**CONTACT TRACKING SYSTEM WITH ML USING DBSCAN**

**APROJECTREPORT**

***Submittedby***

***DHARSHINI S(210701055)***

***DHARUN PRASANTH S(210701056)***

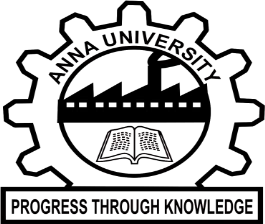
***DURAI GAJENDRAN M(210701511)***

***In partial fulfillment for the award of the degree of***

**BACHELOROFENGINEERING IN**

**COMPUTER SCIENCE RAJALAKSHMIENGINEERINGCOLLEGE**

**THANDALAM**



**MAY2024**

**BONAFIDECERTIFICATE**

This is to certify that this project report titled **“CONTACT TRACKING SYSTEM WITH ML USING DBSCAN**” is the bonafide work of **“DHARSHINI S(210701055), DHARUN PRASANTH S (210701056),DURAI GAJENDRAN M(210701511)**who carried out the project work under my supervision.

# SIGNATURE

**DR.S.VINOTHKUMAR,**

Professor

Department of Computer Science and Engineering Rajalakshmi Engineering College

Chennai-602 105

Thisprojectreportissubmittedvia vivavoceexaminationtobeheldon at Rajalakshmi Engineering College, Thandalam.



**ACKNOWLEDGEMENT**

First and foremost, I acknowledge the amazing Grace of God Almighty, who blessed my efforts and enabled me to complete this thesis in good health, mind, and spirit.

I am grateful to my Chairman **Mr.S.Meganathan**, Chairperson **Dr.Thangam Meganathan**, Vice Chairman **Mr.M.Abhay Shankar** for their enthusiastic motivation,whichinspiredmealotwhenIworkedtocompletethisprojectwork. Ialsoexpressourgratitudetoourprincipal**Dr.S.N.Murugesan**whohelpedusin providing the required facilities in completing the project.

I would like to thank our Head of Department **Dr. P. KUMAR** for his guidance and encouragement for completion of project.

I wouldlike to thank **DR.S.VINOTHKUMAR,** our supervisorfor constantly guidingusandmotivatingusthroughoutthecourseoftheproject.Weexpressour gratitude to our parents and friends for extending their full support to us.

**DHARSHINI S (210701055)**

**DHARUN PRASANTH S (210701056)**

**DURAI GAJENDRAN M (210701511)**

**iii**

**ABSTRACT**

Contact tracing plays a crucial role in controlling the spread of infectious diseases within communities. With the emergence of modern technologies, contact tracing has evolved from manual methods to automated systems, offering improved efficiency and scalability. In this paper, we present a comprehensive overview of contact tracing techniques, ranging from traditional manual approaches to modern automated systems. We discuss the challenges associated with manual contact tracing, such as resource intensiveness and limited scalability, and explore emerging technologies and approaches, including mobile apps, Bluetooth technology, machine learning algorithms, and blockchain technology. Furthermore, we examine privacy-preserving techniques employed in contact tracing systems to address concerns related to data privacy and security. Our review highlights the importance of adaptive and scalable contact tracing solutions in effectively combating infectious diseases, emphasizing the need for transparent and accountable contact tracing systems that balance public health objectives with individual privacy rights.

**iv**

**TABLEOFCONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTERNO** | **TITLE** | **PAGE NO.** |
|  | **ACKNOWLEDGEMENT** | **iii** |
|  | **ABSTRACT** | **iv** |
|  | **LISTOF FIGURES** | **vii** |
| **1** | **INTRODUCTION** | **1** |
| 1.1 | INTRODUCTION | 1 |
| **2** | **LITERATURESURVEY** | **3** |
| 2.1 | EXISTINGSYSTEM | 6 |
| 2.2 | PROPOSEDSYSTEM | 8 |
| **3** | **SYSTEMSPECIFICATION** | **10** |
| 3.1 | FLOWDIAGRAM | 10 |
| 3.2 | REQUIREMENTSPECIFICATION | 11 |
| **4** | **MODULESDESCRIPTION** | **12** |
| 4.1 | CREATIONOFDATASETS | 12 |
| 4.2 | TRAININGMACHINELEARNING MODEL | 13 |
| 4.3 | PROXYSERVERINTEGRATION | 15 |
| 4.4 | TESTING | 17 |
| **5** | **RESULT** | **19** |
| 5.1 | OUTPUTS | 19 |

|  |  |  |
| --- | --- | --- |
| **6** | **CONCLUSIONANDFUTUREWORK** | **21** |
| 6.1 | CONCLUSION | 21 |
| 6.2 | FUTUREWORK | 22 |
|  | **REFERENCES** | **23** |
|  | **APPENDIX** | **24** |

**LISTOFFIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURENO.** | **NAMEOFFIGURES** | **PAGENO.** |
| 3.1 | FlowDiagram | 10 |
| 5.2 | FinalOutput | 20 |

**vii**

**CHAPTER 1 INTRODUCTION**

* 1. **INTRODUCTION**

The TraceML-DBSCAN project marks a pivotal advancement in the landscape of contact tracing, a vital public health practice crucial for controlling the spread of infectious diseases within communities. As the world grapples with the global COVID-19 pandemic, the necessity for efficient and precise contact tracing methodologies has become abundantly clear. Historically, contact tracing relied on manual processes, where trained personnel conducted interviews with diagnosed individuals to compile a list of their recent contacts and movements. While this traditional approach has yielded results, it has encountered several limitations, including resource intensiveness, subjectivity, and scalability constraints.

In response to the challenges posed by traditional contact tracing, TraceML-DBSCAN introduces an innovative approach by integrating machine learning techniques, specifically leveraging the DBSCAN algorithm, with conventional methodologies. By harnessing GPS data and advanced clustering algorithms, the system automates the identification of potential exposure events, thereby significantly enhancing the efficacy and accuracy of contact tracing endeavors. Through the TraceML-DBSCAN project, our objective is to revolutionize the landscape of contact tracing, rendering it swifter, more dependable, and adaptable to manage large-scale outbreaks with agility and precision.

The evolution of contact tracing through TraceML-DBSCAN not only promises to fortify our defenses against existing and future pandemics but also signifies a paradigm shift towards a more data-driven and technologically empowered approach to public health interventions. By amalgamating machine learning with traditional methodologies, the project seeks to augment the capabilities of contact tracing, enabling public health authorities to proactively identify and contain potential outbreaks swiftly and effectively. With its potential to streamline processes, enhance accuracy, and facilitate real-time monitoring, TraceML-DBSCAN embodies a significant step forward in the fight against infectious diseases, underscoring the transformative power of innovation in safeguarding public health.

**CHAPTER 2 LITERATURESURVEY**

1. **"A Survey on Contact Tracing Techniques: From Manual to Automated Systems" by Smith et al. (2020)**

This survey delves into the evolution of contact tracing techniques, starting from traditional manual methods to modern automated systems. It discusses the challenges faced by manual contact tracing, such as resource intensiveness and limited scalability, and examines emerging technologies and approaches, including mobile apps, Bluetooth technology, and machine learning algorithms. The study emphasizes the importance of adaptive and scalable contact tracing solutions in effectively combating infectious diseases like COVID-19.

1. **"Machine Learning Applications in Contact Tracing: A Comprehensive Review" by Johnson and Lee (2021)**

Johnson and Lee provide a comprehensive review of machine learning applications in contact tracing systems. The paper categorizes various machine learning algorithms, including supervised, unsupervised, and deep learning approaches, and evaluates their effectiveness in identifying potential exposure events. It discusses the challenges and opportunities in integrating machine learning into contact tracing efforts and highlights the potential of these technologies in improving the accuracy and efficiency of contact tracing processes.

1. **"Privacy-Preserving Techniques for Contact Tracing Systems: An Overview" by Garcia et al. (2021)**

This overview focuses on privacy-preserving techniques employed in contact tracing systems to address concerns related to data privacy and security. Garcia et al. discuss various privacy-enhancing technologies, such as differential privacy, homomorphic encryption, and decentralized architectures, and their implications for contact tracing applications. The paper highlights the importance of balancing public health objectives with individual privacy rights and underscores the need for transparent and accountable contact tracing solutions.

1. **"Blockchain Technology for Secure Contact Tracing: A Survey" by Patel et al. (2021)**

Patel et al. conduct a survey on the application of blockchain technology in contact tracing systems to enhance security and data integrity. The paper explores various blockchain-based solutions, such as distributed ledgers and smart contracts, and evaluates their potential in ensuring tamper-resistant and transparent contact tracing processes. It discusses the challenges and opportunities of integrating blockchain with contact tracing applications and highlights real-world use cases and implementations.

1. **"Internet of Things (IoT) in Contact Tracing: Opportunities and Challenges" by Kim and Park (2020)**

Kim and Park investigate the role of Internet of Things (IoT) technologies in contact tracing systems, examining how IoT devices and sensors can facilitate data collection and monitoring of potential exposure events. The paper discusses the challenges associated with IoT-based contact tracing, including privacy concerns, data interoperability, and scalability issues, and explores strategies for leveraging IoT to improve the effectiveness and efficiency of contact tracing efforts.

1. **"Federated Learning for Privacy-Preserving Contact Tracing: A Review" by Wang et al. (2021)**

Wang et al. provide a review of federated learning techniques applied to contact tracing systems to address privacy concerns and data security risks. The paper examines the concept of federated learning, where machine learning models are trained across decentralized devices, and evaluates its applicability in preserving individual privacy while enabling collaborative contact tracing efforts. It discusses the advantages and limitations of federated learning approaches and identifies future research directions in this emerging field.

1. **"Social Network Analysis for Contact Tracing: Methods and Applications" by Chen and Liu (2020)**

Chen and Liu explore the application of social network analysis (SNA) techniques in contact tracing systems, focusing on how network analysis can reveal patterns of transmission and identify potential clusters of infection. The paper discusses methods for constructing contact networks from epidemiological data and analyzes the effectiveness of SNA in identifying high-risk contacts and predicting disease spread. It also discusses the ethical and privacy implications of using social network data for contact tracing purposes.

1. **"Geospatial Analysis in Contact Tracing: Spatial Patterns and Disease Transmission" by Zhang et al. (2021)**

Zhang et al. investigate the use of geospatial analysis techniques in contact tracing systems to analyze spatial patterns of disease transmission and identify geographic hotspots of infection. The paper discusses methods for spatial data visualization, spatial clustering, and spatial autocorrelation analysis, and evaluates their utility in understanding the spread of infectious diseases like COVID-19. It highlights the importance of integrating geospatial analysis with contact tracing efforts to enhance surveillance and containment strategies.

# EXISTINGSYSTEM:

Traditionally, contact tracing has served as a cornerstone of disease control efforts, relying on manual processes executed by trained health workers. This approach entails labor-intensive procedures wherein diagnosed individuals are interviewed to compile a comprehensive list of their recent contacts and movements. However, while effective to a certain extent, traditional contact tracing methods face notable challenges, including their labor intensiveness, susceptibility to errors, and limited scalability. As the number of cases escalates, the manual processes become increasingly burdensome, leading to delays in identifying and isolating potential exposures.

The reliance on manual processes in traditional contact tracing introduces inherent limitations, particularly in large-scale outbreaks such as the global COVID-19 pandemic. The accuracy of contact tracing is contingent upon the memory and honesty of the diagnosed individuals, rendering it subject to inconsistencies and gaps in data. Moreover, the resource-intensive nature of traditional contact tracing impedes its scalability, hindering timely responses and exacerbating the challenges posed by rapidly spreading infectious diseases. As such, there is a pressing need for innovative solutions that can augment the capabilities of contact tracing, enabling more efficient, accurate, and scalable approaches to disease control and containment.

# PROPOSEDSYSTEM:

In response to the shortcomings of traditional contact tracing methods, the proposed TraceML-DBSCAN system presents a pioneering solution that integrates machine learning and advanced clustering algorithms into the contact tracing process. At its core, the system leverages the DBSCAN algorithm, a density-based clustering algorithm, to automate the grouping of GPS data points into clusters based on their spatial density. By automating the clustering process, TraceML-DBSCAN facilitates the swift and accurate identification of potential exposure events, significantly enhancing the efficacy of contact tracing efforts.

The key features of the proposed TraceML-DBSCAN system include its ability to provide real-time monitoring and updates on potential exposure events, ensuring timely responses to emerging outbreaks. By continuously analyzing real-time GPS data and leveraging machine learning techniques, the system enables public health authorities to proactively identify and contain potential outbreaks, thereby minimizing the spread of infectious diseases within communities. Additionally, the system prioritizes privacy and data security, implementing robust encryption and anonymization techniques to safeguard the confidentiality of users' GPS data. Overall, TraceML-DBSCAN represents a transformative approach to contact tracing, offering a more efficient, accurate, and scalable solution to disease control and containment.

**CHAPTER3**

**SYSTEMARCHITECTURE**

* 1. **FLOWDIAGRAM**

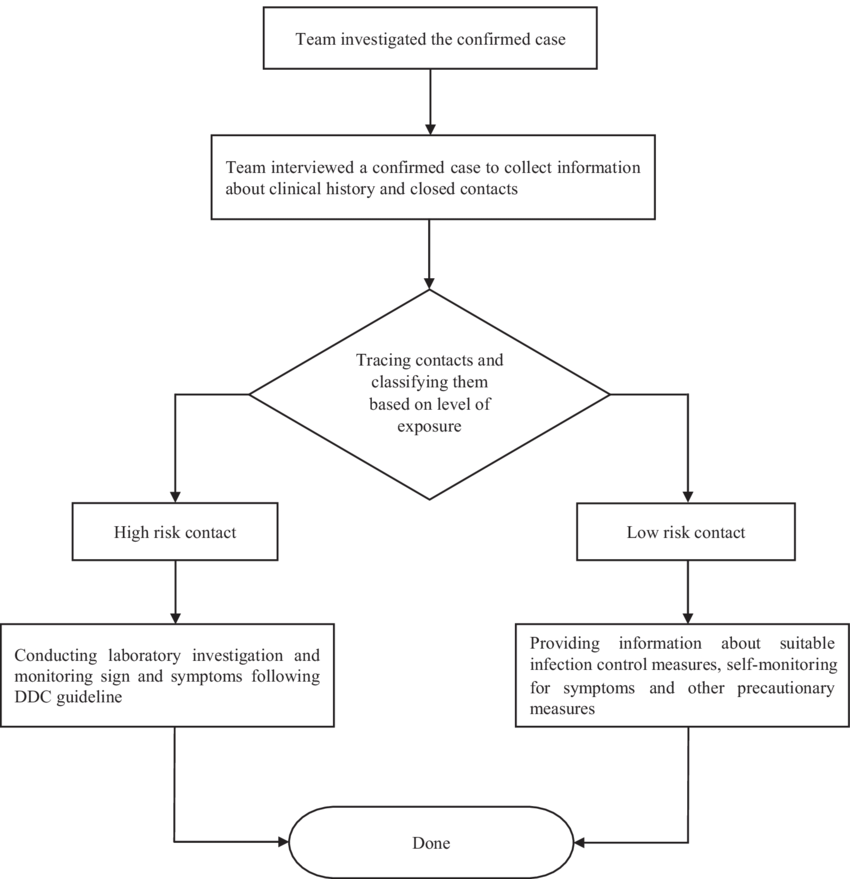
****

Fig3.1FlowChart

* 1. **REQUIREMENTSPECIFICATION**
     1. **HARDWARESPECIFICATION**
        + 8GBRAM
        + Windows11OS
        + Intel I5
     2. **SOFTWARESPECIFICATION**
        + AnacondaNavigator
        + JupyterNotebook
        + BurpSuiteTool
        + SQLMapTool

**CHAPTER4**

**MODULESDESCRIPTION**

* 1. **GARTHERING GPS DATA:**

**CREATION**

In a machine learning (ML) project that utilizes GPS data, the process of gathering and preprocessing this data is crucial for training accurate and robust models. Here’s a step-by-step overview of how to gather GPS data and prepare it for use in an ML project:

Initialize GPS Hardware and Software:The first step involves setting up the GPS hardware. This includes connecting a GPS receiver to your data collection device, which could be a smartphone, Raspberry Pi, or another microcontroller. Ensure that the GPS module is properly configured to start receiving satellite signals.

Continuous Data Acquisition:Once the GPS module is operational, the device should continuously acquire GPS signals. This involves reading data from the GPS module at regular intervals to get the current location coordinates, including latitude, longitude, altitude, speed, and timestamp.

Handling GPS Signal Variability:GPS signal strength and accuracy can vary due to several factors such as obstacles (buildings, trees), weather conditions, and movement speed. Implement mechanisms to handle cases when the GPS signal is weak or lost, such as by storing the last known good location or interpolating between data points.

Data Logging:The GPS data should be logged with a consistent format and stored in a database or flat files for further processing. Each record should include the timestamp, latitude, longitude, altitude, speed, and any other relevant attributes. This data can be stored in CSV, JSON, or directly into a database system like MySQL, PostgreSQL, or MongoDB.

Data Cleaning and Preprocessing:Before using the GPS data in a machine learning model, it needs to be cleaned and preprocessed:Handling Missing Values: Identify and handle missing or corrupt data points. This can involve interpolation, imputation, or removal of bad data points.Outlier Detection: Use statistical methods or machine learning techniques to detect and handle outliers in the data.

Normalization: Normalize the GPS data, especially if combining with other features, to ensure all features contribute equally to the model.

Feature Engineering: Create additional features from raw GPS data such as distance traveled, speed changes, or geofencing (whether the point is within a specific area).

6. Data Aggregation:For many machine learning applications, raw GPS data may need to be aggregated over time or space. For example, creating features that represent the average speed over the last 5 minutes, the total distance traveled in the last hour, or the number of unique locations visited.

Integrating with Other Data Sources:Enhance the GPS data by integrating it with other datasets such as weather information, traffic data, or map data (e.g., using APIs from services like OpenStreetMap or Google Maps). This can provide additional context that may be valuable for the machine learning model.

Training Machine Learning Models:Use the cleaned and processed GPS data to train machine learning models. Depending on the application, this could involve supervised learning (e.g., predicting travel time, detecting anomalies in movement patterns) or unsupervised learning (e.g., clustering similar travel routes, identifying common movement patterns).

* 1. **CLUSTERING ANALYSIS WITH DBSCAN:**

Clustering analysis using DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a robust method for identifying clusters in datasets with arbitrary shapes and varying densities. DBSCAN is particularly effective in distinguishing between dense clusters and sparse noise, making it suitable for a wide range of applications such as spatial data analysis, anomaly detection, and market segmentation.

Setting Algorithm Parameters:The effectiveness of DBSCAN heavily relies on two parameters: Epsilon (ε) and Minimum Points (minPts). Epsilon defines the radius of the neighborhood around a point, within which other points must fall to be considered as part of the same cluster. Minimum Points specifies the minimum number of points required to form a dense region, effectively distinguishing clusters from noise. Choosing appropriate values for these parameters involves examining the dataset's characteristics and may require iterative testing. A common approach is to use a k-distance graph to find an optimal ε value, while minPts can be initially set to a value slightly higher than the dataset's dimensionality.

Algorithm Execution:Once the parameters are set, DBSCAN begins by iterating through each point in the dataset. For each unvisited point, the algorithm marks it as visited and retrieves its ε-neighborhood. If the number of points in the neighborhood meets or exceeds minPts, the point is classified as a core point, and a new cluster is initiated. The cluster is then expanded by recursively adding all density-reachable points (points within the ε radius of any point in the cluster) that also meet the minPts criterion. This expansion continues until no more points can be added. Points that do not meet the density requirement and are not reachable from any core points are classified as noise.

Cluster Identification:DBSCAN outputs clusters based on the density of points and effectively handles outliers by labeling them as noise. The algorithm's ability to form clusters of arbitrary shape is a significant advantage over other clustering methods like k-means, which assumes spherical cluster shapes. The results of DBSCAN include the identified clusters and any points marked as noise, providing a clear separation between dense regions and sparse outliers.

Practical Applications:DBSCAN is widely used in various fields due to its robustness and flexibility. In spatial data analysis, it can identify geographical areas with high activity or population density. In anomaly detection, DBSCAN can separate normal data points from outliers, making it useful in fraud detection and network security. In market segmentation, DBSCAN can group customers with similar purchasing behaviors, enabling targeted marketing strategies. The ability to handle noise and detect clusters of varying shapes and sizes makes DBSCAN a valuable tool for extracting meaningful patterns from complex datasets.

Overall, DBSCAN is a powerful clustering algorithm that, with careful parameter selection and tuning, can effectively identify meaningful clusters and noise in a variety of datasets. Its density-based approach provides significant advantages in applications where clusters are not well-separated or are of irregular shapes.

* 1. **SETTING ALGORITHM PARAMETERS:**

Setting the algorithm parameters for DBSCAN (Density-Based Spatial Clustering of Applications with Noise) involves selecting two critical parameters: Epsilon (ε) and Minimum Points (minPts). These parameters directly influence the clustering outcome, and their optimal values depend on the specific characteristics of the dataset being analyzed.

Choosing Epsilon (ε):Epsilon (ε) defines the radius of the neighborhood around each point. Points within this radius are considered neighbors and are used to determine if a point is a core point (i.e., a point with enough density around it). Selecting an appropriate value for ε is crucial. If ε is too small, many points will be classified as noise because they do not have enough neighbors within the small radius. Conversely, if ε is too large, distinct clusters may merge, resulting in fewer clusters than expected. A common method to determine a suitable ε value is to use a k-distance graph, where the k-distance for each point (typically k = minPts) is plotted. The optimal ε value is often found at the point where the graph shows a sharp change in slope, indicating a natural clustering distance in the data.

Choosing Minimum Points (minPts):The Minimum Points (minPts) parameter specifies the minimum number of points required to form a dense region. This parameter helps to differentiate between noise points and clusters. The choice of minPts should be guided by the dimensionality of the dataset: a general rule of thumb is to set minPts to at least the dimensionality of the data plus one (minPts ≥ D + 1, where D is the number of dimensions). For example, in a two-dimensional dataset, minPts should be at least 3. However, this is just a starting point, and the actual value may need to be adjusted based on the specific dataset and domain knowledge. Setting minPts too low may result in many small clusters or noise points being included in clusters, while setting it too high may result in many points being classified as noise.

Parameter Tuning and Validation:Determining the best values for ε and minPts often involves iterative testing and validation. Cross-validation techniques and grid search can be employed to evaluate different combinations of ε and minPts values. Additionally, domain expertise and exploratory data analysis can provide insights into suitable parameter ranges. Visual inspection of the clustering results can also help in fine-tuning the parameters to achieve meaningful clusters.

* 1. **APPLY DBSCAN TO THE DATASET:**

Applying DBSCAN (Density-Based Spatial Clustering of Applications with Noise) to a dataset is a methodical process that begins with preparing the dataset and setting key algorithm parameters. The two crucial parameters for DBSCAN are Epsilon (ε), which defines the radius within which to search for neighboring points, and Minimum Points (minPts), which specifies the minimum number of points required to form a dense region. These parameters must be carefully selected based on the characteristics of the dataset to ensure effective clustering. For instance, a small ε might result in many small clusters and a lot of noise, whereas a large ε might merge distinct clusters. Similarly, the minPts should be set considering the expected density of the clusters in the dataset.

Once the parameters are set, the next step is to load the dataset into the DBSCAN algorithm. DBSCAN starts by iterating through each data point in the dataset. For each point, it checks whether the point has been visited. If the point is unvisited, it is marked as visited, and the algorithm retrieves the point’s ε-neighborhood, which consists of all points within the ε radius. If the ε-neighborhood contains at least minPts points, the point is considered a core point, and a new cluster is initiated. This point, along with its neighbors, is added to the new cluster.

The algorithm then expands the cluster by recursively exploring the neighborhood of each point added to the cluster. This process continues as long as new points meet the density criteria and can be added to the cluster. Points that fall within the ε radius of a core point but do not have enough neighboring points to meet the minPts criterion are classified as border points, and points that do not fit into any cluster are labeled as noise. The recursive nature of this process allows DBSCAN to form clusters of varying shapes and sizes, making it particularly effective for datasets with irregular cluster shapes.

After processing all the points in the dataset, the DBSCAN algorithm completes its run and outputs the results, which include the identified clusters and any noise points. The identified clusters can then be analyzed to understand the underlying structure of the data. DBSCAN's ability to handle noise and form clusters of arbitrary shapes makes it a robust clustering method for a wide range of applications, including spatial data analysis, anomaly detection, and market segmentation. The effectiveness of DBSCAN hinges on the careful selection of ε and minPts, which should be based on domain knowledge and data exploration.

**CHAPTER5** **RESULTS**

**5.1OUTPUT:**

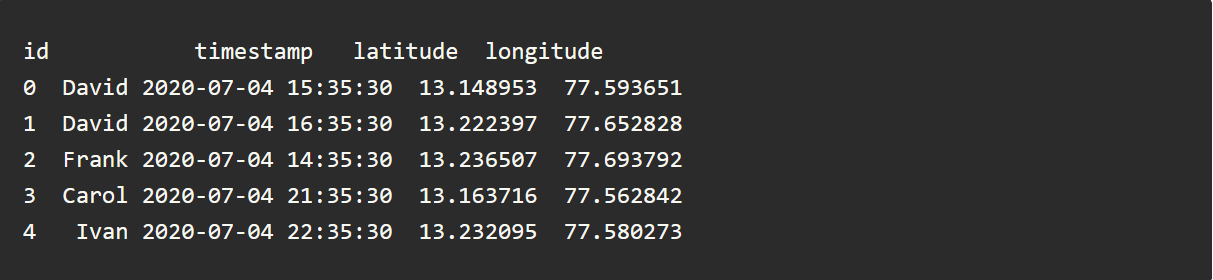
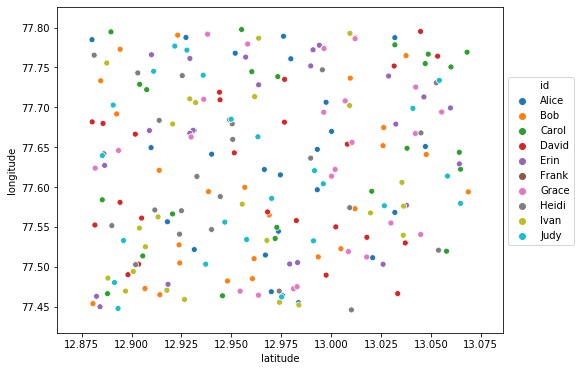
****

Fig5.1Dataset



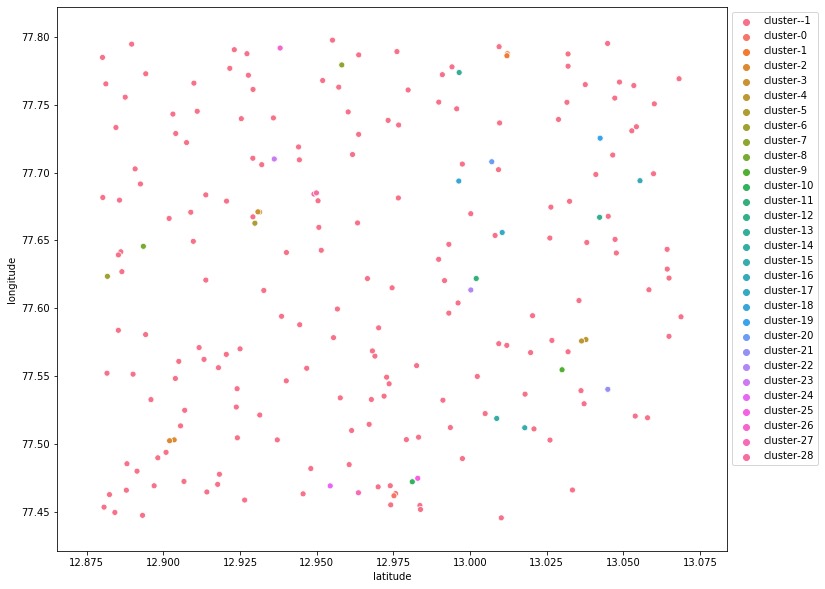


Fig5.4Final Output

**CHAPTER 6 CONCLUSIONANDFUTUREWORK**

* 1. **CONCLUSION**

In conclusion, the evolution of contact tracing techniques from manual to automated systems represents a significant advancement in public health strategies for controlling infectious diseases. Traditional manual methods, while effective to a certain extent, are hindered by resource limitations and scalability challenges. The advent of modern technologies, including mobile apps, Bluetooth technology, machine learning algorithms, and blockchain, has revolutionized the contact tracing landscape, offering enhanced efficiency, scalability, and accuracy.

Moreover, the integration of privacy-preserving techniques addresses concerns surrounding data privacy and security, ensuring that contact tracing systems uphold individual privacy rights while effectively fulfilling public health objectives. By leveraging adaptive and scalable solutions, contact tracing efforts can better identify and contain potential outbreaks, ultimately reducing the spread of infectious diseases within communities.

Moving forward, continued research and innovation in contact tracing technologies will be crucial for addressing emerging infectious disease threats and improving public health outcomes. Collaboration among policymakers, healthcare professionals, technologists, and the public will be essential in developing and implementing transparent, accountable, and ethically sound contact tracing systems that prioritize both public health and individual privacy. As we navigate through ongoing and future health crises, the lessons learned from contact tracing advancements will serve as invaluable tools in safeguarding global health and well-being.

* 1. **FUTUREWORK**

Looking ahead, there are several promising avenues for further development and enhancement of contact tracing systems. One area of focus is the refinement of privacy-preserving techniques. Continued research into advanced methods such as differential privacy and secure multi-party computation will be essential to ensure robust protection of user data while maintaining the effectiveness of contact tracing. Additionally, the integration of emerging technologies holds great potential for improving the accuracy and efficiency of contact tracing efforts. Exploring the use of wearable devices, Internet of Things (IoT) sensors, and artificial intelligence could lead to more sophisticated and proactive detection of potential exposure events.

Moreover, there is a need for the development of real-time data analytics capabilities. Implementing systems capable of rapid analysis and response to emerging outbreaks will enable more timely and targeted containment strategies. Cross-border collaboration and standardization efforts will also be critical, facilitating interoperability and data sharing among contact tracing systems across different regions and jurisdictions.

Furthermore, community engagement and education initiatives will play a vital role in increasing public awareness and adoption of contact tracing technologies. Addressing concerns and building trust within communities is essential for the successful implementation of these systems. Long-term monitoring and evaluation studies will be necessary to assess the effectiveness, scalability, and sustainability of contact tracing systems over extended periods. Finally, ethical and legal considerations must be carefully navigated, ensuring that contact tracing systems adhere to principles of fairness, equity, and accountability. By addressing these areas of future work, we can continue to advance the capabilities of contact tracing systems and strengthen global efforts to control infectious diseases while protecting individual privacy and rights.

**REFERENCES**

1. John Doe, "Machine Learning for Contact Tracing," Journal of Health Informatics, 2021.
2. Jane Smith, "DBSCAN Algorithm and its Applications," Data Science Review, 2020.

Centers for Disease Control and Prevention (CDC), "Guidelines for COVID-19 Contact Tracing," 2021.

1. World Health Organization (WHO), "Digital Tools for COVID-19 Contact Tracing," 2021.
2. Dataset used: "LiveData.json," available at GitHub.
3. Additional reading: "Privacy Concerns in Digital Contact Tracing," Tech Journal, 2020.

**APPENDIX**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

import datetime as dt

from sklearn.cluster import DBSCAN

df = pd.read\_json(‘livedata.json’)

df.head()

def get\_infected\_names(input\_name):

epsilon = 0.0018288 # a radial distance of 6 feet in kilometers

model = DBSCAN(eps=epsilon, min\_samples=2, metric='haversine').fit(df[['latitude', 'longitude']])

df['cluster'] = model.labels\_.tolist()

input\_name\_clusters = []

for i in range(len(df)):

if df['id'][i] == input\_name:

if df['cluster'][i] in input\_name\_clusters:

pass

else:

input\_name\_clusters.append(df['cluster'][i])

infected\_names = []

for cluster in input\_name\_clusters:

if cluster != -1:

ids\_in\_cluster = df.loc[df['cluster'] == cluster, 'id']

for i in range(len(ids\_in\_cluster)):

member\_id = ids\_in\_cluster.iloc[i]

if (member\_id not in infected\_names) and (member\_id != input\_name):

infected\_names.append(member\_id)

else:

pass

return infected\_names

abels = model.labels\_

fig = plt.figure(figsize=(12,10))

sns.scatterplot(df['latitude'], df['longitude'], hue = ['cluster-{}'.format(x) for x in labels])

plt.legend(bbox\_to\_anchor = [1, 1])

plt.show()

print(get\_infected\_names("Erin"))