IRSTI 50.47.02 https://doi.org/10.26577/jpcsit202541

Bekarys Nurzhaubaev[1], Nursultan Nyssanov[1], Alisher Batkuldin[1], Ali Myrzatay[1,2*], Murat Zhakenov[3]

[1]Research and Innovation Center "CyberTech", Astana IT University, Astana, Kazakhstan
[2] "Digital government support center" RSE on the REM, Astana, Kazakhstan
[3] "Digital Heritage of Eurasia" LLP, Astana, Kazakhstan
*e-mail: mirzataitegiali@gmail.com

NEW AUTONOMOUS SYSTEM FOR SPATIOTEMPORAL CLUSTERING AND VISUALIZATION OF DEVICE TRAJECTORIES IN FORENSIC INVESTIGATIONS

Abstract. This study presents «trajectory_analyzer», a Python-based system designed for the forensic analysis and visualization of geolocation data extracted from mobile devices. With the increasing volume of spatial-temporal data collected from sources such as GPS, Wi-Fi, and image metadata, forensic professionals face growing challenges in structuring and interpreting mobility patterns. Existing solutions often lack flexibility, require supervised models, or depend on proprietary infrastructure. Our approach applies an unsupervised DBSCAN-based trajectory clustering method, temporal ordering, and a real-time web map interface to reveal behavioral insights without the need for manual labeling or cloud services. Compared to prior research, the system improves spatial accuracy, source transparency, and visual clarity. Experimental results show that the proposed clustering method successfully identifies movement clusters and transitions while maintaining full offline operability. However, this improvement comes at the expense of more local storage because of embedded map tiles. Overall, this work provides a practical, understandable, and independent foundation for investigators dealing with unstructured multi-source geolocation data.

Keywords: Digital forensics, Geolocation analysis, Trajectory clustering, Unsupervised learning, DBSCAN, GPS tracking, Offline tools.

1. Introduction

Nowadays, the volume and detail of data on movement trajectories has experienced a sharp growth spurt due to the widespread use of GPS-enabled mobile devices. The use and need for this technology has also grown in the field of automotive systems and surveillance. The resulting volume of spatiotemporal data opens up new possibilities in digital forensics, especially in the field of reconstruction and visualization of human movements for investigative purposes. However, working with such a volume of geolocation data remains a methodological problem [15].

The main challenge lies in the lack of effective tools with intuitive and scalable visualization, especially for massive and heterogeneous trajectory datasets. Existing approaches, including vector field analysis [1], skeletal trajectory classification [2], and institutional tracking systems [3] that often require structured environments, rely on supervised learning, or lack flexibility for processing multi-source device data.

Existing forensic and trajectory analysis tools often rely on cloud infrastructures, lack offline reproducibility, and offer limited integration of heterogeneous sources.

Our system addresses these gaps by providing (1) an autonomous modular architecture that works without internet access, (2) a unified JSON schema preserving data provenance, and (3) transparent clustering and visualization workflows reproducible in local environments.

Additionally, difficulties arise due to the content of the input data itself. Trajectory logs are often out of time order, contain uneven sampling, and are generated from various sources such as GPS, Wi-Fi scanning, EXIF image metadata [17], or communication timestamps. Although the DBSCAN clustering algorithm has shown promising results in extracting spatial structure from such data [4], they still require fairly careful parameter settings and rarely integrate well with visualization tools. Moreover, preprocessing remains a necessary but underdeveloped component in many forensic pipelines [5].

To address these challenges, this study introduces trajectory_analyzer, a robust, offline, and open-source framework designed for forensic analysis of geolocation data. It accepts input from various sources, performs automatic preprocessing, and extracts meaningful behavioral patterns through geometric clustering and temporal segmentation–without requiring any training data or manual annotation.

The core research question guiding this work is:

How can interpretable and meaningful mobility patterns be identified from irregular, noisy, and heterogeneous geolocation data without the use of supervised learning or manual labeling?

Presented research hypothesis is that such patterns can be effectively inferred by combining density-based clustering with chronological ordering. This allows us to detect frequently visited or significant locations (clusters), reconstruct transitions or movement routes between them, and analyze the role each data source plays in shaping the spatial resolution of the trajectory.

By merging unsupervised analysis with interactive, map-based visualization, this study provides forensic specialists with an intuitive toolset to reconstruct and interpret complex movement behaviors. This work aims to bridge the gap between raw geolocation logs and human-readable insights, especially in contexts where cloud-based solutions are impractical or inadvisable.

The main contributions of this study are as follows:

i. A modular offline system that enables autonomous forensic trajectory analysis without reliance on cloud infrastructures.

ii. A unified JSON data schema with source provenance, providing consistent integration of heterogeneous geolocation sources.

iii. Use of geodesic distance (Haversine) with the DBSCAN algorithm, including a reproducible procedure for selecting the ε and minPts parameters.

iv. Dynamic source-level filters for interactive selection and comparison of trajectory subsets across multiple data origins.

v. A reproducible offline-tile HTML report that combines spatiotemporal clustering, visual analytics, and statistical summaries in a portable format.

2. Literature Review

In recent years, the amount of mobile geolocation data in the form of GPS traces, Wi-Fi scans, as well as EXIF data in images and videos, has increased exponentially. This multimodal spatial-temporal data is a valuable resource in digital forensics, as it allows investigators to track user activity, locate sites that a user has been to, as well as match temporal activity with digital artifacts. However, the recent emergence of a wide range of sensors as well as formats makes these sources inconsistent in terms of sample rate, accuracy, as well as origin.

Nevertheless, despite the recent ubiquity of cloud-based analytical tools, in forensic applications, they continue to be hampered by issues of privacy, sovereignty of data, as well as chain-of-custody issues. This is primarily since cloud-based analysis could involve transmitting forensic data over cloud servers that could contravene confidentiality laws as well as instances that could interrupt the chain of custody of digital evidence. As a result, the need to develop fully offline systems that are able to integrate diverse sources of geolocation information has emerged as a burning concern in today's forensics.

As noted in [6], trajectory clustering remains a core component of GPS data analysis. However, the high dimensionality of raw geolocation logs presents difficulties for computational efficiency and human interpretation. In response, the study proposed various dimensionality reduction techniques in conjunction with DBSCAN-based clustering to improve processing speed and visual clarity. Yet, this process often requires domain-specific tuning and lacks generalized parameter estimation techniques.

In complementary efforts, researchers in [7] and [8] emphasized the need for preprocessing pipelines to clean and normalize GPS data before modeling. These studies outlined common artifacts such as signal drift, duplicate records, and inconsistent sampling intervals. Their solutions included interpolation techniques and network-based correction models. While technically sound, they often assume access to high-quality or real-time datasets, limiting their forensic application where data may be sparse or corrupted.

In the context of behavioral analysis, the study in [8] reviewed trajectory tracking systems in autonomous vehicles. The article underscored the importance of accurate localization, anomaly detection, and route prediction – all of which are translatable to forensic movement reconstruction. Meanwhile, [9] introduced a hardware-integrated edge computing GPS tracking platform. While promising for field deployments, its design prioritizes efficiency over flexibility, and its visual output remains rudimentary compared to forensic needs.

As outlined in [1], [11], and [14], the integration of geolocation and digital trace data into forensic cyber-physical investigations has become more prevalent. Their proposed tools focus on timeline reconstruction, correspondence analysis, and multi-source correlation. Although these interfaces support event sequencing, they are limited in their geospatial resolution and tend to lack interactive trajectory mapping features essential for field-level analysis.

Recent visualization frameworks have emerged in studies like [2] and [12], which applied vector field and density partitioning methods respectively. These offer macroscopic views of movement patterns in large datasets. However, their utility in forensic casework is restricted, as investigators often require micro-level insights – such as dwell times, visit frequencies, and source-specific behavior – which these approaches abstract away.

Studies [3] and [4] shifted the focus toward institutional and behavioral surveillance. The former evaluated crime scene classification based on skeletal trajectory analysis in surveillance settings, highlighting operational benefits and the potential for pattern recognition. The latter investigated staff perceptions and usability of GPS tagging in forensic psychiatric units, revealing gaps in data transparency and adaptability. Both studies confirm the growing reliance on geolocation data in controlled environments but underscore the absence of open systems for independent review or public domain research.

In [5], Yu et al. stressed the importance of pipeline robustness, advocating for modular preprocessing and clustering layers. Their work provided a foundation for reproducibility in GPS data workflows, though their system lacks integrated visualization or input flexibility. Similarly, [10] advanced stream-based clustering for trajectory segmentation, with real-time visualization capabilities. While scalable, such systems depend heavily on structured, continuous input – a luxury often unavailable in forensic scenarios.

Investigations into semantic and behavioral pattern extraction, such as those presented in [3], [11], and [13], move toward higher-level understanding of mobility and digital presence. These works proposed frameworks for detecting anomalies and identifying common routines across individuals. However, their reliance on annotated training data and machine learning infrastructure limits practical adoption in forensic workflows, which often operate with sparse and unlabeled datasets.

To summarize, past literature provides a diverse range of tools for geolocation analysis – from trajectory clustering and dimensionality reduction to stream processing and semantic modeling. However, many of these are either too abstract for forensic application, too rigid in data input requirements, or too opaque for field investigators. In our research, we address these gaps by combining the modularity of unsupervised clustering [6], the preprocessing awareness from [5], and the visual transparency from [1], [2]. Our trajectory_analyzer system offers an accessible, offline platform that includes clustering, timeline filtering, and map-based visualization in a single package inspired by principles introduced in [3] and [11].

3. Materials and Methods

3.1 Data Structure and Preprocessing

This module is designed to work on devices with a large amount of memory (e.g., SSD, HDD based systems and mobile devices) and functions offline to ensure reproducibility, transparency of forensic examination results and security. It collects a wide range of spatial and temporal types of tags and a wide range of available data types: GPS logs from devices, metadata obtained from scanning Wi-Fi access points, EXIF geotags embedded in photos and videos, from call and chat history, and system timestamps.

For greater clarity of the logic of the module, a block diagram of the system was developed, shown as Figure 1, reflecting the main stages from loading input data to generating a report.

This diversified approach reflects the increasing complexity of digital movement data, with location-related data often scattered across multiple sensors and applications. As noted in a previous study [1], using GPS data exclusively can lead to incomplete and biased reconstructions of situations, especially indoors or in places with a difficult signal. Thus, our module is designed to support and work with combined data from multiple input streams, to support more comprehensive and detailed trajectory modeling.

After extracting the data from another module, the collected information is combined into a single structured format. The data set is stored in a JSON object with the key "trajectory_points", where each record represents a discrete space-time observation. Each observation contains four main fields:

1. Timestamp in ISO 8601 format (e.g., "2023-12-10T12:42:00Z");

2. Coordinates specified as latitude and longitude in decimal degrees;

3. Source, a categorical label indicating the origin of the record (e.g.," GPS", "Wi-Fi", "image", "received").

The implementation of recording incoming data was partially shown in Figure 2. This format is designed to handle heterogeneity while preserving provenance and temporal integrity two critical dimensions in forensic casework. Sensor provenance allows analysts to filter or weigh observations by reliability, while accurate temporal ordering enables timeline reconstruction, path tracing, and behavioral segmentation.

Figure 2 illustrates the required structure and composition of the input data in JSON format used by the system. Each record follows the schema described above, including the three mandatory attributes: timestamp, coordinates, and source, ensuring interoperability across heterogeneous inputs. All coordinates are expressed in the WGS-84 coordinate reference system. The example also demonstrates how the system parses these records through the data loading routine, where each entry is converted into Python objects for further processing.

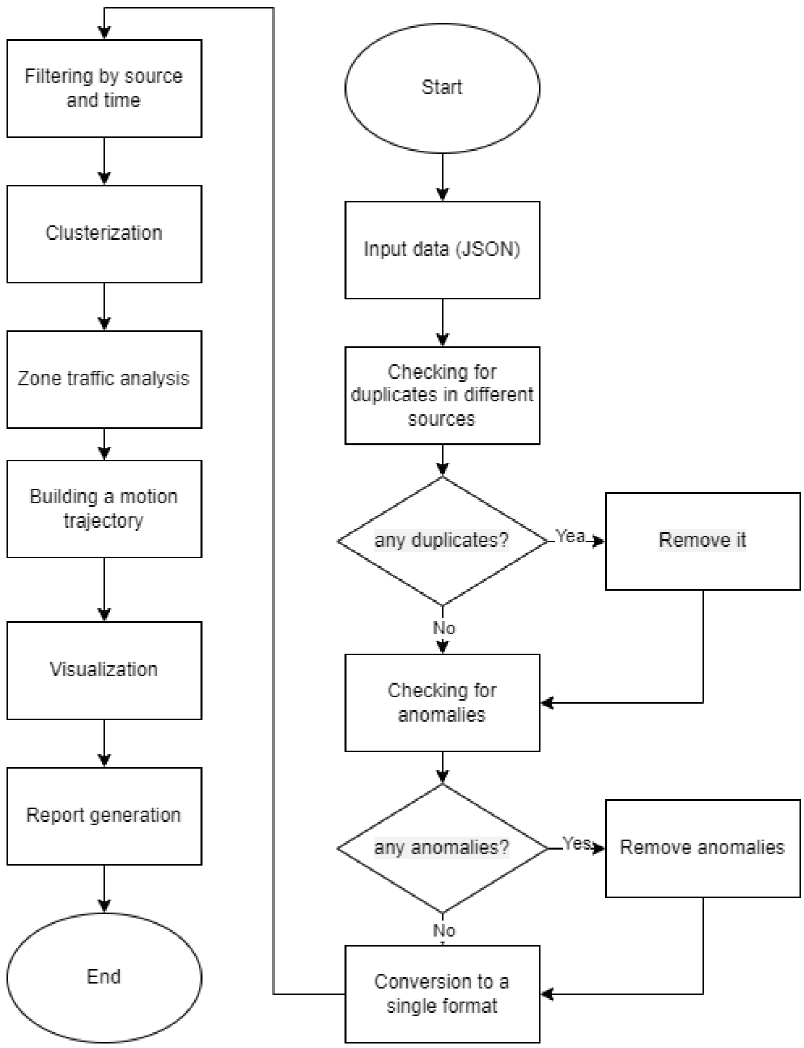


Figure 1 – Block diagram of the system operation

```python
import json
from datetime import datetime
import numpy as np
from sklearn.cluster import DBSCAN

def serve_offline_map(data):
    with open("static/offline_map.html", "r", encoding="utf-8") as f:
        html = f.read()

    points_json = json.dumps([
        {
            "lat": p["coordinates"]["lat"],
            "lon": p["coordinates"]["lon"],
            "timestamp": p["timestamp"],
            "source": p.get("source", "UNKNOWN").upper()
        }
        for p in data["trajectory_points"]
    ])
```

Figure 2 – Required type of incoming data in Json format

To prepare the dataset for clustering and trajectory analysis, a multi-step preprocessing pipeline is implemented. This step is crucial to clean, normalize, and structure the data in a form that allows consistent mathematical treatment. As emphasized by Petrescu et al. [2], trajectory datasets derived from real devices are often noisy, irregularly sampled, and may include corrupted or semantically redundant points.

Each entry is validated individually. Points with missing values, zeroed coordinates, or unreasonable accuracy values (e.g., over 10,000 meters) are removed. This ensures that subsequent calculations, especially those involving distance or clustering, are not distorted by invalid data.

The dataset is then chronologically sorted based on timestamps. Since timestamps are initially provided in ISO 8601 human-readable format, they are converted into Unix epoch time (the number of seconds since 1 January 1970 UTC). This conversion enables straightforward computation of time differences and alignment of asynchronous observations from multiple sources.

Where needed (especially during distance calculations), coordinates are converted from degrees to radians, enabling trigonometric operations such as those used in the haversine formula. This ensures the geospatial integrity of computed values like step distances, cluster radii, and overall route length.

The outcome of this preprocessing stage is a temporally ordered and spatially consistent sequence of geolocation points. These cleaned and normalized data are then passed to the clustering module, where they serve as the foundation for route reconstruction and behavior analysis.

This structure is modeled mathematically as a sequence of observations:

$$D = \{d_i = (t_i, x_i = (\varphi_i, \lambda_i), a_i, s_i)\}_{i=1}^{n} \tag{1}$$

Where:

- $t_i \in$ Ris a time value (after conversion to Unix timestamp),
- $x_i \in \mathrm{R}^2$ is the spatial coordinate pair: latitude and longitude,
- $a_i \in \mathrm{R}{\geq}0$ is the reported accuracy,
- $s_i \in S$ is a label from a finite set of known source types.

Unlike traditional datasets with fixed intervals and clean annotations, this real-world format embraces irregular sampling, missing intervals, and varying source trustworthiness. However, it is precisely this challenge that the proposed framework is designed to overcome. By integrating rigorous preprocessing with unsupervised analysis techniques designed for resilience to noise, the system allows forensic experts to make sense of inconsistent yet highly informative data streams–without the need for manual annotation or supervised training.

3.2 Distance Calculation (Haversine Formula) and DBSCAN Clustering

All spatial comparisons are done using the haversine formula, which calculates the great-circle distance between two points on Earth. Coordinates are expressed in WGS-84

Given two locations:

$$x_1=(\phi_1,\lambda_1),\ x_2=(\phi_2,\lambda_2) \tag{2}$$

the spherical distance in meters is:

$$d=2r\cdot\arcsin\left(\sqrt{\left(\sin^2\left(\frac{\Delta\varphi}{2}\right)+\cos(\phi_1)\cos(\varphi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)\right)}\right) \tag{3}$$

Where:

1. $-\Delta\phi=\phi_2-\phi_1$,
2. $-\Delta\lambda=\lambda_2-\lambda_1$,
3. $-r=6{,}371{,}000$ *meters*.

This metric is used for clustering and route calculations.

The *DBSCAN algorithm* identifies clusters of spatially dense points,$C_j\subseteq D$. It has two parameters:

- ε: maximum distance to be considered part of a neighborhood (typically 30–50 meters),

- minPts: the minimum number of points to form a dense cluster (typically 3–5).

A point p is a core point if:

$$|N_\varepsilon(p)|\geq \min Pts,\ N_\varepsilon(p)=q\in D|d(p,q)\leq\varepsilon \tag{4}$$

The algorithm constructs clusters by linking core points and their reachable neighbors.

In figure 3 shows Clustered locations visualized on a map using the DBSCAN algorithm (ε = 50 m, minPts = 3). The visualization covers the observation period from **December 7, 2021 to July 16, 2025**, showing trajectory points derived from **image (JPG)** and **video (MP4)** metadata. All coordinates are expressed in the **WGS-84 (EPSG:4326)** coordinate reference system, and distances are computed geodesically using the haversine formula. Clustered zones are highlighted as **red circular markers**, while isolated trajectory points are shown in neutral tones to indicate noise or transitional movement.This figure demonstrates how spatially dense locations are detected and grouped by DBSCAN, forming clusters annotated with centroid coordinates, visit counts, and time intervals

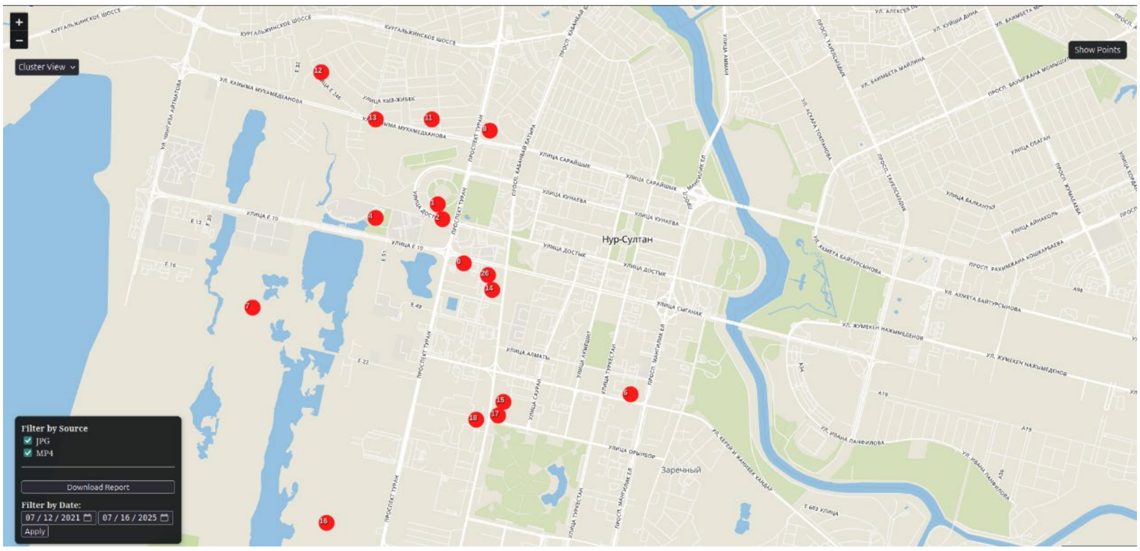


Figure 3 – Clustered locations visualized on map using DBSCAN.

Each resulting cluster C_j is annotated with:
- Centroid:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i \qquad (5)$$

- Time interval:

$$T_j = [\min(t_i), \max(t_i)], d_i \in C_j \qquad (6)$$

- Visit count: $N_j = |C_j|$,
- Source set: $S_j = \{ s_i \mid d_i \in C_j \}$

3.3 The reconstruction of the route, filtering and visualization tools

The trajectory T is the ordered list of coordinates:

$$T = x_{(1)}, x_{(2)}, \ldots, x_{(n)} \qquad (7)$$

The route is defined as:

$$Route = \left(x_{(i)}, x_{(i+1)}\right) | 1 \leq i < n \qquad (8)$$

This is visualized on the map as a polyline. Interpolation is currently not applied. From the raw data set, a Trajectory is recreated reflecting the exact sampling rate and continuity of movement.

Then we are going to filtering sources: Every point has a category-based source label. During visualization, the user can apply a filter, $S' \subseteq S$ to create a new dataset:

$$D_{S'} = d_i \in D | s_i \in S' \qquad (9)$$

In Figure 4, the first process of grouping JSON data is highlighted. This process shows how the system derives the latitude and longitude values from the input data, which are then transformed from meters to radians in accordance with the WGS-84 reference system. By this means, the algorithm DBSCAN is executed with epsilon set at 50 meters and the value of minPts. Every identified group is marked with a relevant ID, center, and count values that are stored in a JSON summary.

```python
coords = np.array([[p["coordinates"]["lat"], p["coordinates"]["lon"]] for p in data["trajectory_points"]])
coords_rad = np.radians(coords)

db = DBSCAN(eps=50 / 6371000, min_samples=1, metric='haversine').fit(coords_rad)
labels = db.labels_

cluster_summary = []
for label in set(labels):
    pts = coords[labels == label]
    centroid = pts.mean(axis=0)
    cluster_summary.append({
        "id": int(label) + 1,
        "lat": float(centroid[0]),
        "lon": float(centroid[1]),
        "visits": len(pts)
    })

coords = [(p["coordinates"]["lat"], p["coordinates"]["lon"]) for p in data["trajectory_points"]]
cluster_data = generate_cluster_data(coords)
cluster_json = json.dumps(cluster_data)
```

Figure 4 – The initial process of clustering json data

Then all clusters and trajectory lines are recalculated using only the filtered set. Using only filtered points allows you to selectively analyze GPS-only data, indoor data (for example, Wi-Fi), or image-based sources. Filtering is applied dynamically, automatically updating visual changes on the map.

3.4 Visualization Interface and Offline Map Rendering

The final stage of the trajectory_analyzer system involves the generation of a fully interactive, offline-capable geolocation visualization interface. Unlike other existing systems based on online mapping services, our software solution creates a dynamic HTML-based report that allows you to study the user's status when moving in real time without having to use any external servers, without network access or third-party API integrations.

The visual output consists of an HTML file (report.html) related to JavaScript and CSS user resources (report_template.html report_style.css) and local map sheets. The latter provides complete offline operation, eliminating dependence on external maps such as OpenStreetMap or Mapbox. The application weighs more than typical cloud-based visualizers due to the embedded tile storage, but offers a practical trade-off in the context of digital forensics, where data sovereignty, stability, and network isolation are often essential.

The interactive map interface itself is not rendered using Folium or Leaflet directly; rather, the system uses a custom-built frontend. Leaflet is utilized only for low-level map layer handling, such as zooming and tile display. All higher-order functionality–including cluster rendering, filter toggles, UI panels, and event responses–is implemented manually using vanilla JavaScript and custom CSS, providing full control over the logic and appearance of the visualization.

After filtering the input data completely, each point of movement is displayed in chronological order, drawing a continuous trajectory of movement. Color coding is applied to the type of data shown, whether it is a route, trajectory, blue, as shown in Figure 5. Clustered zones, red. This allows analysts to immediately distinguish between categories of data. Individual points on the route are interactive. When you hover the mouse over them, pop-up windows appear displaying the point's index, source, timestamp, and the number of visits to that location. Unlike many clustering systems that visualize only centroids or aggregate data, this implementation emphasizes granularity, exposing every recorded stop to detailed inspection.

In parallel, clustered locations–calculated through DBSCAN as described in earlier sections–are rendered using larger custom markers, visually distinguishing them from transient path points. These clusters include summary pop-ups detailing the average coordinates (centroid), the time span during which the cluster was active, and the number of constituent records. This dual-layer view (trajectory path + static clusters) allows the analyst to quickly separate stationary behavior (e.g., place visits) from transitional motion (e.g., commuting or travel).

A collapsible side panel is integrated into the map interface, providing investigators with a interactive filtering mechanism. Through intuitive checkboxes and sliders, the user can toggle visibility of specific data sources or limit the visualized route to a selected time interval. These controls operate in real time and require no page reloads or backend reprocessing. This interactivity allows analysts to test hypotheses, isolate anomalies, or correlate movement patterns with other data (e.g., crime timestamps, device logs).

Below the map, a set of visual summaries is presented in the form of interactive charts and diagrams. These include:

- A pie chart of the most frequently visited locations (by cluster density),
- A bar chart of the last N visited places,
- A chronological list of all locations in order, with metadata including time, coordinates, and source.

These visualizations are automatically generated during the report creation process and provide compact insight into behavioral tendencies, such as routine places and movement regularity. All diagrams are embedded within the HTML file and rendered with client-side JavaScript libraries, ensuring they remain functional even in isolated environments.

Finally, a dedicated button is available for exporting a full forensic report as a compressed .zip archive (report.zip). This export contains:

- The full visualization HTML,
- All embedded resources (CSS, JS, map tiles),
- A JSON summary of the clustered and raw data,
- A preformatted PDF-style document with detailed tables of all recorded points, sources, and cluster summaries.

This modular reporting format ensures accuracy, convenience and transparency in accordance with the best practices of digital forensics. The non-overloaded interface design focuses on the convenience of searching and working with it quickly. The report provides high-resolution spatial detail, provides a temporal context, and shows the complete chronological sequence of the vehicle's movement. all this works without compromising security, because the entire system is independent of Internet services.

Additionally, the report has two chart variations. The first diagram shows the top visited locations over a total period (75 m per pixel tile resolution) of time and is shown below as Figure 5. The second diagram is designed to view the most recently visited locations and is shown in Figure 6.

Thus, the trajectory_analyzer visualization layer transforms the raw geolocation data into a user-friendly, reliable interface from the point of view of forensic examination. With offline functionality, interpretability, and interactivity, it serves as both a diagnostic tool and a formal reporting mechanism in investigative workflows.

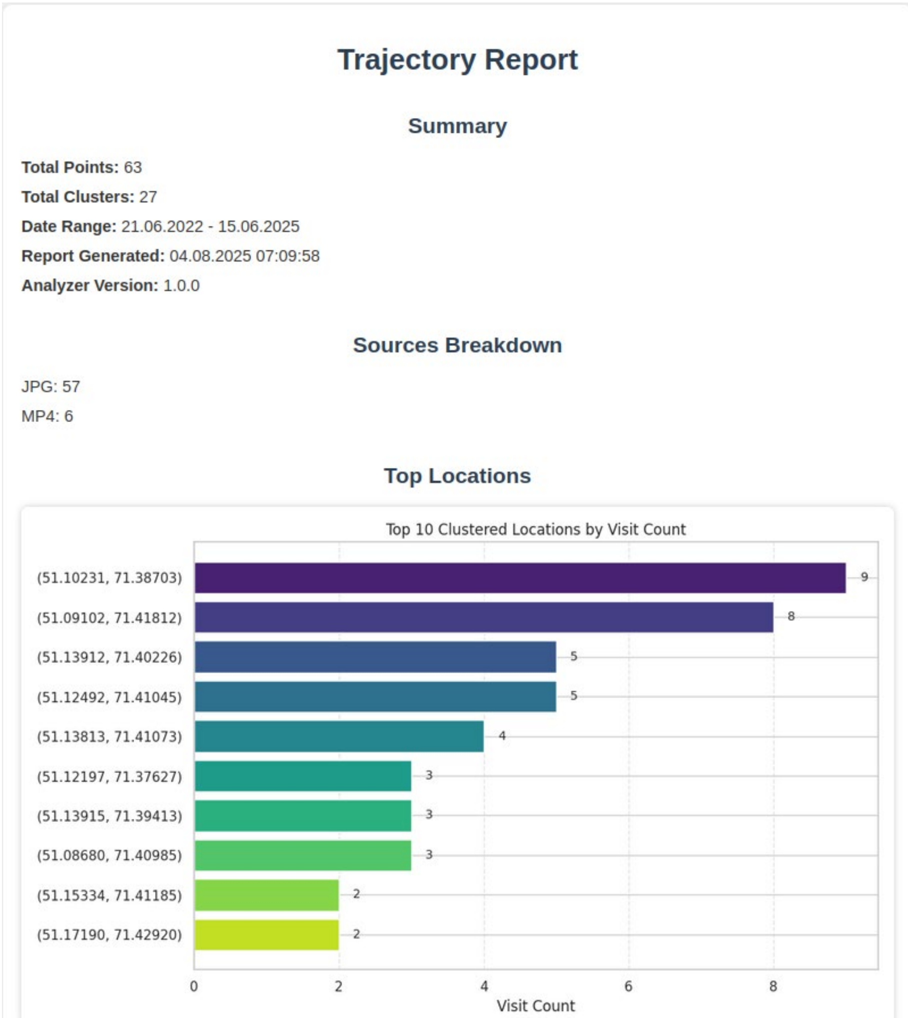
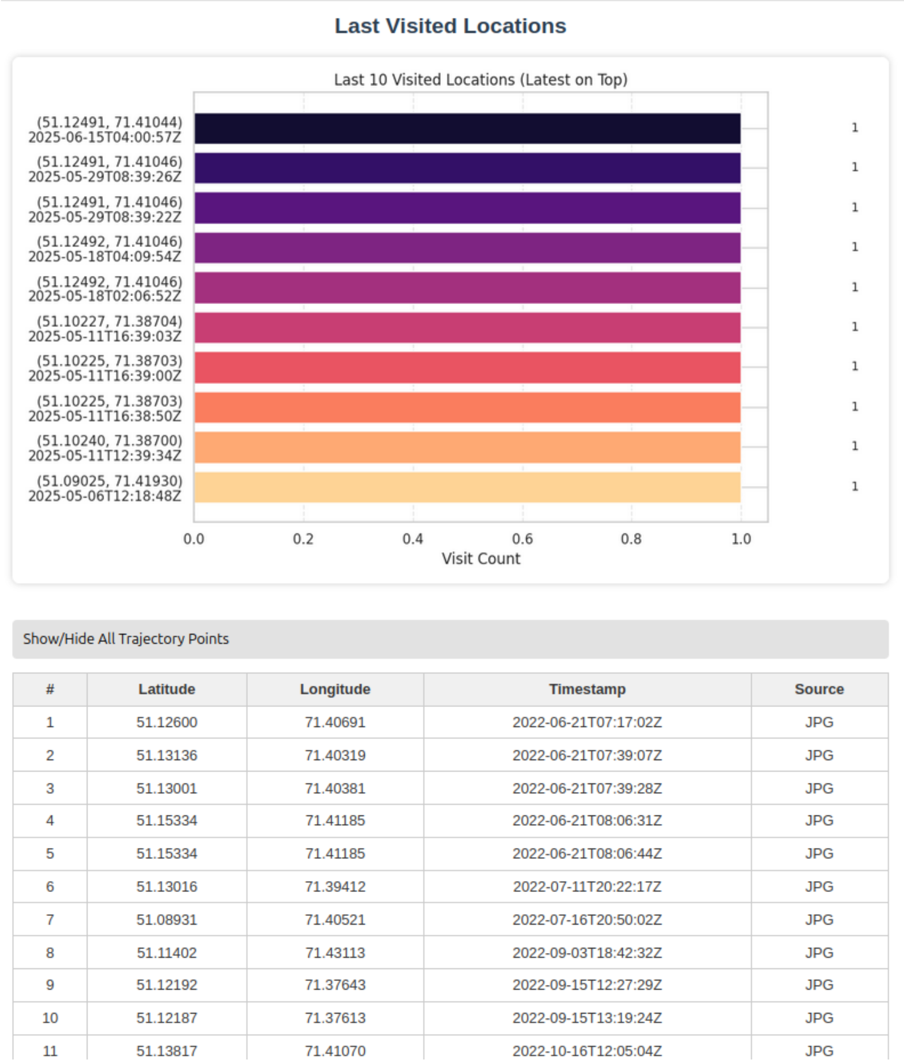


Figure 5 – Most visited locations (total)



#	Latitude	Longitude	Timestamp	Source
1	51.12600	71.40691	2022-06-21T07:17:02Z	JPG
2	51.13136	71.40319	2022-06-21T07:39:07Z	JPG
3	51.13001	71.40381	2022-06-21T07:39:28Z	JPG
4	51.15334	71.41185	2022-06-21T08:06:31Z	JPG
5	51.15334	71.41185	2022-06-21T08:06:44Z	JPG
6	51.13016	71.39412	2022-07-11T20:22:17Z	JPG
7	51.08931	71.40521	2022-07-16T20:50:02Z	JPG
8	51.11402	71.43113	2022-09-03T18:42:32Z	JPG
9	51.12192	71.37643	2022-09-15T12:27:29Z	JPG
10	51.12187	71.37613	2022-09-15T13:19:24Z	JPG
11	51.13817	71.41070	2022-10-16T12:05:04Z	JPG

Figure 6 – Recent visits by location.

4. Results and Discussion

The developed system successfully processed multi-source geolocation data and visualized user movement patterns, including route reconstruction and clustered visit locations. Compared to previous DBSCAN-based frameworks [5], the integration of preprocessing steps and source-aware filtering appears to improve the clarity and reliability of clustering outcomes.

On Figure 7, shows the complete trajectory derived from raw multi-source data, plotted in chronological order within the WGS-84 coordinate system. Each point represents an individual recorded location, while the continuous blue line visualizes the sequential path of movement over the full observation period (December 7, 2021 – July 16, 2025). This figure demonstrates the system's ability to reproduce detailed movement routes without clustering, preserving temporal accuracy and source integrity.

In contrast to semi-supervised pipelines described in [6][7], the fully offline nature of our tool enhances responsiveness and usability, especially in privacy-sensitive environments. Visualization is rendered nearly instantaneously for small and medium datasets, supporting quick interpretation during local forensic investigations.

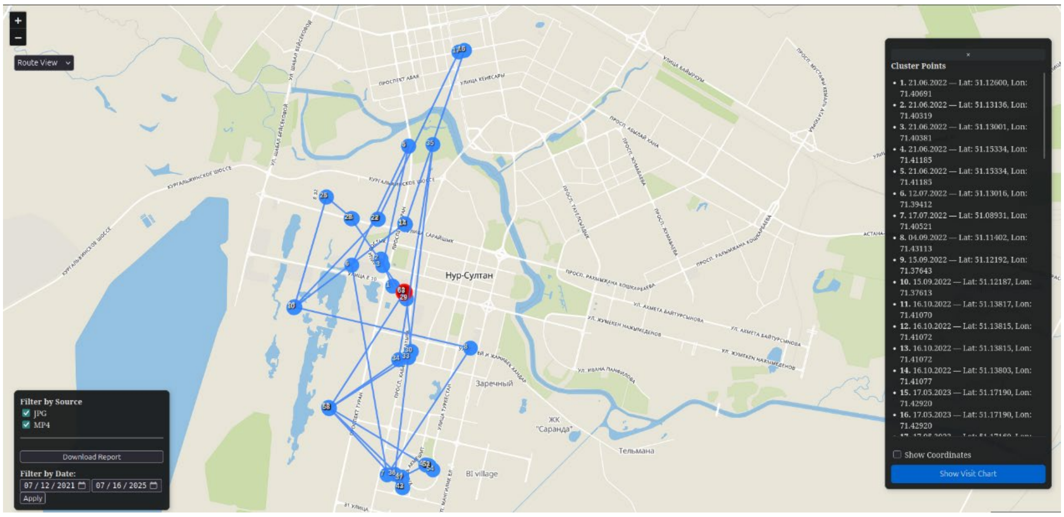


Figure 7 – User movement route reconstructed from raw data (shown on an interactive map).

This approach aligns well with needs in forensic practice, where fast access, transparency, and data locality are often prioritized over dependence on remote APIs or web-based solutions. However, the system's local map rendering and multiple visual layers may result in greater disk usage than lightweight alternatives [3][9] and [16].

Overall, trajectory_analyzer demonstrates a practical and interpretable method for digital forensic mobility analysis, with strong applicability in settings that require secure and autonomous data processing.

Conclusion

This study introduced *trajectory_analyzer*, a modular and fully offline system for reconstructing and visualizing geolocation data in forensic investigations. The framework integrates key technical components–including temporal preprocessing, spherical distance computation using the Haversine formula, and unsupervised clustering via DBSCAN–to extract meaningful behavioral patterns from unstructured, multi-source data.

A major novelty of the system is its **combined approach**, which unites:

- real unsupervised clustering,
- spherical distance metrics,
- source-aware dynamic recomputation,

and a fully offline, interactive visualization layer.

This design enables the tool to operate independently of cloud services or training datasets, making it ideal for use in sensitive forensic contexts where **data privacy, reproducibility, and speed** are paramount. Investigators can explore clusters, trace user routes, and analyze the role of different data sources–all within an interpretable and responsive interface.

Despite its advantages in speed, the system's reliance on local map assets increases storage requirements, which may limit portability in some scenarios.

Future work will focus on several directions:

1. Automating the selection of DBSCAN parameters (ε and minPts) for different dataset scales.
2. Optimizing the visual layers footprint through compressed or vector-based tile storage.
3. Extending support for additional input and export formats.

Overall, *trajectory_analyzer* delivers a practical, transparent, and extensible solution for geolocation analysis, one that aligns with the needs of modern digital forensics for **modular, offline, and interpretable** tools.

Funding

This study was carried out with the financial support of the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan under Contract №388/PTF-24-26 dated 01.10.2024 under the scientific project IRN

BR24993232 “Development of innovative technologies for conducting digital forensic investigations using intelligent software-hardware complexes”.

Author Contributions

Conceptualization, N.N., and A.B.; Methodology L.R., M.Zh.; Software, N.N., B.N., and A.B.; Validation, N.N., and B.N.; Formal Analysis, N.N., and B.N.; Investigation, A.B.; Resources, N.N., and B.N.; Data Curation, N.N., and A.B.; Writing – Original Draft Preparation, B.N.; Writing – Review & Editing, B.N. and A.M.; Visualization, B.N.; Supervision, A.M.; Project Administration, N.N.; Funding Acquisition, N.N.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Macedo, I., Wanous, S., Oliveira, N., Sousa, O., & Praça, I. (2021, March). A tool to support the investigation and visualization of cyber and/or physical incidents. In World Conference on Information Systems and Technologies (pp. 130-140). Cham: Springer International Publishing.

2. Li, A., Xu, Z., Zhang, J., Li, T., Cheng, X., & Hu, C. (2023). A Vector Field Visualization Method for Trajectory Big Data. *ISPRS International Journal of Geo-Information*, 12(10), 398.

3. Matei, A. D., Talavera, E., & Aghaei, M. (2025). Crime scene classification from skeletal trajectory analysis in surveillance settings. *Engineering Applications of Artificial Intelligence*, *141*, 109800.

4. Ashworth, G., Waldron, G., & Kahai, B. (2025). A service evaluation exploring staff perceptions about the use and impact of electronic monitoring (GPS tracking) in a medium secure forensic psychiatric unit. *The Journal of Forensic Psychiatry & Psychology*, 1-18.

5. Yu, X., Long, W., Li, Y., Gao, L., & Shi, X. (2022). Trajectory dimensionality reduction and hyperparameter settings of DBSCAN for trajectory clustering. *IET Intelligent Transport Systems*, *16*(5), 691-710.

6. Abruzzo, A., Ferrante, M., & De Cantis, S. (2021). A pre-processing and network analysis of GPS tracking data. *Spatial Economic Analysis*, *16*(2), 217-240.

7. Sadeghian, P., Håkansson, J., & Zhao, X. (2021). Review and evaluation of methods in transport mode detection based on GPS tracking data. *Journal of Traffic and Transportation Engineering (English Edition)*, *8*(4), 467-482.

8. Li, L., Li, J., & Zhang, S. (2021). State-of-the-art trajectory tracking of autonomous vehicles. *Mechanical Sciences*, *12*(1), 419-432.

9. Klymenko, M. V., & Striuk, A. M. (2023). Design and implementation of an edge computing-based GPS tracking system. *Journal of Edge Computing*, *2*(2), 175-189.

10. Reyes, G., Lanzarini, L., Estrebou, C., & Bariviera, A. (2022, October). Data Stream Processing Method for Clustering of Trajectories. In *International Conference on Technologies and Innovation* (pp. 151-163). Cham: Springer International Publishing.

11. Huang, W., & Wang, L. (2022). Towards big data behavioral analysis: rethinking GPS trajectory mining approaches from geographic, semantic, and quantitative perspectives. *Architectural Intelligence*, *1*(1), 7.

12. Wei, B., Zhang, J., Hu, C., & Wen, Z. (2023). A Clustering Visualization Method for Density Partitioning of Trajectory Big Data Based on Multi-Level Time Encoding. *Applied Sciences*, *13*(19), 10714.

13. Idrissova, M., Kim, S., Amirgaliyev, B., Yedilkhan, D., & Rzayeva, L. (2025). Digital footprints: clustering browser history for user profiling using machine learning. *Journal of Problems in Computer Science and Information Technologies*, *3*(2), 16-28.

14. Rzayeva, L., D. Pogolovkin, & I. Shaya. (2025). Development of a correspondence analysis service using artificial intelligence technology and a vector database for digital forensics. *International journal of information and communication technologies*, *6*(1), 201–225.

15. Mirza, M. M., & Karabiyik, U. (2021, October). Enhancing IP address geocoding, geolocating and visualization for digital forensics. In *2021 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-7). IEEE.

16. Pace, L. R., Salmon, L. A., Bowen, C. J., Baggili, I., & Richard III, G. G. (2023). Every step you take, I'll be tracking you: Forensic analysis of the tile tracker application. *Forensic science international: digital investigation*, *45*, 301559.

17. Daraghmi, E., & Hamoudi, A. Mobile Forensics: Extracting Geo-Location Data from Photos on Android Smartphones. *vol*, *9*, 1915-1921.

Information About Authors

Bekarys Nurzhaubaev is a bachelor’s student at Astana IT University and a junior software developer at the Research and Innovation Center (RIC) “CyberTech” at Astana IT University. His research interests include machine learning, neural networks, and the development of systems integrating machine learning and neural network algorithms. OrcID: 0009-0008-4683-9350

Nursultan Nyssanov is a master’s student at Astana IT University, a software developer, and a researcher at the Research and Innovation Center (RIC) “CyberTech” at Astana IT University. His research interests include machine learning, neural networks, the development of systems integrating machine learning and neural network algorithms, and digital forensics. OrcID: 0009-0002-8128-0595.

Alisher Batkuldin holds an M.Sc. in Electrical and Computer Engineering from Nazarbayev University. He is a software developer and researcher at the Research and Innovation Center (RIC) "CyberTech" at Astana IT University. His research interests include machine learning, neural networks, the development of systems integrating machine learning and neural network algorithms, and digital forensics. OrcID: 0009-0004-2097-5419.

Ali Myrzatay holds a Ph.D. in Automation and Control from L.N. Gumilyov Eurasian National University. He works at the "Digital Government Support Center" RSE on the REM, Astana, Kazakhstan, as a project manager and as a researcher at the Research and Innovation Center (RIC) "CyberTech" at Astana IT University. His research interests include machine learning, neural networks, and the development of systems integrating machine learning and neural network algorithms. OrcID: 0000-0002-5339-2437.

Murat Zhakenov is a Candidate of Technical Sciences and works at "Digital Heritage of Eurasia" LLP, Astana, Kazakhstan. His research interests include machine learning and neural networks. OrcID: 0009-0005-9672-4365.

Submission received: 08 August, 2025.
Revised: 19 November, 2025.
Accepted: 19 November, 2025.