Clustering and Labeling of a V2V Communication Dataset

based on CAM and DENM Messages with Malicious Data Injection

Analysis and Implications

ALESSANDRA BLASIOLI - PROJECT WORK

INTRO

VEHICULAR COMMUNICATION SYSTEMS

Computer networks where vehicles and roadside units (RSU) serve as communicating nodes

VEHICLE-TO-VEHICLE COMMUNICATION (V2V)

Wireless information exchange about the speed and position of nearby vehicles.
Offering great promise in accident prevention.

INTRO

DATASETS

Two different types of messages: Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Messages (DENM).

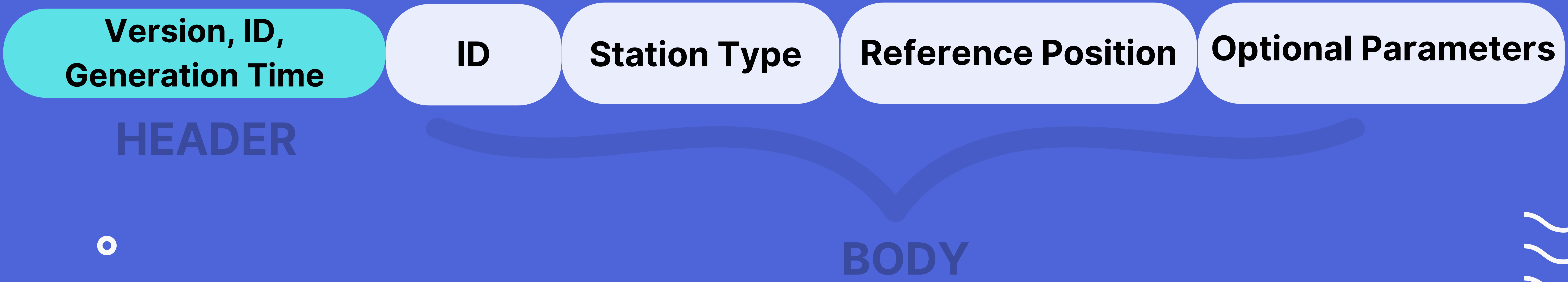
OBJECTIVE OF THE WORK

Data preprocessing, clustering, introduce noise into the provided dataset to simulate potential malicious reports, labeling the data based on clustering and identifying potential outliers.

CAM MESSAGES

defined by the European Telecommunications Standards Institute (ETSI) in 2011

- Basic awareness service by sending status data to **nearby nodes**
- Distributing messages about **presence**, **location**, and **fundamental status**



Version, ID, Generation Time

ID

Station Type

Reference Position

Optional Parameters

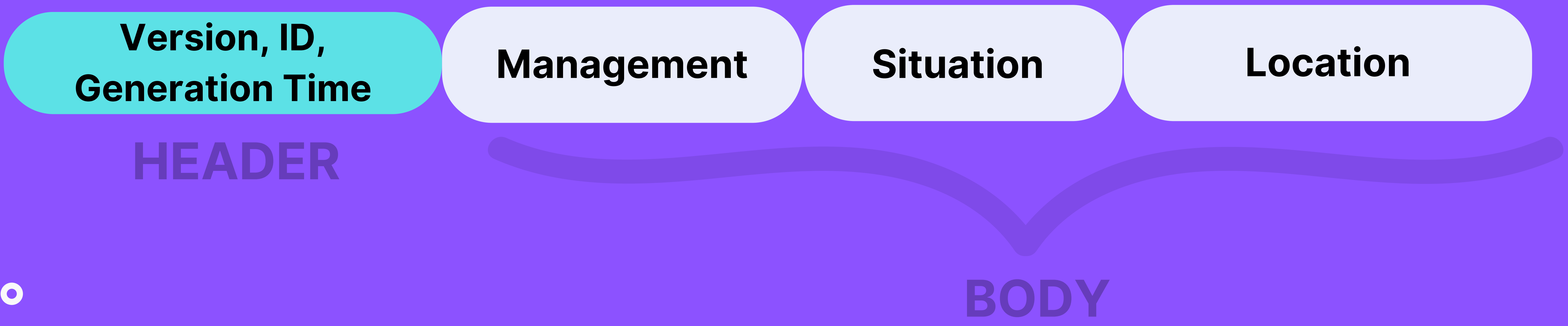
HEADER

BODY

DENM MESSAGES

defined by the European Telecommunications Standards Institute (ETSI) in 2011

- Notification service regarding **road status**
- Support active road **safety** applications



Version, ID, Generation Time | **Management** | **Situation** | **Location**

HEADER

BODY

STEP 1: CLUSTERING

X-Means Algorithm

A **variant of K-Means algorithm**, determines **automatically** the optimal number of clusters in the data without requiring a predefinited specification, based on **recursion**.

STEP 1: CLUSTERING

X-Means Algorithm

- Initially applies **standard K-Means** with an initial cluster count
- Assesses **clustering quality using a measure** like Sum of Squared Errors (SSE)
- Checks if **splitting clusters** improves overall quality
- Uses criteria like Akaike Information Criterion (AIC) for significant improvements
- **Divides clusters** with K-Means if advantageous

STEP 1: CLUSTERING

X-Means Algorithm

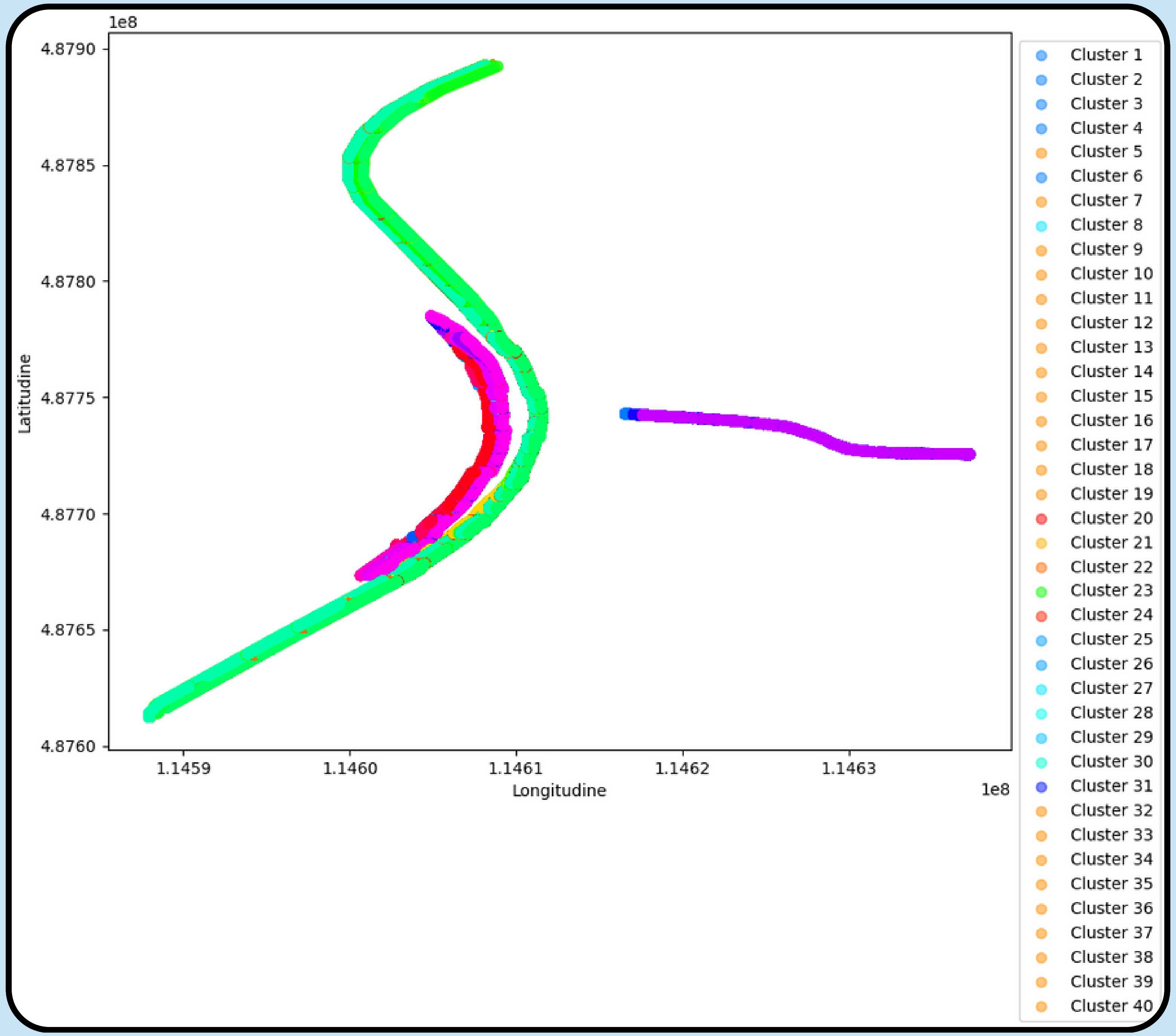
- Repeats **division and evaluation for existing clusters** and allows potential creation of new sub-clusters for improved clustering
- **Stops** when **no further cluster division** is possible; halts if the division **doesn't significantly enhance results** compared to complexity
- Returns **optimal clusters determined automatically**

STEP 1: CLUSTERING

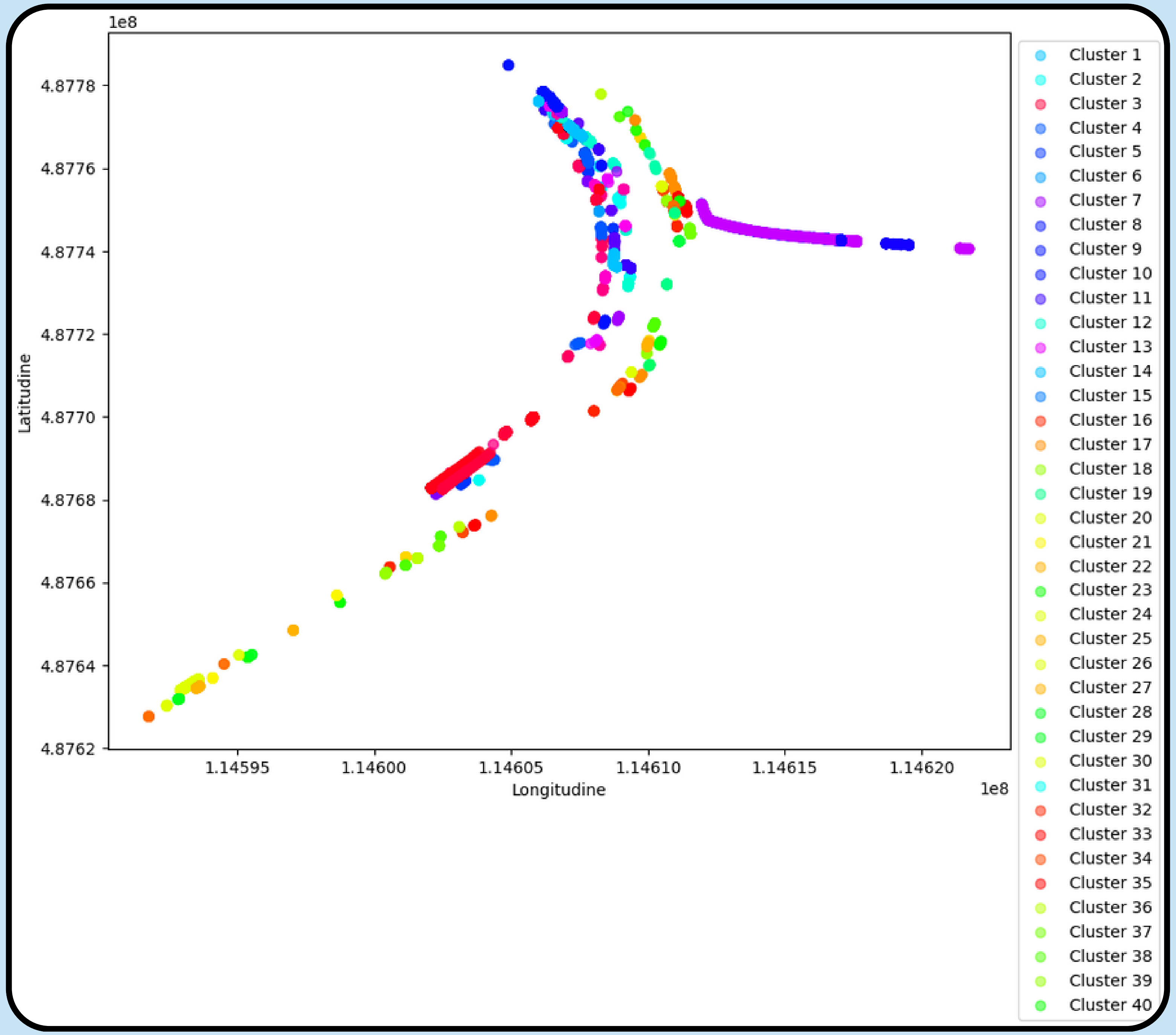
X-Means Algorithm

- **Longer computational time** compared to traditional K-Means due to iterative nature
- Suitable for **specific objectives** without requiring a predefined cluster count
- Part of ***pyclustering*** open-source library

STEP 1:
C



Clustering of the provided CAM dataset

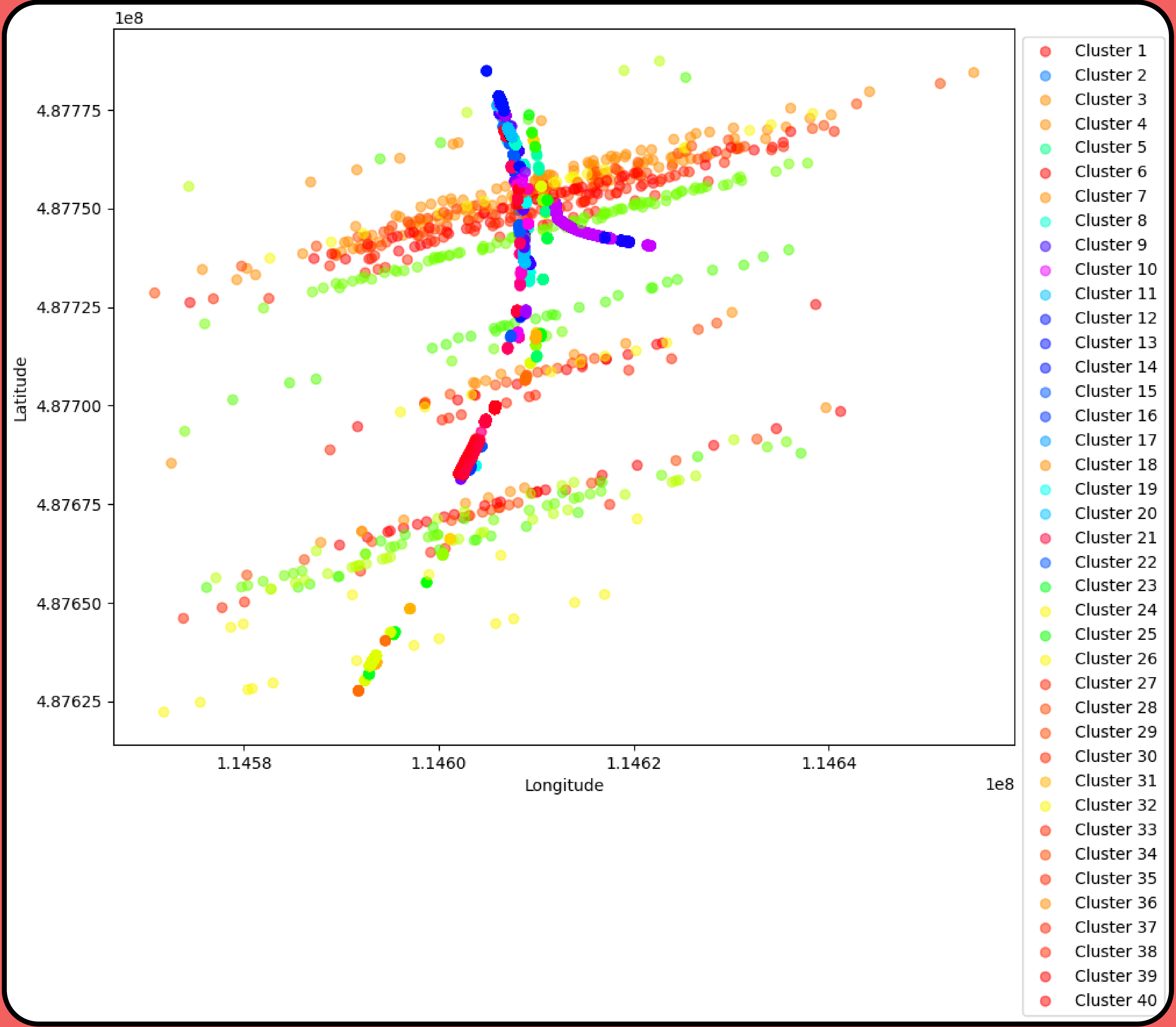


Clustering of the provided DENM dataset

STEP 2: DATA CONTAMINATION

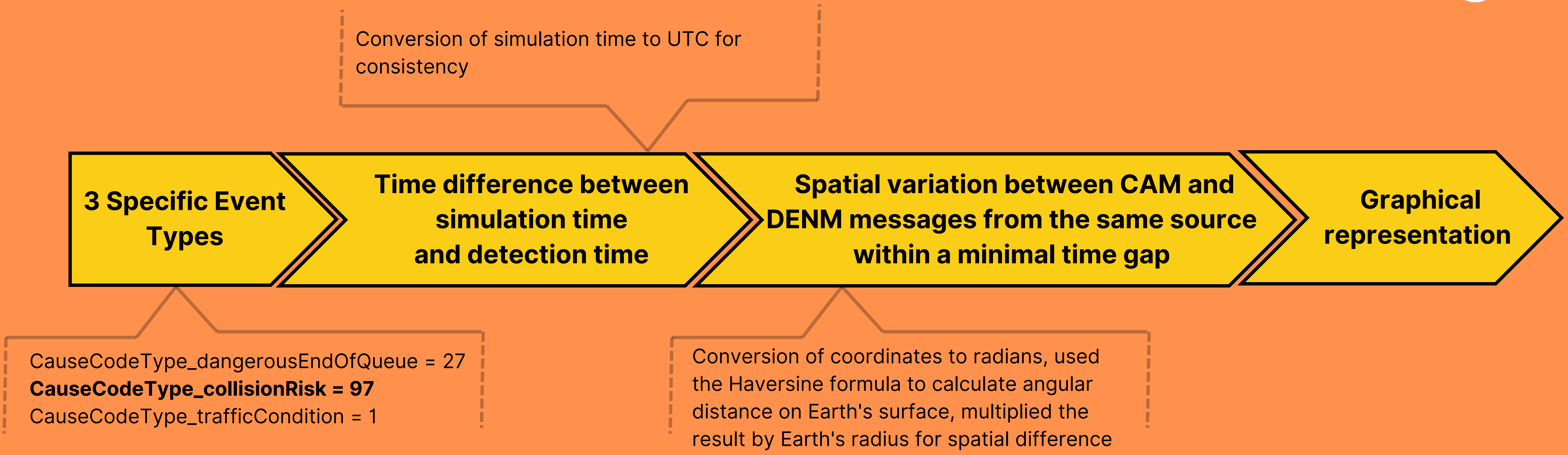
- Varied coordinates using a **Gaussian distribution** (mean: 0, standard deviation: 1) multiplied by a factor of 100
- Kept variations within **a range of approximately 100 to 900 meters on the map**, considering degree-based coordinates
- Contaminated data based on the number of sources to simulate malicious vehicles
- Focused contamination on **sources with eventType value 97**, representing the most significant cluster
- Contaminated **20% (8 sources) of this specific eventType**

STEP 2: DATA C



Clustering Following Data Contamination

STEP 3: EVENT TYPE ANALYSIS



3 Specific Event Types

CauseCodeType_dangerousEndOfQueue = 27
CauseCodeType_collisionRisk = 97
CauseCodeType_trafficCondition = 1

Time difference between simulation time and detection time

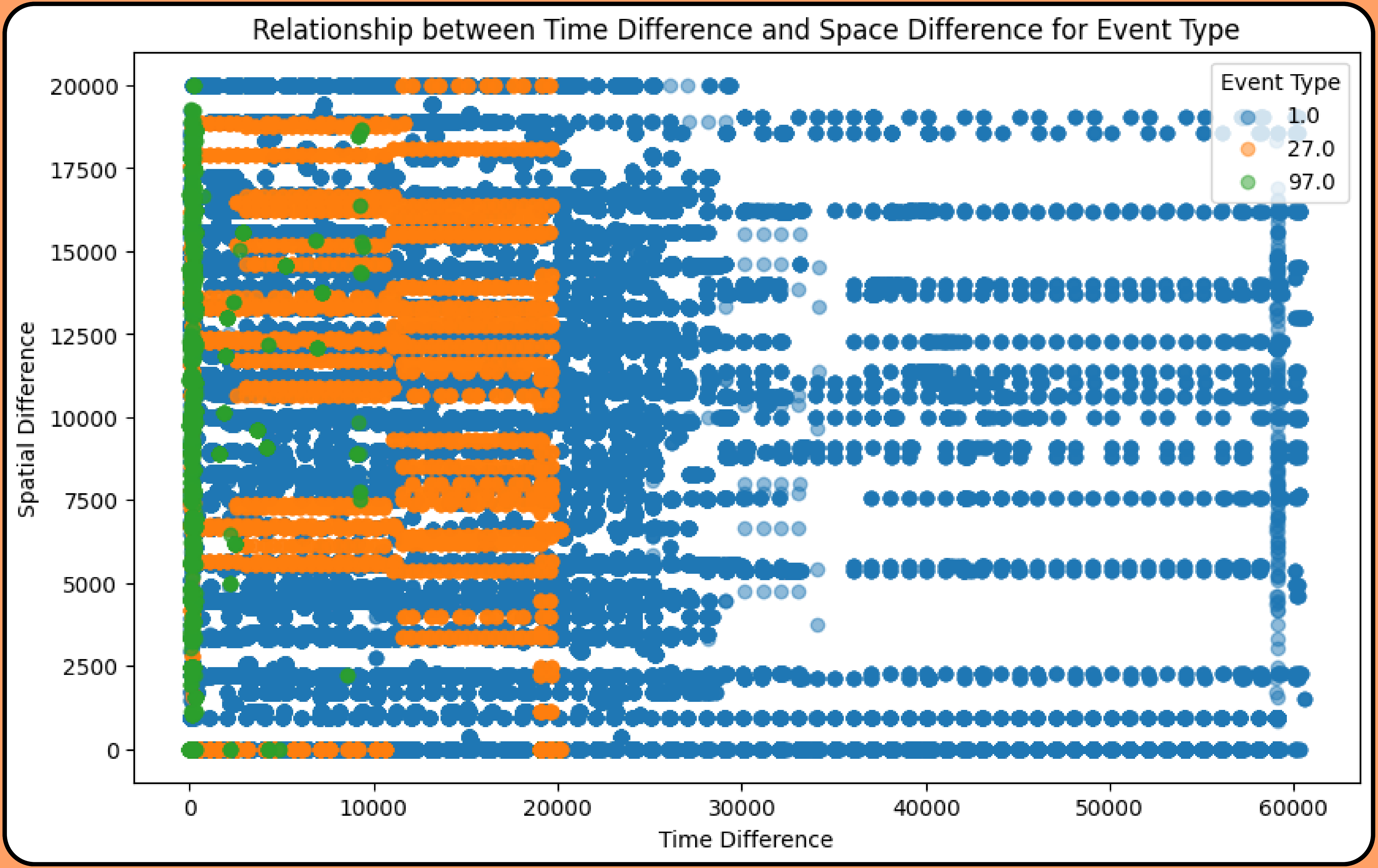
Conversion of simulation time to UTC for consistency

Spatial variation between CAM and DENM messages from the same source within a minimal time gap

Conversion of coordinates to radians, used the Haversine formula to calculate angular distance on Earth's surface, multiplied the result by Earth's radius for spatial difference

Graphical representation

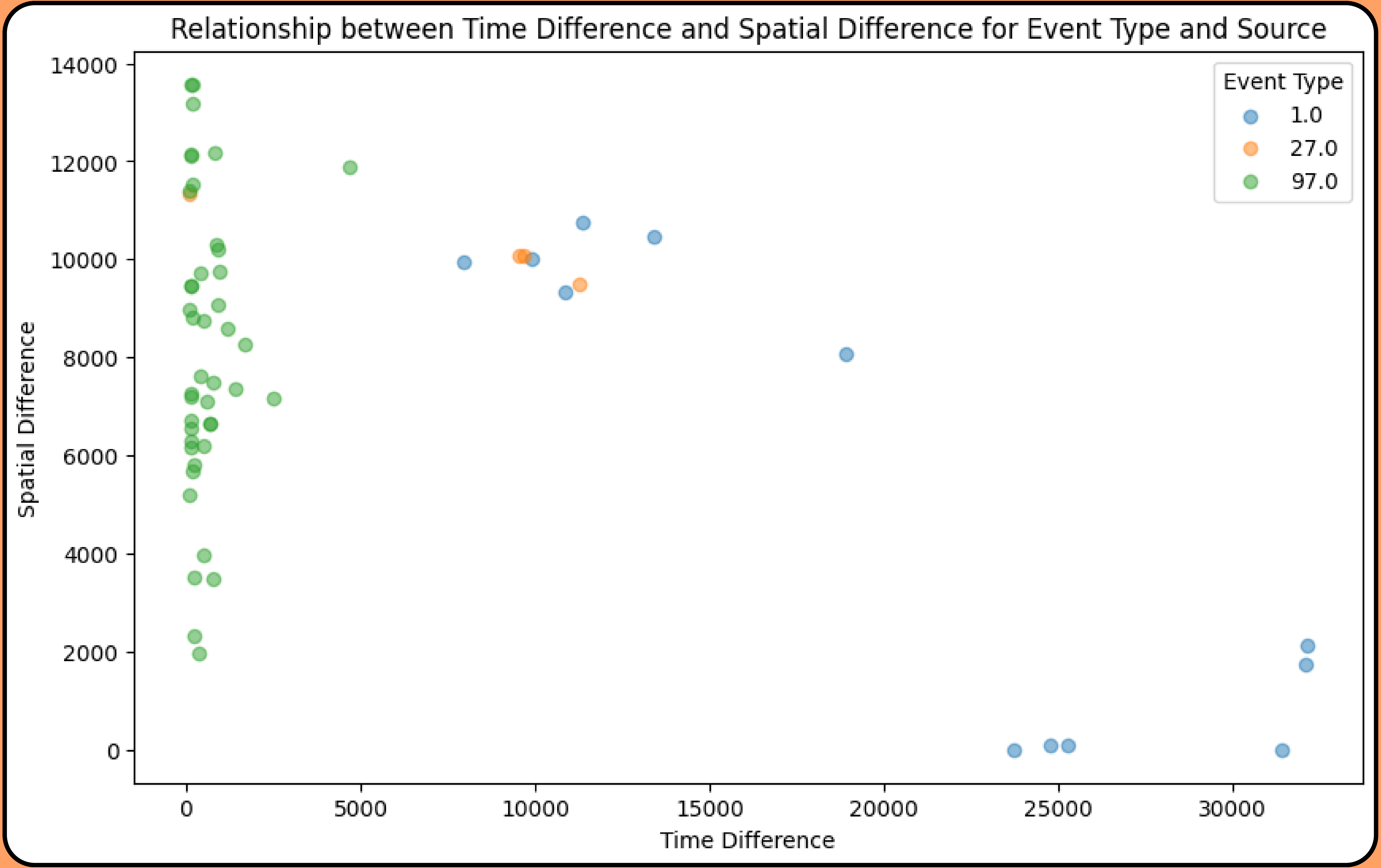
STEP 3: EVENT TYPE
A



Ca
Ca
Ca

STEP 3: EVENT TYPE
A

Ca
Ca
Ca



STEP 4: OUTLIER DETECTION

- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) **algorithm**

⟶ Groups points based on the data density in space

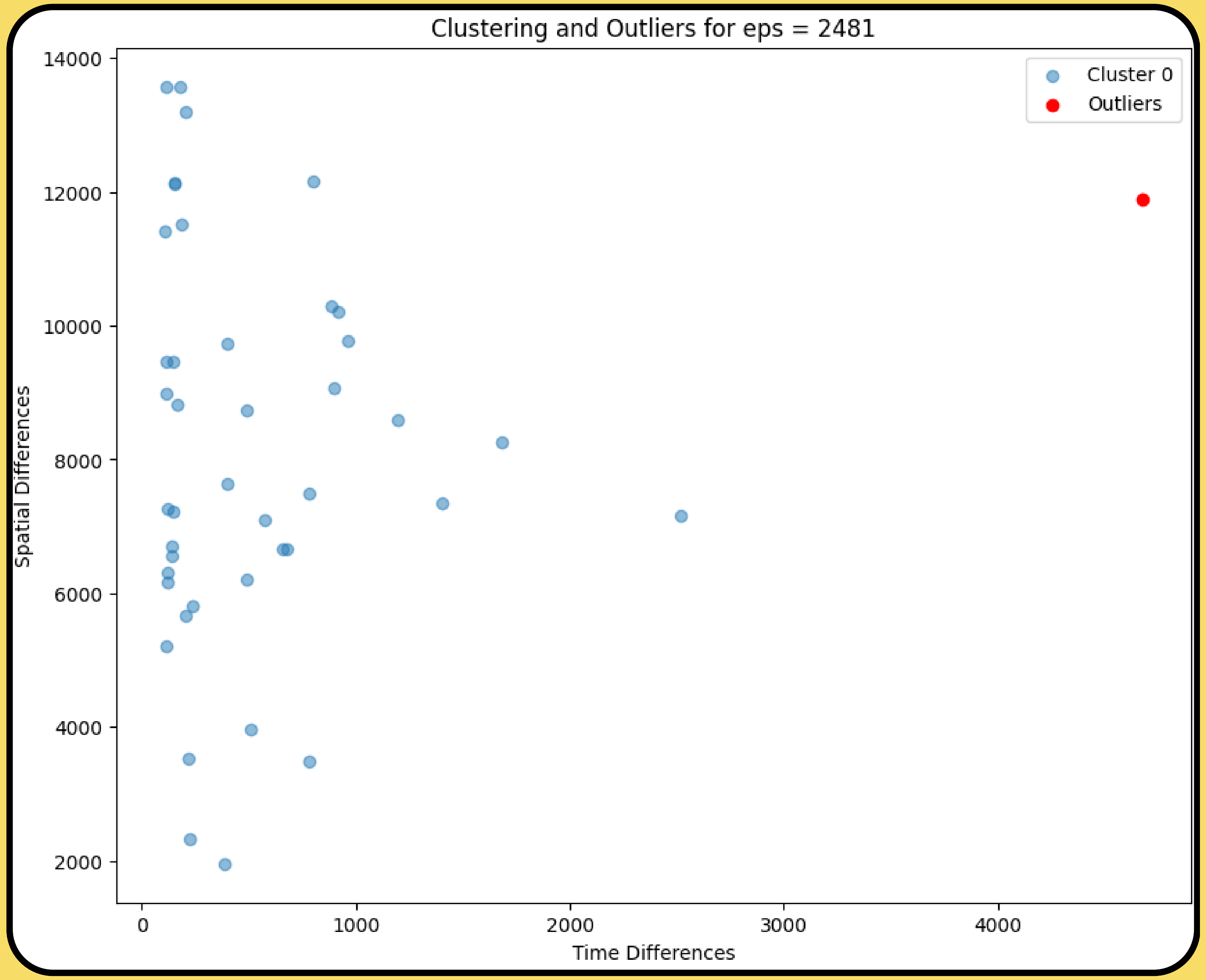
⟶ Two parameters:

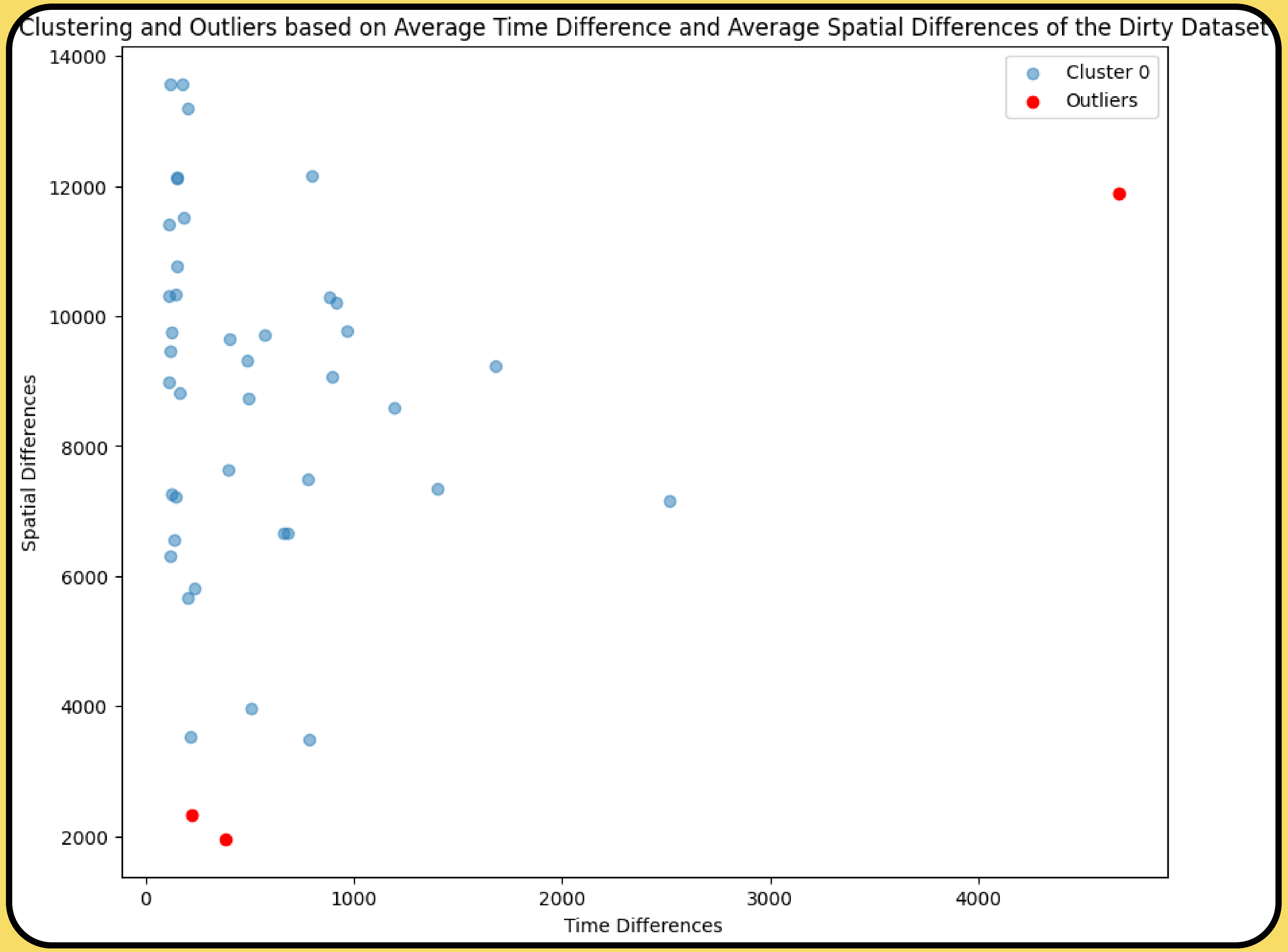
  - **epsilon (ε)**, the maximum distance between two points to consider them part of the same cluster;
  - **minPoints**, the minimum number of points required to form a cluster.

- Experiment varying **epsilon values** to find the optimal configuration

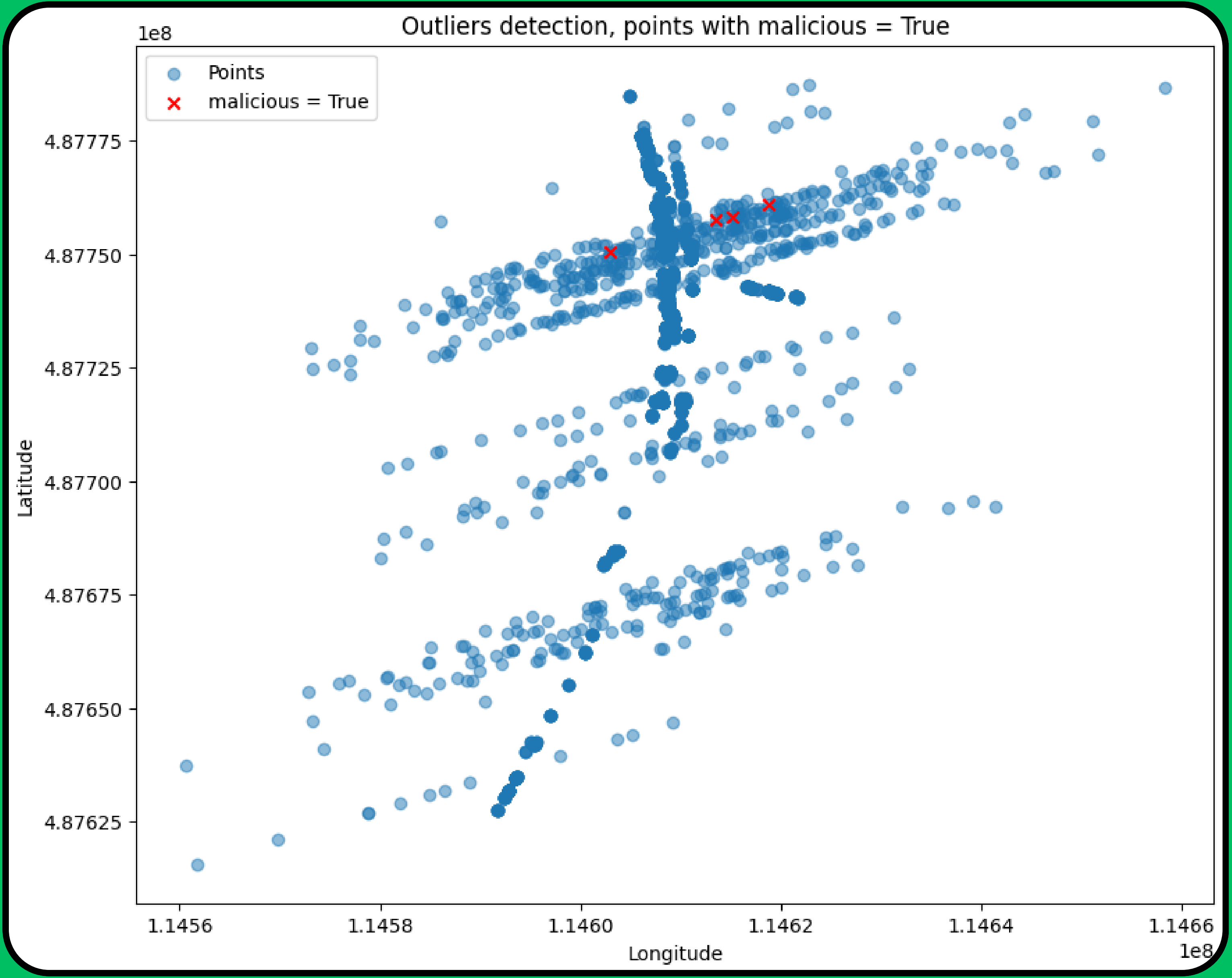
⟶ The best epsilon is the fewest possible outliers, the data we are working with is clean!

STEP 4: OUTLIER D

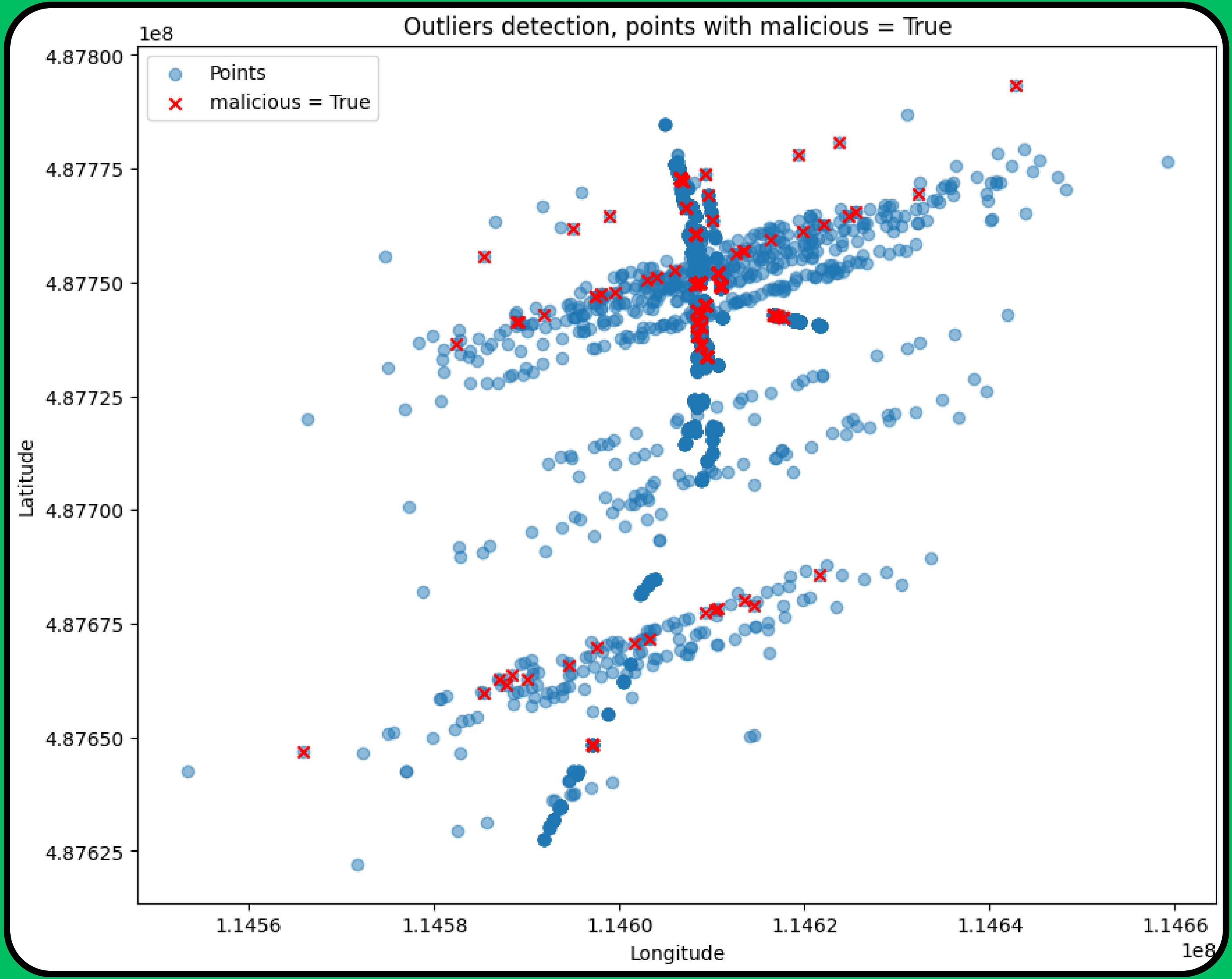




STEP 5: LABELING



Outliers detection for epsilon = 2481 with the dirty dataset



Outliers detection for epsilon = 1281 with the dirty dataset

CONCLUSIONS

Algorithm’s actual efficacy: ability to discern the spatiotemporal differences between CAM and DENM messages. It **can, based on messages sent within a similar time and space, distinguish potential malicious ones**. In a real-world application, this tool would prove efficient.

An attacker might lack precision, **unaware of such analytical tools**, thereby **transmitting messages from a distant space or time** for a specific event, **promptly flagged by the algorithm**.

Thanks for the attention!

ALESSANDRA BLASIOLI - PROJECT WORK