe.

CHAPTER 9

REFERENCES

1. **Apache Hive Documentation** – “Apache Hive – Data Warehousing and SQL-Like Queries.” [Online]. Available: https://cwiki.apache.org/confluence/display/Hive/Home

2. **Apache Hadoop Documentation** – “Hadoop: Distributed Storage and Processing Framework.” [Online]. Available: https://hadoop.apache.org/

3. **Folium Python Library Documentation** – “Folium – Python Data Visualization for Maps.” [Online]. Available: https://python-visualization.github.io/folium/

4. **Chart.js Documentation** – “Chart.js – Open Source JavaScript Charts.” [Online]. Available: https://www.chartjs.org/

5. **Government of India Open Data Portal** – “Crime India Dataset.” [Online]. Available: https://data.gov.in/catalog/crime-india

6. R. S. Brunsdon and L. Comber, *An Introduction to GIS and Spatial Analysis in Criminology*, 2nd ed., Routledge, 2020.

7. J. Ratcliffe, *GIS and Crime Mapping*, Wiley, 2016.

8. A. Mohler, M. Short, P. Brantingham, F. Schoenberg, and G. Tita, "Self-Exciting Point Process Modeling of Crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011. [Online]. Available: https://www.tandfonline.com/doi/full/10.1198/jasa.2011.ap09546

9. P. Wang, W. Lin, and X. Li, "Crime Prediction Using Big Data and Machine Learning," *IEEE Access*, vol. 7, pp. 125394–125404, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8765605

10. B. K. Mohanty, S. Das, and A. Panigrahi, "Big Data Analytics for Urban Crime Analysis: Tools and Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 3, pp. 456–464, 2019. [Online]. Available: https://thesai.org/Downloads/Volume10No3/Paper_60-Big_Data_Analytics_for_Urban_Crime.pdf